
INVENT

Release 1.0.0

Girish Joshi

Oct 06, 2021

CONTENTS:

1	About	3
2	Tutorials	5
3	API Reference	7
3.1	InvManage	7
3.1.1	Subpackages	7
3.1.1.1	InvManage.forms	7
3.1.1.1.1	Submodules	7
3.1.1.1.1.1	InvManage.forms.forms	7
3.1.1.1.1.2	Module Contents	7
3.1.1.1.1.3	Classes	7
3.1.1.1.1.4	InvManage.forms.vendor_forms	18
3.1.1.1.1.5	Module Contents	18
3.1.1.1.1.6	Classes	18
3.1.1.1.2	Package Contents	22
3.1.1.1.2.1	Classes	22
3.1.1.2	InvManage.models	37
3.1.1.2.1	Submodules	37
3.1.1.2.1.1	InvManage.models.misc	37
3.1.1.2.1.2	Module Contents	37
3.1.1.2.1.3	Classes	37
3.1.1.2.1.4	InvManage.models.objects	39
3.1.1.2.1.5	Module Contents	39
3.1.1.2.1.6	Classes	39
3.1.1.2.1.7	InvManage.models.reuse	52
3.1.1.2.1.8	Module Contents	52
3.1.1.2.1.9	Classes	52
3.1.1.2.1.10	InvManage.models.states	56
3.1.1.2.1.11	Module Contents	56
3.1.1.2.1.12	Classes	56
3.1.1.2.2	Package Contents	57
3.1.1.2.2.1	Classes	57
3.1.1.3	InvManage.scripts	75
3.1.1.3.1	Submodules	75
3.1.1.3.1.1	InvManage.scripts.filters	75
3.1.1.3.1.2	Module Contents	75
3.1.1.3.1.3	Functions	75
3.1.1.4	InvManage.views	76
3.1.1.4.1	Submodules	76

3.1.1.4.1.1	InvManage.views.company_views	76
3.1.1.4.1.2	Module Contents	76
3.1.1.4.1.3	Functions	76
3.1.1.4.1.4	InvManage.views.consumer_views	80
3.1.1.4.1.5	Module Contents	80
3.1.1.4.1.6	Functions	80
3.1.1.4.1.7	InvManage.views.goods_receipt_note_views	84
3.1.1.4.1.8	Module Contents	84
3.1.1.4.1.9	Functions	84
3.1.1.4.1.10	InvManage.views.history_views	92
3.1.1.4.1.11	Module Contents	92
3.1.1.4.1.12	Functions	92
3.1.1.4.1.13	InvManage.views.product_views	94
3.1.1.4.1.14	Module Contents	94
3.1.1.4.1.15	Functions	94
3.1.1.4.1.16	InvManage.views.purchase_order_views	98
3.1.1.4.1.17	Module Contents	98
3.1.1.4.1.18	Functions	98
3.1.1.4.1.19	InvManage.views.sales_order_views	110
3.1.1.4.1.20	Module Contents	110
3.1.1.4.1.21	Functions	110
3.1.1.4.1.22	InvManage.views.vendor_views	118
3.1.1.4.1.23	Module Contents	118
3.1.1.4.1.24	Functions	118
3.1.1.4.2	Package Contents	124
3.1.1.4.2.1	Classes	124
3.1.1.4.2.2	Functions	131
3.1.2	Submodules	436
3.1.2.1	InvManage.filters	436
3.1.2.1.1	Module Contents	436
3.1.2.1.1.1	Classes	436
3.1.2.2	InvManage.serializers	439
3.1.2.2.1	Module Contents	439
3.1.2.2.1.1	Classes	439
4	Indices and tables	463
	Python Module Index	465
	HTTP Routing Table	467
	Index	469

INVENT is an inventory management software. It is a minimalistic demo application based on *Django* web framework. It was created to demonstrate the rapid prototyping capabilities of *Django* for web application development.

The screenshot displays the INVENT web application interface. On the left, a sidebar menu contains links for Company, Purchase, Inventory, Sales, and Reports. Below the menu is a 'History' panel showing a list of recent transactions, including PurchaseOrders and GoodsReceiptNotes, with their status (Created or Deleted) and timestamps. The main area features a 'Products' table with columns for Name, Category, Quantity, Identifier, Location, Price, and actions (Update, Delete). The table lists 16 products, each with a unique identifier and price. At the bottom of the table, there are pagination controls showing 'Page 4 of 16' and a count of 'Count: 46 - 60 of 239'.

Name	Category	Quantity	Identifier	Location	Price	Update	Delete
Surdo	Parturient Montes Corporation	21238	PROD65	Deschambault	2660.0	Update	Delete
Den-den daiko	Duis Sit PC	13932	PROD66	St. Albans	1811.0	Update	Delete
Inyhura	Ut Nisi PC	2254	PROD67	Parbhani	23.0	Update	Delete
Maktoum (maktoum, katem)	Cursus Industries	5956	PROD68	Bozeman	1625.0	Update	Delete
Krakebs	Quam Elementum At Incorporated	41596	PROD69	Ucluelet	1686.0	Update	Delete
Nagara	Egestas Nunc LLC	2411	PROD70	Berg	2098.0	Update	Delete
Ashiko	Pulvinar Limited	38169	PROD71	Cargovil	1410.0	Update	Delete
Maddale	Odio Phasellus At Associates	32774	PROD72	Hohenems	519.0	Update	Delete
Mirwas	Vulputate Dui Nec Consulting	26068	PROD73	Heusden-Zolder	497.0	Update	Delete
Cuica	Lectus LLC	47733	PROD74	Hachy	2514.0	Update	Delete
Barrel drum	Sit Corp.	43018	PROD75	Wolfsburg	965.0	Update	Delete
buleador	Diam Pellentesque Habitant Company	48359	PROD76	San Sebastiano al Vesuvio	3.0	Update	Delete
Uruttu chenda	Parturient Montes Nascetur LLP	34947	PROD77	Los Angeles	1660.0	Update	Delete
Tapan, Topan, Tupan	Sapient Aenean PC	33656	PROD78	Greenock	884.0	Update	Delete
Dimdi	Mauris Nulla PC	11115	PROD79	New Haven	1539.0	Update	Delete

ABOUT

INVENT is a Django based RESTful web application. It implements modular and reusable code with clear separation of concerns using Model-Template-View architectural pattern.

- *Models* describe the logical data structure.
- Templates represent static HTML fragments of the user interface.
- *Views* are responsible for handling requests and deciding the response.

In addition, the application uses [Django Forms API](#) and [Django REST framework](#). Django Forms API allows hassle free rendering of web forms, while Django REST framework provides web browsable API, authentication, and serialization capabilities. See [Forms](#) and [Serializers](#) sections of the documentation.

The application exposes the following endpoints, which are listed in the `urls.py` file and documented in [Views](#) section.:

```
# Routes for managing company
path('company/', create_company_view, name='company'),
path('companies/', display_companies_view, name='companies'),
path('company/update', update_company_view, name='update_company'),
path('company/<str:pk>/delete', delete_company_view, name='delete_company'),

# Routes for managing products
path('product/', create_product_view, name='product'),
path('products/', display_products_view, name='products'),
path('product/update', update_product_view, name='update_product'),
path('product/<str:pk>/delete', delete_product_view, name='delete_product'),

# Routes for managing vendors
path('vendor/', create_vendor_view, name='vendor'),
path('vendors/', display_vendors_view, name='vendors'),
path('vendor/update', update_vendor_view, name='update_vendor'),
path('vendor/<str:pk>/delete', delete_vendor_view, name='delete_vendor'),
path('get_vendor/', get_vendor, name='get_vendor'), # ajax call in create purchase order.
↳ on vendor dropdown

# Routes for managing consumers
path('consumer/', create_consumer_view, name='consumer'),
path('consumers/', display_consumers_view, name='consumers'),
path('consumer/update', update_consumer_view, name='update_consumer'),
path('consumer/<str:pk>/delete', delete_consumer_view, name='delete_consumer'),
path('get_consumer/', get_consumer, name='get_consumer'), # ajax call in create sales.
↳ order on consumer dropdown
```

(continues on next page)

(continued from previous page)

```
# Routes for managing GRNs
path('grn/', create_grn_view, name='grn'),
path('grns/', display_grns_view, name='grns'),
path('grn/update', update_grn_view, name='update_grn'),
path('grn/<str:pk>/delete', delete_grn_view, name='delete_grn'),
path('grn/<str:pk>/print', print_grn_view, name='print_grn'),

# Routes for managing POs
path('purchase_order/', create_purchase_order_view, name='purchase_order'),
path('purchase_orders/', display_purchase_orders_view, name='purchase_orders'),
path('purchase_order/update', update_purchase_order_view, name='update_purchase_order'),
path('purchase_order/<str:pk>/delete', delete_purchase_order_view, name='delete_purchase_
↳ order'),
path('purchase_order/<str:pk>/print', print_purchase_order_view, name='print_purchase_
↳ order'),
path('product_purchase_entries/', get_product_purchase_entries_view, name='product_
↳ purchase_entries'), # JSON API to get PPEs of POs

# Routes for managing SOs
path('sales_order/', create_sales_order_view, name='sales_order'),
path('sales_orders/', display_sales_orders_view, name='sales_orders'),
path('sales_order/update', update_sales_order_view, name='update_sales_order'),
path('sales_order/<str:pk>/delete', delete_sales_order_view, name='delete_sales_order'),
path('sales_order/<str:pk>/print', print_sales_order_view, name='print_sales_order'),

path('upload/<data>/', uploadCSV, name='upload'),
path('admin/', admin.site.urls),

# Route for history
path('history/', display_history_view, name='history'),
```


TUTORIALS

INVENT handles purchase, inventory, and sales stages of the products. The following videos provide workflow related to these stages.

API REFERENCE

This page contains auto-generated API reference documentation¹.

3.1 InvManage

3.1.1 Subpackages

3.1.1.1 InvManage.forms

3.1.1.1.1 Submodules

3.1.1.1.1.1 InvManage.forms.forms

3.1.1.1.1.2 Module Contents

3.1.1.1.1.3 Classes

<i>ProductBasicInfoForm</i>	Form for basic product information.
<i>ProductDetailedInfoForm</i>	Form for product physical details
<i>ThumbnailForm</i>	For for product thumbnail image.
<i>ProductStorageInfoForm</i>	Form for storage information.
<i>ProductPricingForm</i>	Form for pricing information of the product.
<i>ProductStatusForm</i>	Form for locating and identifying the product.
<i>PurchaseOrderBasicInfo</i>	Form for basic information of the purchase order.
<i>GRNInfo</i>	Form for goods receipt note (GRN).
<i>SalesOrderBasicInfo</i>	Form for basic information on sales order.
<i>ConsumerForm</i>	Form for consumer.
<i>GRNEntryForm</i>	Form for goods receipt note entry (GRNE).
<i>ProductPurchaseEntryForm</i>	Form for product purchase entry.
<i>ProductSalesEntryForm</i>	Form for product sales entry.
<i>CompanyForm</i>	Form for company.
<i>HistoryForm</i>	Form for history view.

class `InvManage.forms.forms.ProductBasicInfoForm`
Bases: `django.forms.Form`

¹ Created with sphinx-autoapi

Form for basic product information.

name

Name of the product.

Type str

item_type

Type of the product.

Type str

category

Product category.

Type str

description

Short description of the product.

Type str

prefix = basic

context

name

item_type

category

description

class InvManage.forms.forms.ProductDetailedInfoForm

Bases: django.forms.Form

Form for product physical details

length

Length of the product.

Type str

width

Width of the product.

Type str

height

Height of the product.

Type str

weight

Weight of the product.

Type str

prefix = detailed

context

length

width

height

weight

class InvManage.forms.forms.ThumbnailForm

Bases: django.forms.Form

For for product thumbnail image.

image

Image of the product.

Type ImageField

prefix = thumbnail

image

class InvManage.forms.forms.ProductStorageInfoForm

Bases: django.forms.Form

Form for storage information.

barcode

Barcode of the product.

Type str

expiry

Expiry date of the product.

Type DateField

prefix = storage

barcode

context

expiry

class InvManage.forms.forms.ProductPricingForm

Bases: django.forms.Form

Form for pricing information of the product.

price

Price of the product.

Type float

discount

Default discount percentage on the product.

Type float

prefix = pricing

context

price

discount

class InvManage.forms.forms.ProductStatusForm

Bases: django.forms.Form

Form for locating and identifying the product.

quantity

Stock quantity of the product.

Type int

identifier

Unique identifier of the product.

Type str

location

Physical location of the product.

Type str

context

prefix = status

quantity**identifier****location**

class InvManage.forms.forms.PurchaseOrderBasicInfo

Bases: django.forms.Form

Form for basic information of the purchase order.

vendor

Vendor associated with the purchahse order.

Type *Vendor*

date

Date of the purchase order creation.

Type DateField

po

Purchase order number.

Type int

discount

Percentage of overall discount.

Type float

tax

Percentage of tax.

Type float

paid

Amount paid to the vendor.

Type float

balance

Balance amount to be paid to the vendor.

Type float

subtotal

Total of all the product purchase entries associated with the purchase order.

Type float
taxtotal
 Total tax applicable on the *subtotal*.
Type float
ordertotal
 Total price of the purchase order including *taxtotal*.
Type float
prefix = po
context
product_choices
vendor_choices
vendor
date
po
discount
tax
paid
balance
subtotal
taxtotal
ordertotal
class InvManage.forms.forms.GRNInfo
 Bases: django.forms.Form
 Form for goods receipt note (GRN).
vendor
 Vendor associated with the goods receipt note.
Type *Vendor*
poRef
 List of identifiers of the purchase orders from which the goods receipt note is derived.
Type *PurchaseOrder*
identifier
 Unique identifier of the goods receipt note.
Type str
date
 Date of GRN creation.
Type DateField
grnType
 Type of GRN (auto or manual).
Type str

amendNumber

Amendment number of the GRN.

Type int

amendDate

Amendment date.

Type DateField

transporter

Name of the transport/shipping service.

Type str

vehicleNumber

Vehicle number using which the products are shipped.

Type str

gateInwardNumber

Gate inward number of the vehicle.

Type str

preparedBy

Name/identifier of the person who created the goods receipt note.

Type str

checkedBy

Name/identifier of the person who validated the goods receipt note.

Type str

checkedBy

Name/identifier of the person who inspected the physical products in the goods receipt note.

Type str

approvedBy

Name/identifier of the authority who approved the goods receipt note.

Type str

prefix = grn**context****product_choices****vendor_choices****vendor**

TYPE_CHOICES = [['manual', 'Blank'], ['auto', 'PO Reference']]

poRef**identifier****date****grnType****amendNumber****amendDate**


```

transporter
vehicleNumber
gateInwardNumber
preparedBy
checkedBy
inspectedBy
approvedBy
class InvManage.forms.forms.SalesOrderBasicInfo
    Bases: django.forms.Form
    Form for basic information on sales order.
    consumer
        Consumer associated with the sales order.
        Type Consumer
    date
        Date of the sales order creation.
        Type DateField
    so
        Sales order number.
        Type int
    discount
        Percentage discount associated with the sales order.
        Type float
    tax
        Percentage of tax applicable.
        Type float
    paid
        Amount received from the consumer.
        Type float
    balance
        Amount balance with the consumer.
        Type float
    subtotal
        Total of all the product sales entries associated with the sales order.
        Type float
    taxtotal
        Total tax applicable on the subtotal.
        Type float
    ordertotal
        Total price of the sales order including taxtotal.
        Type float

```

```
    prefix = so
    context
    product_choices
    consumer_choices
    consumer
    date
    so
    discount
    tax
    paid
    balance
    subtotal
    taxtotal
    ordertotal

class InvManage.forms.forms.ConsumerForm
    Bases: django.forms.Form
    Form for consumer.

    name
        Name of the consumer.
        Type str

    identifier
        Unique identifier of the consumer.
        Type str

    gstin
        GSTIN number of the consumer.
        Type str

    phone
        Contact number.
        Type str

    address
        Address of the consumer.
        Type str

    email
        E-mail address of the consumer.
        Type str

    location
        City of the consumer.
        Type str

    prefix = consumer
```

```

    context
    name
    identifier
    gstin
    phone
    address
    email
    location
class InvManage.forms.forms.GRNEntryForm
    Bases: django.forms.Form
    Form for goods receipt note entry (GRNE).
    product
        Product associated with the goods receipt note entry.
        Type Product
    grne_id
        Unique identifier of the GRNE.
        Type int
    ppe_id
        Unique identifier of the product purchase entry associated with the GRNE.
        Type int
    quantity
        Ordered quantity of product with reference to product purchase entry.
        Type int
    remark
        Remarks of the quality engineer or the GRN creator about status of products received.
        Type str
    receivedQty
        Quantity of product received against the ordered quantity.
        Type str
    acceptedQty
        Quantity of product accepted as OK.
        Type str
    rejectedQty
        Quantity of product rejected (not OK, on HOLD, extra delivery, etc.)
        Type str
    prefix = form
    context
    product
    grne_id

```

ppe_id

quantity

remark

receivedQty

acceptedQty

rejectedQty

class InvManage.forms.forms.**ProductPurchaseEntryForm**

Bases: django.forms.Form

Form for product purchase entry.

ppe_id

Unique identifier of the product purchase entry.

Type int

product

Product associated wit the product purchase entry.

Type *Product*

quantity

Quantity of the product to be ordered.

Type int

price

Price of the product.

Type float

discount

Percentage discount on the product purchase.

Type float

prefix = form

context

ppe_id

product

quantity

price

discount

class InvManage.forms.forms.**ProductSalesEntryForm**

Bases: django.forms.Form

Form for product sales entry.

pse_id

Unique identifier of the product sales entry.

Type int

product

Product associated with the product sales entry.

Type *Product*

quantity

Quantity of the product.

Type int

price

Price of the product.

Type float

discount

Percentage discount on the product.

Type float

prefix = form

context

pse_id

product

quantity

price

discount

class InvManage.forms.forms.CompanyForm

Bases: django.forms.Form

Form for company.

name

Name of the company.

Type str

owner

Name of the owner of the company.

Type str

gstin

GSTIN number of the company.

Type str

phone

Contact number of the company.

Type str

address

Postal address.

Type str

email

E-mail address of the contact person.

Type str

location

City of the company.

```
        Type str
    prefix = comp
    context
    name
    owner
    gstin
    phone
    address
    email
    location

class InvManage.forms.forms.HistoryForm
    Bases: django.forms.Form
    Form for history view.
    qlen
        Count of the events visible on the history view.
        Type int
    prefix = history
    context
    qlen
```

3.1.1.1.1.4 InvManage.forms.vendor_forms

3.1.1.1.1.5 Module Contents

3.1.1.1.1.6 Classes

<i>VendorForm</i>	Form for basic informaion of the vendor.
<i>ShippingAddressForm</i>	For for shipping address.
<i>CommunicationForm</i>	Form for communication information.
<i>PurchaseDataForm</i>	Form for purchase related data.
<i>BankAccountForm</i>	Form for bank account details.

```
class InvManage.forms.vendor_forms.VendorForm
    Bases: django.forms.Form
    Form for basic informaion of the vendor.
    name
        Name of the vendor.
        Type str
    identifier
        Unique identifier of the vendor.
        Type str
```

```

gstn
    GSTIN number of the vendor.
        Type str
prefix = vend
context
name
identifier
gstn
class InvManage.forms.vendor_forms.ShippingAddressForm
    Bases: django.forms.Form
    For for shipping address.
title
    Title of the firm.
        Type str
name
    Name of the firm.
        Type str
phone
    Contact number of the firm.
        Type str
address
    Postal address.
        Type str
city
    City.
        Type str
state
    State.
        Type str
country
    Country.
        Type str
website
    Official website of the firm.
        Type URLField
post
    Postal code of the firm.
        Type str
prefix = ship
context

```

title
name
phone
address
city
state
country
website
post

```
class InvManage.forms.vendor_forms.CommunicationForm
    Bases: django.forms.Form
```

Form for communication information.

language
Language of communication.
Type str

phone
Contact number of the firm.
Type str

email
E-mail of the contact person.
Type EmailField

fax
Fax number.
Type str

prefix = com

context

language

phone

email

fax

```
class InvManage.forms.vendor_forms.PurchaseDataForm
    Bases: django.forms.Form
```

Form for purchase related data.

currency
Currency of purchase.
Type str

minorder
Minimum order quantity.
Type int

contactperson

Name of the contact person.

Type str**refcode**

Reference code.

Type str**transportmode**

Mode of transport of the shipment.

Type str**prefix = pdform****context****currency****minorder****contactperson****refcode****transportmode****class** InvManage.forms.vendor_forms.BankAccountForm

Bases: django.forms.Form

Form for bank account details.

name

Name of the bank.

Type str**branch**

Branch number of the bank.

Type str**region**

City in which the branch is located.

Type str**route**

Transit number.

Type str**number**

Bank account number.

Type int**accttype**

Type of bank account.

Type str**iban**

IBAN number.

Type str

code
Bank code.
Type str

branchcode
Branch code.
Type str

prefix = bank

context

name

branch

region

route

number

acctype

iban

code

branchcode

3.1.1.1.2 Package Contents

3.1.1.1.2.1 Classes

<i>ProductBasicInfoForm</i>	Form for basic product information.
<i>ProductDetailedInfoForm</i>	Form for product physical details
<i>ProductPricingForm</i>	Form for pricing information of the product.
<i>ProductPurchaseEntryForm</i>	Form for product purchase entry.
<i>ProductSalesEntryForm</i>	Form for product sales entry.
<i>ProductStatusForm</i>	Form for locating and identifying the product.
<i>ProductStorageInfoForm</i>	Form for storage information.
<i>PurchaseOrderBasicInfo</i>	Form for basic information of the purchase order.
<i>GRNInfo</i>	Form for goods receipt note (GRN).
<i>GRNEntryForm</i>	Form for goods receipt note entry (GRNE).
<i>ThumbnailForm</i>	For for product thumbnail image.
<i>SalesOrderBasicInfo</i>	Form for basic information on sales order.
<i>ConsumerForm</i>	Form for consumer.
<i>CompanyForm</i>	Form for company.
<i>HistoryForm</i>	Form for history view.
<i>ShippingAddressForm</i>	For for shipping address.
<i>VendorForm</i>	Form for basic informaion of the vendor.
<i>CommunicationForm</i>	Form for communication information.
<i>PurchaseDataForm</i>	Form for purchase related data.
<i>BankAccountForm</i>	Form for bank account details.

class InvManage.forms.ProductBasicInfoForm

Bases: `django.forms.Form`

Form for basic product information.

name

Name of the product.

Type `str`

item_type

Type of the product.

Type `str`

category

Product category.

Type `str`

description

Short description of the product.

Type `str`

prefix = basic

context

name

item_type

category

description

class `InvManage.forms.ProductDetailedInfoForm`

Bases: `django.forms.Form`

Form for product physical details

length

Length of the product.

Type `str`

width

Width of the product.

Type `str`

height

Height of the product.

Type `str`

weight

Weight of the product.

Type `str`

prefix = detailed

context

length

width

height

weight

class `InvManage.forms.ProductPricingForm`

Bases: `django.forms.Form`

Form for pricing information of the product.

price

Price of the product.

Type `float`

discount

Default discount percentage on the product.

Type `float`

prefix = `pricing`

context

price

discount

class `InvManage.forms.ProductPurchaseEntryForm`

Bases: `django.forms.Form`

Form for product purchase entry.

ppe_id

Unique identifier of the product purchase entry.

Type `int`

product

Product associated with the product purchase entry.

Type *`Product`*

quantity

Quantity of the product to be ordered.

Type `int`

price

Price of the product.

Type `float`

discount

Percentage discount on the product purchase.

Type `float`

prefix = `form`

context

ppe_id

product

quantity

price

```

    discount
class InvManage.forms.ProductSalesEntryForm
    Bases: django.forms.Form
    Form for product sales entry.
    pse_id
        Unique identifier of the product sales entry.
        Type int
    product
        Product associated with the product sales entry.
        Type Product
    quantity
        Quantity of the product.
        Type int
    price
        Price of the product.
        Type float
    discount
        Percentage discount on the product.
        Type float
    prefix = form
    context
    pse_id
    product
    quantity
    price
    discount
class InvManage.forms.ProductStatusForm
    Bases: django.forms.Form
    Form for locating and identifying the product.
    quantity
        Stock quantity of the product.
        Type int
    identifier
        Unique identifier of the product.
        Type str
    location
        Physical location of the product.
        Type str
    context

```

prefix = status

quantity

identifier

location

class InvManage.forms.ProductStorageInfoForm

Bases: django.forms.Form

Form for storage information.

barcode

Barcode of the product.

Type str

expiry

Expiry date of the product.

Type DateField

prefix = storage

barcode

context

expiry

class InvManage.forms.PurchaseOrderBasicInfo

Bases: django.forms.Form

Form for basic information of the purchase order.

vendor

Vendor associated with the purchase order.

Type *Vendor*

date

Date of the purchase order creation.

Type DateField

po

Purchase order number.

Type int

discount

Percentage of overall discount.

Type float

tax

Percentage of tax.

Type float

paid

Amount paid to the vendor.

Type float

balance

Balance amount to be paid to the vendor.

Type float

subtotal

Total of all the product purchase entries associated with the purchase order.

Type float

taxtotal

Total tax applicable on the *subtotal*.

Type float

ordertotal

Total price of the purchase order including *taxtotal*.

Type float

prefix = po

context

product_choices

vendor_choices

vendor

date

po

discount

tax

paid

balance

subtotal

taxtotal

ordertotal

class InvManage.forms.GRNInfo

Bases: django.forms.Form

Form for goods receipt note (GRN).

vendor

Vendor associated with the goods receipt note.

Type *Vendor*

poRef

List of identifiers of the purchase orders from which the goods receipt note is derived.

Type *PurchaseOrder*

identifier

Unique identifier of the goods receipt note.

Type str

date

Date of GRN creation.

Type DateField

grnType

Type of GRN (auto or manual).

Type str**amendNumber**

Amendment number of the GRN.

Type int**amendDate**

Amendment date.

Type DateField**transporter**

Name of the transport/shipping service.

Type str**vehicleNumber**

Vehicle number using which the products are shipped.

Type str**gateInwardNumber**

Gate inward number of the vehicle.

Type str**preparedBy**

Name/identifier of the person who created the goods receipt note.

Type str**checkedBy**

Name/identifier of the person who validated the goods receipt note.

Type str**checkedBy**

Name/identifier of the person who inspected the physical products in the goods receipt note.

Type str**approvedBy**

Name/identifier of the authority who approved the goods receipt note.

Type str**prefix** = grn**context****product_choices****vendor_choices****vendor****TYPE_CHOICES** = [['manual', 'Blank'], ['auto', 'PO Reference']]**poRef****identifier****date**


```

    grnType
    amendNumber
    amendDate
    transporter
    vehicleNumber
    gateInwardNumber
    preparedBy
    checkedBy
    inspectedBy
    approvedBy
class InvManage.forms.GRNEEntryForm
    Bases: django.forms.Form
    Form for goods receipt note entry (GRNE).
    product
        Product associated with the goods receipt note entry.
        Type Product
    grne_id
        Unique identifier of the GRNE.
        Type int
    ppe_id
        Unique identifier of the product purchase entry associated with the GRNE.
        Type int
    quantity
        Ordered quantity of product with reference to product purchase entry.
        Type int
    remark
        Remarks of the quality engineer or the GRN creator about status of products received.
        Type str
    receivedQty
        Quantity of product received against the ordered quantity.
        Type str
    acceptedQty
        Quantity of product accepted as OK.
        Type str
    rejectedQty
        Quantity of product rejected (not OK, on HOLD, extra delivery, etc.)
        Type str
    prefix = form
    context

```

product
grne_id
ppe_id
quantity
remark
receivedQty
acceptedQty
rejectedQty

class InvManage.forms.**ThumbnailForm**

Bases: django.forms.Form

For for product thumbnail image.

image

Image of the product.

Type ImageField

prefix = thumbnail

image

class InvManage.forms.**SalesOrderBasicInfo**

Bases: django.forms.Form

Form for basic information on sales order.

consumer

Consumer associated with the sales order.

Type *Consumer*

date

Date of the sales order creation.

Type DateField

so

Sales order number.

Type int

discount

Percentage discount associated with the sales order.

Type float

tax

Percentage of tax applicable.

Type float

paid

Amount received from the consumer.

Type float

balance

Amount balance with the consumer.

Type float

subtotal

Total of all the product sales entries associated with the sales order.

Type float

taxtotal

Total tax applicable on the *subtotal*.

Type float

ordertotal

Total price of the sales order including *taxtotal*.

Type float

prefix = so

context

product_choices

consumer_choices

consumer

date

so

discount

tax

paid

balance

subtotal

taxtotal

ordertotal

class InvManage.forms.ConsumerForm

Bases: django.forms.Form

Form for consumer.

name

Name of the consumer.

Type str

identifier

Unique identifier of the consumer.

Type str

gstin

GSTIN number of the consumer.

Type str

phone

Contact number.

Type str

address

Address of the consumer.

Type str**email**

E-mail address of the consumer.

Type str**location**

City of the consumer.

Type str**prefix = consumer****context****name****identifier****gstn****phone****address****email****location****class** InvManage.forms.CompanyForm

Bases: django.forms.Form

Form for company.

name

Name of the company.

Type str**owner**

Name of the owner of the company.

Type str**gstn**

GSTIN number of the company.

Type str**phone**

Contact number of the company.

Type str**address**

Postal address.

Type str**email**

E-mail address of the contact person.

Type str

```

    location
        City of the company.
        Type str
    prefix = comp
    context
    name
    owner
    gstin
    phone
    address
    email
    location
class InvManage.forms.HistoryForm
    Bases: django.forms.Form
    Form for history view.
    qlen
        Count of the events visible on the history view.
        Type int
    prefix = history
    context
    qlen
class InvManage.forms.ShippingAddressForm
    Bases: django.forms.Form
    For for shipping address.
    title
        Title of the firm.
        Type str
    name
        Name of the firm.
        Type str
    phone
        Contact number of the firm.
        Type str
    address
        Postal address.
        Type str
    city
        City.
        Type str

```

state

State.

Type str

country

Country.

Type str

website

Official website of the firm.

Type URLField

post

Postal code of the firm.

Type str

prefix = ship

context

title

name

phone

address

city

state

country

website

post

class InvManage.forms.**VendorForm**

Bases: django.forms.Form

Form for basic informaion of the vendor.

name

Name of the vendor.

Type str

identifier

Unique identifier of the vendor.

Type str

gsttin

GSTIN number of the vendor.

Type str

prefix = vend

context

name

identifier

gstn

class InvManage.forms.CommunicationForm

Bases: django.forms.Form

Form for communication information.

language

Language of communication.

Type str

phone

Contact number of the firm.

Type str

email

E-mail of the contact person.

Type EmailField

fax

Fax number.

Type str

prefix = com

context

language

phone

email

fax

class InvManage.forms.PurchaseDataForm

Bases: django.forms.Form

Form for purchase related data.

currency

Currency of purchase.

Type str

minorder

Minimum order quantity.

Type int

contactperson

Name of the contact person.

Type str

refcode

Reference code.

Type str

transportmode

Mode of transport of the shipment.

Type str

```
    prefix = pdform
    context
    currency
    minorder
    contactperson
    refcode
    transportmode

class InvManage.forms.BankAccountForm
    Bases: django.forms.Form
    Form for bank account details.

    name
        Name of the bank.
        Type str

    branch
        Branch number of the bank.
        Type str

    region
        City in which the branch is located.
        Type str

    route
        Transit number.
        Type str

    number
        Bank account number.
        Type int

    acctype
        Type of bank account.
        Type str

    iban
        IBAN number.
        Type str

    code
        Bank code.
        Type str

    branchcode
        Branch code.
        Type str

    prefix = bank
    context
```


name
branch
region
route
number
acctype
iban
code
branchcode

3.1.1.2 InvManage.models

3.1.1.2.1 Submodules

3.1.1.2.1.1 InvManage.models.misc

3.1.1.2.1.2 Module Contents

3.1.1.2.1.3 Classes

<i>Object</i>	Model for a generic object.
<i>EventType</i>	Model for event type.
<i>ObjectModel</i>	Model associated with an object.
<i>EventCard</i>	Model for event card to be displayed in the history view.
<i>HistoryFilterState</i>	Model of filter state of the history.

class InvManage.models.misc.**Object**

Bases: django.db.models.Model

Model for a generic object.

identifier

Unique identifier of the object.

Type str

name

Name of the object.

Type str

model

Model associated with the object.

Type str

identifier

name

model

class `InvManage.models.misc.EventType`

Bases: `django.db.models.Model`

Model for event type.

name

Name of the event being triggered.

Type `str`

label

Label associated with the event.

Type `str`

Returns Returns the string representation of the label when the event type is queried.

Return type `Label`

name

label

__str__(*self*)

class `InvManage.models.misc.ObjectModel`

Bases: `django.db.models.Model`

Model associated with an object.

name

Name of the object.

Type `str`

label

Label string of the object.

Type `str`

modName

Name of the model associated with the object.

Type `str`

Returns Returns string representation of the label when object model is queried.

Return type `Label`

name

label

modName

__str__(*self*)

class `InvManage.models.misc.EventCard`

Bases: `django.db.models.Model`

Model for event card to be displayed in the history view.

obj

Object associated with the event.

Type `OneToOneField`

objId
Unique identifier of the object.

Type str

objname
Name of the object.

Type str

objmodel
Model of the object.

Type str

date
Timestamp of the event.

Type DateField

operation
Type of event.

Type str

obj

objId

objname

objmodel

date

operation

class InvManage.models.misc.HistoryFilterState

Bases: django.db.models.Model

Model of filter state of the history.

params

Paramameters of the configured filters.

Type str

params

3.1.1.2.1.4 InvManage.models.objects

3.1.1.2.1.5 Module Contents

3.1.1.2.1.6 Classes

<i>Dashboard</i>	Model of the dashboard.
<i>Company</i>	Model of the company.
<i>Vendor</i>	Model of the vendor.
<i>Consumer</i>	Model of the consumer.
<i>PurchaseOrder</i>	Model of the purchase order.

continues on next page

Table 5 – continued from previous page

<i>GoodsReceiptNote</i>	Model of the goods receipt note (GRN).
<i>SalesOrder</i>	Model of the sales order.
<i>Product</i>	Model of the product.
<i>ProductPurchaseEntry</i>	Model of the product purchase entry.
<i>GRNEntry</i>	Model of the goods receipt note entry (GRNE).
<i>ProductSalesEntry</i>	Model of the product sales entry.
<i>PurchaseInvoice</i>	Model for the purchase invoice.
<i>GRNInvoice</i>	Model for the goods receipt note invoice.
<i>SalesInvoice</i>	Model for the sales order invoice.

class `InvManage.models.objects.Dashboard`

Bases: `django.db.models.Model`

Model of the dashboard.

class `InvManage.models.objects.Company`

Bases: `django.db.models.Model`

Model of the company.

name

Name of the company.

Type `str`

owner

Owner of the company.

Type `str`

gsttin

GSTIN number of the company.

Type `str`

phone

Contact number.

Type `str`

address

Postal address.

Type `str`

email

E-mail address of the contact person.

Type `str`

location

City of the company.

Type `str`

image

Photo/logo of the company.

Type `ImageField`

shippingaddress

Primary key of the `ShippingAddress` instance associated with the company.

Type *ShippingAddress*

Returns String representation of the company name.

Return type str

name

owner

gstn

phone

address

email

location

image

shippingaddress

__str__(self)

class InvManage.models.objects.**Vendor**

Bases: django.db.models.Model

Model of the vendor.

name

Name of the vendor.

Type str

identifier

Unique identifier of the vendor.

Type str

gstn

GSTIN number of the vendor.

Type str

address

Primary key of the ShippingAddress instance associated with the vendor.

Type *ShippingAddress*

communication

Primary key of the Communication instance associated with the vendor.

Type *Communication*

backaccount

Primary key of the BankAccount instance associated with the vendor.

Type *BankAccount*

purchasedata

Primary key of the PurchaseData instance associated with the vendor.

Type *PurchaseData*

Returns String representation of the vendor name.

Return type str

name

identifier

gstn

address

communication

bankaccount

purchasedata

__str__(*self*)

class InvManage.models.objects.**Consumer**

Bases: django.db.models.Model

Model of the consumer.

name

Name of the consumer.

Type str

identifier

Unique identifier of the consumer.

Type str

gstn

GSTIN number of the consumer.

Type str

phone

Contact number.

Type str

address

Address of the consumer.

Type str

email

E-mail address of the consumer.

Type str

location

City of the consumer.

Type str

Returns [description]

Return type [type]

name

identifier

gstn

```

    phone
    address
    email
    location
    __str__(self)

class InvManage.models.objects.PurchaseOrder
    Bases: django.db.models.Model
    Model of the purchase order.
    vendor
        Vendor associated with the purchase order.
        Type Vendor
    date
        Date and time of the purchase order creation.
        Type DateTimeField
    po
        Purchase order number.
        Type int
    discount
        Percentage discount.
        Type float
    tax
        Percentage of tax applicable for the purchase.
        Type float
    paid
        Amount paid against the PO.
        Type float
    balance
        Amount balance which remains to be paid.
        Type float
    subtotal
        Total of all the product purchase entries associated with the purchase order.
        Type float
    taxtotal
        Total tax applicable on the subtotal.
        Type float
    ordertotal
        Total price of the purchase order including taxtotal.
        Type float
    is_complete()
        Returns the completion status (boolean).

```

pending_ppes()

Returns list of product purchase entries which are not completed.

vendor

date

po

discount

tax

paid

balance

subtotal

taxtotal

ordertotal

is_complete(self)

pending_ppes(self)

class InvManage.models.objects.**GoodsReceiptNote**

Bases: django.db.models.Model

Model of the goods receipt note (GRN).

vendor

Vendor associated with the goods receipt note.

Type ModelChoiceField

poRef

List of identifiers of the purchase orders from which the goods receipt note is derived.

Type MultipleChoiceField

identifier

Unique identifier of the goods receipt note.

Type str

date

Date of GRN creation.

Type DateField

grnType

Type of GRN (auto or manual).

Type ChoiceField

amendNumber

Amendment number of the GRN.

Type int

amendDate

Amendment date.

Type DateField

transporter

Name of the transport/shipping service.

Type str

vehicleNumber

Vehicle number using which the products are shipped.

Type str

gateInwardNumber

Gate inward number of the vehicle.

Type str

preparedBy

Name/identifier of the person who created the goods receipt note.

Type str

checkedBy

Name/identifier of the person who validated the goods receipt note.

Type str

checkedBy

Name/identifier of the person who inspected the physical products in the goods receipt note.

Type str

approvedBy

Name/identifier of the authority who approved the goods receipt note.

Type str

TYPE_CHOICES = [['manual', 'Blank'], ['auto', 'PO Reference']]

vendor

poRef

identifier

date

grnType

amendNumber

amendDate

transporter

vehicleNumber

gateInwardNumber

preparedBy

checkedBy

inspectedBy

approvedBy

class InvManage.models.objects.SalesOrder

Bases: django.db.models.Model

Model of the sales order.

consumer

Consumer associated with the sales order.

Type ModelChoiceField

date

Date of the sales order creation.

Type DateField

so

Sales order number.

Type int

discount

Percentage discount associated with the sales order.

Type float

tax

Percentage of tax applicable.

Type float

paid

Amount received from the consumer.

Type float

balance

Amount balance with the consumer.

Type float

subtotal

Total of all the product sales entries associated with the sales order.

Type float

taxtotal

Total tax applicable on the *subtotal*.

Type float

ordertotal

Total price of the sales order including *taxtotal*.

Type float

consumer**date****so****discount****tax****paid****balance****subtotal****taxtotal**

ordertotal**class** InvManage.models.objects.Product

Bases: django.db.models.Model

Model of the product.

name

Name of the product.

Type str**category**

Product category.

Type str**item_type**

Type of the product.

Type str**description**

Short description of the product.

Type str**price**

Price of the product.

Type float**quantity**

Stock quantity of the product.

Type int**identifier**

Unique identifier of the product.

Type str**location**

Physical location of the product.

Type str**length**

Length of the product.

Type str**width**

Width of the product.

Type str**height**

Height of the product.

Type str**weight**

Weight of the product.

Type str

discount

Default discount percentage on the product.

Type float

barcode

Barcode of the product.

Type str

expiry

Expiry date of the product.

Type DateField

image

Image of the product.

Type ImageField

name**category****item_type****description****price****quantity****identifier****location****length****width****height****weight****discount****barcode****expiry****image**

__str__(*self*)

class InvManage.models.objects.**ProductPurchaseEntry**

Bases: django.db.models.Model

Model of the product purchase entry.

product

Product associated wit the product purchase entry.

Type ModelChoiceField

quantity

Quantity of the product to be ordered.

Type int

price

Price of the product.

Type float

discount

Percentage discount on the product purchase.

Type float

order

Referenced purchase order.

Type *PurchaseOrder*

receivedQty

Quantity of product received against the ordered quantity.

Type str

acceptedQty

Quantity of product accepted as OK.

Type str

rejectedQty

Quantity of product rejected (not OK, on HOLD, extra delivery, etc.)

Type str

is_complete()

Returns the completion status (boolean).

pending_quantity()

Returns the pending quantity against the entry (int).

product**quantity****price****discount****order****receivedQty****acceptedQty****rejectedQty****is_complete(self)****property pending_quantity(self)****class** InvManage.models.objects.**GRNEntry**

Bases: django.db.models.Model

Model of the goods receipt note entry (GRNE).

product

Primary key of the Product instance associated with the goods receipt note entry.

Type *Product*

quantity

Ordered quantity of product with reference to product purchase entry.

Type int

grne

Primary key of the GoodsReceiptNote instance.

Type int

ppe

Primary key of the ProductPurchaseEntry instance associated with the GRNE.

Type int

remark

Remarks of the quality engineer or the GRN creator about status of products received.

Type str

receivedQty

Quantity of product received against the ordered quantity.

Type str

acceptedQty

Quantity of product accepted as OK.

Type str

rejectedQty

Quantity of product rejected (not OK, on HOLD, extra delivery, etc.)

Type str

product

quantity

grn

ppes

remark

receivedQty

acceptedQty

rejectedQty

class InvManage.models.objects.**ProductSalesEntry**

Bases: django.db.models.Model

Model of the product sales entry.

product

Primary key of the Product associated with the product sales entry.

Type *Product*

quantity

Quantity of the product.

Type int

price

Price of the product.

Type float

discount

Percentage discount on the product.

Type float

order

Primary key of the SalesOrder referenced by the entry.

Type *SalesOrder*

product**quantity****price****discount****order**

```
class InvManage.models.objects.PurchaseInvoice
```

Bases: django.db.models.Model

Model for the purchase invoice.

company

Primary key of the Company.

Type *Company*

po

Primary key of the PurchaseOrder.

Type *PurchaseOrder*

shippingaddress

Primary key of the ShippingAddress.

Type *ShippingAddress*

communication

Primary key of the Communication.

Type *Communication*

company**po****shippingaddress****communication**

```
class InvManage.models.objects.GRNInvoice
```

Bases: django.db.models.Model

Model for the goods receipt note invoice.

company

Primary key of the Company.

Type *Company*

grn

Primary key of the GoodsReceiptNote.

Type *GoodsReceiptNote*

shippingaddress

Primary key of the ShippingAddress.

Type *ShippingAddress***communication**

Primary key of the Communication.

Type *Communication***company****grn****shippingaddress****communication****class** `InvManage.models.objects.SalesInvoice`Bases: `django.db.models.Model`

Model for the sales order invoice.

company

Primary key of the Company.

Type *Company***so**

Primary key of the SalesOrder.

Type *SalesOrder***shippingaddress**

Primary key of the ShippingAddress.

Type *ShippingAddress***company****so****shippingaddress****3.1.1.2.1.7** `InvManage.models.reuse`**3.1.1.2.1.8** **Module Contents****3.1.1.2.1.9** **Classes**

<i>ShippingAddress</i>	Model of the shipping address.
<i>Communication</i>	Model of the communication information.
<i>PurchaseData</i>	Model of the purchase data.
<i>BankAccount</i>	Model of the bank account details.

class `InvManage.models.reuse.ShippingAddress`Bases: `django.db.models.Model`

Model of the shipping address.

title

Title of the firm.

Type str

name

Name of the firm.

Type str

phone

Contact number of the firm.

Type str

address

Postal address.

Type str

city

City.

Type str

state

State.

Type str

country

Country.

Type str

website

Official website of the firm.

Type URLField

post

Postal code of the firm.

Type str

title

name

phone

address

city

state

country

website

post

class InvManage.models.reuse.**Communication**

Bases: django.db.models.Model

Model of the communication information.

language

Language of communication.

Type str**phone**

Contact number of the firm.

Type str**email**

E-mail of the contact person.

Type EmailField**fax**

Fax number.

Type str**language****phone****email****fax****class** InvManage.models.reuse.PurchaseData

Bases: django.db.models.Model

Model of the purchase data.

currency

Currency of purchase.

Type str**minorder**

Minimum order quantity.

Type int**contactperson**

Name of the contact person.

Type str**refcode**

Reference code.

Type str**transportmode**

Mode of transport of the shipment.

Type str**currency****minorder****contactperson****refcode****transportmode**

```
class InvManage.models.reuse.BankAccount
```

```
    Bases: django.db.models.Model
```

```
    Model of the bank account details.
```

```
    name
```

```
        Name of the bank.
```

```
        Type str
```

```
    branch
```

```
        Branch number of the bank.
```

```
        Type str
```

```
    region
```

```
        City in which the branch is located.
```

```
        Type str
```

```
    route
```

```
        Transit number.
```

```
        Type str
```

```
    number
```

```
        Bank account number.
```

```
        Type int
```

```
    acctype
```

```
        Type of bank account.
```

```
        Type str
```

```
    iban
```

```
        IBAN number.
```

```
        Type str
```

```
    code
```

```
        Bank code.
```

```
        Type str
```

```
    branchcode
```

```
        Branch code.
```

```
        Type str
```

```
    name
```

```
    branch
```

```
    region
```

```
    route
```

```
    number
```

```
    acctype
```

```
    iban
```

```
    code
```

```
    branchcode
```

3.1.1.2.1.10 InvManage.models.states

3.1.1.2.1.11 Module Contents

3.1.1.2.1.12 Classes

<i>FilterState</i>	Model of the filter state.
<i>FilterColumn</i>	Model of the column state.

class InvManage.models.states.**FilterState**

Bases: django.db.models.Model

Model of the filter state.

name

Name of the filter state.

Type str

name

class InvManage.models.states.**FilterColumn**

Bases: django.db.models.Model

Model of the column state.

name

Name of the column.

Type str

label

Label of the column.

Type str

visible

Visibility flag.

Type bool

position

Index of the column.

Type int

state

Primary key of the **FilterState** instance being referenced by the column instance.

Type *FilterState*

name

label

visible

position

state

3.1.1.2.2 Package Contents

3.1.1.2.2.1 Classes

<i>Product</i>	Model of the product.
<i>ProductPurchaseEntry</i>	Model of the product purchase entry.
<i>GRNEntry</i>	Model of the goods receipt note entry (GRNE).
<i>ProductSalesEntry</i>	Model of the product sales entry.
<i>SalesOrder</i>	Model of the sales order.
<i>PurchaseOrder</i>	Model of the purchase order.
<i>GoodsReceiptNote</i>	Model of the goods receipt note (GRN).
<i>Consumer</i>	Model of the consumer.
<i>Vendor</i>	Model of the vendor.
<i>Company</i>	Model of the company.
<i>PurchaseInvoice</i>	Model for the purchase invoice.
<i>SalesInvoice</i>	Model for the sales order invoice.
<i>FilterColumn</i>	Model of the column state.
<i>FilterState</i>	Model of the filter state.
<i>Object</i>	Model for a generic object.
<i>EventCard</i>	Model for event card to be displayed in the history view.
<i>EventType</i>	Model for event type.
<i>ObjectModel</i>	Model associated with an object.
<i>HistoryFilterState</i>	Model of filter state of the history.
<i>ShippingAddress</i>	Model of the shipping address.
<i>Communication</i>	Model of the communication information.
<i>PurchaseData</i>	Model of the purchase data.
<i>BankAccount</i>	Model of the bank account details.

```

class InvManage.models.Product
    Bases: django.db.models.Model

    Model of the product.

    name
        Name of the product.
        Type str

    category
        Product category.
        Type str

    item_type
        Type of the product.
        Type str

    description
        Short description of the product.
        Type str

    price
        Price of the product.
        Type float

```

quantity

Stock quantity of the product.

Type int

identifier

Unique identifier of the product.

Type str

location

Physical location of the product.

Type str

length

Length of the product.

Type str

width

Width of the product.

Type str

height

Height of the product.

Type str

weight

Weight of the product.

Type str

discount

Default discount percentage on the product.

Type float

barcode

Barcode of the product.

Type str

expiry

Expiry date of the product.

Type DateField

image

Image of the product.

Type ImageField

name**category****item_type****description****price****quantity****identifier**

location
length
width
height
weight
discount
barcode
expiry
image
__str__(*self*)

class InvManage.models.**ProductPurchaseEntry**

Bases: django.db.models.Model

Model of the product purchase entry.

product

Product associated wit the product purchase entry.

Type ModelChoiceField

quantity

Quantity of the product to be ordered.

Type int

price

Price of the product.

Type float

discount

Percentage discount on the product purchase.

Type float

order

Referenced purchase order.

Type *PurchaseOrder*

receivedQty

Quantity of product received against the ordered quantity.

Type str

acceptedQty

Quantity of product accepted as OK.

Type str

rejectedQty

Quantity of product rejected (not OK, on HOLD, extra delivery, etc.)

Type str

is_complete()

Returns the completion status (boolean).

pending_quantity()

Returns the pending quantity against the entry (int).

product

quantity

price

discount

order

receivedQty

acceptedQty

rejectedQty

is_complete(self)

property pending_quantity(self)

class InvManage.models.GRNEntry

Bases: django.db.models.Model

Model of the goods receipt note entry (GRNE).

product

Primary key of the Product instance associated with the goods receipt note entry.

Type *Product*

quantity

Ordered quantity of product with reference to product purchase entry.

Type int

grne

Primary key of the GoodsReceiptNote instance.

Type int

ppe

Primary key of the ProductPurchaseEntry instance associated with the GRNE.

Type int

remark

Remarks of the quality engineer or the GRN creator about status of products received.

Type str

receivedQty

Quantity of product received against the ordered quantity.

Type str

acceptedQty

Quantity of product accepted as OK.

Type str

rejectedQty

Quantity of product rejected (not OK, on HOLD, extra delivery, etc.)

Type str

product
quantity
grn
ppes
remark
receivedQty
acceptedQty
rejectedQty

class InvManage.models.ProductSalesEntry

Bases: django.db.models.Model

Model of the product sales entry.

product
 Primary key of the Product associated with the product sales entry.

Type *Product*

quantity
 Quantity of the product.

Type int

price
 Price of the product.

Type float

discount
 Percentage discount on the product.

Type float

order
 Primary key of the SalesOrder referenced by the entry.

Type *SalesOrder*

product
quantity
price
discount
order

class InvManage.models.SalesOrder

Bases: django.db.models.Model

Model of the sales order.

consumer
 Consumer associated with the sales order.

Type ModelChoiceField

date
 Date of the sales order creation.

Type DateField

so

Sales order number.

Type int

discount

Percentage discount associated with the sales order.

Type float

tax

Percentage of tax applicable.

Type float

paid

Amount received from the consumer.

Type float

balance

Amount balance with the consumer.

Type float

subtotal

Total of all the product sales entries associated with the sales order.

Type float

taxtotal

Total tax applicable on the *subtotal*.

Type float

ordertotal

Total price of the sales order including *taxtotal*.

Type float

consumer

date

so

discount

tax

paid

balance

subtotal

taxtotal

ordertotal

class InvManage.models.PurchaseOrder

Bases: django.db.models.Model

Model of the purchase order.

vendor

Vendor associated with the purchase order.

Type *Vendor*

date

Date and time of the purchase order creation.

Type *DateField*

po

Purchase order number.

Type *int*

discount

Percentage discount.

Type *float*

tax

Percentage of tax applicable for the purchase.

Type *float*

paid

Amount paid against the PO.

Type *float*

balance

Amount balance which remains to be paid.

Type *float*

subtotal

Total of all the product purchase entries associated with the purchase order.

Type *float*

taxtotal

Total tax applicable on the *subtotal*.

Type *float*

ordertotal

Total price of the purchase order including *taxtotal*.

Type *float*

is_complete()

Returns the completion status (boolean).

pending_ppes()

Returns list of product purchase entries which are not completed.

vendor**date****po****discount****tax****paid**

balance

subtotal

taxtotal

ordertotal

is_complete(*self*)

pending_ppes(*self*)

class InvManage.models.GoodsReceiptNote

Bases: django.db.models.Model

Model of the goods receipt note (GRN).

vendor

Vendor associated with the goods receipt note.

Type ModelChoiceField

poRef

List of identifiers of the purchase orders from which the goods receipt note is derived.

Type MultipleChoiceField

identifier

Unique identifier of the goods receipt note.

Type str

date

Date of GRN creation.

Type DateField

grnType

Type of GRN (auto or manual).

Type ChoiceField

amendNumber

Amendment number of the GRN.

Type int

amendDate

Amendment date.

Type DateField

transporter

Name of the transport/shipping service.

Type str

vehicleNumber

Vehicle number using which the products are shipped.

Type str

gateInwardNumber

Gate inward number of the vehicle.

Type str

preparedBy

Name/identifier of the person who created the goods receipt note.

Type str

checkedBy

Name/identifier of the person who validated the goods receipt note.

Type str

checkedBy

Name/identifier of the person who inspected the physical products in the goods receipt note.

Type str

approvedBy

Name/identifier of the authority who approved the goods receipt note.

Type str

TYPE_CHOICES = [['manual', 'Blank'], ['auto', 'PO Reference']]

vendor

poRef

identifier

date

grnType

amendNumber

amendDate

transporter

vehicleNumber

gateInwardNumber

preparedBy

checkedBy

inspectedBy

approvedBy

class InvManage.models.Consumer

Bases: django.db.models.Model

Model of the consumer.

name

Name of the consumer.

Type str

identifier

Unique identifier of the consumer.

Type str

gstIn

GSTIN number of the consumer.

Type str

phone

Contact number.

Type str**address**

Address of the consumer.

Type str**email**

E-mail address of the consumer.

Type str**location**

City of the consumer.

Type str**Returns** [description]**Return type** [type]**name****identifier****gstin****phone****address****email****location****__str__**(*self*)**class** InvManage.models.**Vendor**

Bases: django.db.models.Model

Model of the vendor.

name

Name of the vendor.

Type str**identifier**

Unique identifier of the vendor.

Type str**gstin**

GSTIN number of the vendor.

Type str**address**

Primary key of the ShippingAddress instance associated with the vendor.

Type *ShippingAddress***communication**

Primary key of the Communication instance associated with the vendor.

Type *Communication*

backaccount

Primary key of the BankAccount instance associated with the vendor.

Type *BankAccount*

purchasedata

Primary key of the PurchaseData instance associated with the vendor.

Type *PurchaseData*

Returns String representation of the vendor name.

Return type str

name

identifier

gstn

address

communication

bankaccount

purchasedata

__str__(self)

class InvManage.models.Company

Bases: django.db.models.Model

Model of the company.

name

Name of the company.

Type str

owner

Owner of the company.

Type str

gstn

GSTIN number of the company.

Type str

phone

Contact number.

Type str

address

Postal address.

Type str

email

E-mail address of the contact person.

Type str

location

City of the company.

Type str

image

Photo/logo of the company.

Type ImageField

shippingaddress

Primary key of the ShippingAddress instance associated with the company.

Type *ShippingAddress*

Returns String representation of the company name.

Return type str

name**owner****gsttin****phone****address****email****location****image****shippingaddress****__str__**(*self*)**class** InvManage.models.**PurchaseInvoice**

Bases: django.db.models.Model

Model for the purchase invoice.

company

Primary key of the Company.

Type *Company*

po

Primary key of the PurchaseOrder.

Type *PurchaseOrder*

shippingaddress

Primary key of the ShippingAddress.

Type *ShippingAddress*

communication

Primary key of the Communication.

Type *Communication*

company**po**

shippingaddress**communication****class** InvManage.models.SalesInvoice

Bases: django.db.models.Model

Model for the sales order invoice.

company

Primary key of the Company.

Type *Company***so**

Primary key of the SalesOrder.

Type *SalesOrder***shippingaddress**

Primary key of the ShippingAddress.

Type *ShippingAddress***company****so****shippingaddress****class** InvManage.models.FilterColumn

Bases: django.db.models.Model

Model of the column state.

name

Name of the column.

Type str**label**

Label of the column.

Type str**visible**

Visibility flag.

Type bool**position**

Index of the column.

Type int**state**

Primary key of the FilterState instance being referenced by the column instance.

Type *FilterState***name****label****visible****position**

state

class InvManage.models.FilterState

Bases: django.db.models.Model

Model of the filter state.

name

Name of the filter state.

Type str

name

class InvManage.models.Object

Bases: django.db.models.Model

Model for a generic object.

identifier

Unique identifier of the object.

Type str

name

Name of the object.

Type str

model

Model associated with the object.

Type str

identifier

name

model

class InvManage.models.EventCard

Bases: django.db.models.Model

Model for event card to be displayed in the history view.

obj

Object associated with the event.

Type OneToOneField

objId

Unique identifier of the object.

Type str

objname

Name of the object.

Type str

objmodel

Model of the object.

Type str

date

Timestamp of the event.

```

        Type DateField

operation
    Type of event.

        Type str

obj
objId
objname
objmodel
date
operation

class InvManage.models.EventType
    Bases: django.db.models.Model

    Model for event type.

    name
        Name of the event being triggered.

        Type str

    label
        Label associated with the event.

        Type str

    Returns Returns the string representation of the label when the event type is queried.

    Return type Label

    name
    label
    __str__(self)

class InvManage.models.ObjectModel
    Bases: django.db.models.Model

    Model associated with an object.

    name
        Name of the object.

        Type str

    label
        Label string of the object.

        Type str

    modName
        Name of the model associated with the object.

        Type str

    Returns Returns string representation of the label when object model is queried.

    Return type Label

```

name

label

modName

__str__(*self*)

class InvManage.models.**HistoryFilterState**

Bases: django.db.models.Model

Model of filter state of the history.

params

Paramameters of the configured filters.

Type str

params

class InvManage.models.**ShippingAddress**

Bases: django.db.models.Model

Model of the shipping address.

title

Title of the firm.

Type str

name

Name of the firm.

Type str

phone

Contact number of the firm.

Type str

address

Postal address.

Type str

city

City.

Type str

state

State.

Type str

country

Country.

Type str

website

Official website of the firm.

Type URLField

post

Postal code of the firm.

```

        Type str

    title
    name
    phone
    address
    city
    state
    country
    website
    post
class InvManage.models.Communication
    Bases: django.db.models.Model
    Model of the communication information.
    language
        Language of communication.
        Type str
    phone
        Contact number of the firm.
        Type str
    email
        E-mail of the contact person.
        Type EmailField
    fax
        Fax number.
        Type str
    language
    phone
    email
    fax
class InvManage.models.PurchaseData
    Bases: django.db.models.Model
    Model of the purchase data.
    currency
        Currency of purchase.
        Type str
    minorder
        Minimum order quantity.
        Type int

```

contactperson

Name of the contact person.

Type str**refcode**

Reference code.

Type str**transportmode**

Mode of transport of the shipment.

Type str**currency****minorder****contactperson****refcode****transportmode****class** InvManage.models.**BankAccount**

Bases: django.db.models.Model

Model of the bank account details.

name

Name of the bank.

Type str**branch**

Branch number of the bank.

Type str**region**

City in which the branch is located.

Type str**route**

Transit number.

Type str**number**

Bank account number.

Type int**accttype**

Type of bank account.

Type str**iban**

IBAN number.

Type str**code**

Bank code.

Type str

branchcode
Branch code.

Type str

name

branch

region

route

number

accttype

iban

code

branchcode

3.1.1.3 InvManage.scripts

3.1.1.3.1 Submodules

3.1.1.3.1.1 InvManage.scripts.filters

3.1.1.3.1.2 Module Contents

3.1.1.3.1.3 Functions

<i>get_columns</i> (state)	Gets the columns of the display table.
<i>sort_ascending_descending</i> (request, model)	Sorts and saves the state of the column.
<i>change_column_position</i> (request, state)	Modifies the column states depending on the request.
<i>paginate</i> (queryset, filter, page_number)	Paginates the list of objects.

InvManage.scripts.filters.get_columns(state)

Gets the columns of the display table.

This function takes a filter state and returns an array of column names that are ordered depending on the position attribute of the columns. This order is important as the queryset will be rendered in the display table according to the column order in this returned array.

Parameters **state** ([FilterState](#)) – Filter state of the columns.

Returns A list that dictates the order of the columns.

Return type List

InvManage.scripts.filters.sort_ascending_descending(request, model)

Sorts and saves the state of the column.

This function takes a request, finds which column needs to be sorted in ascending/descending order, and returns the sorted queryset.

Parameters

- **request** (*HttpRequest*) – GET request received for retrieving the list of model instances.
- **model** (*Model*) – Model of the objects being retrieved.

Returns List of sorted object instances.

Return type QuerySet

`InvManage.scripts.filters.change_column_position(request, state)`

Modifies the column states depending on the request.

This function:

- Takes a filter state
- Extracts the column names into an order array
- Modifies the column order in the array depending on user input direction (left/right)
- Saves this modified state of columns into the database
- Returns the modified column names array

Parameters

- **request** (*HttpRequest*) – GET request for retrieving the state of the columns.
- **state** (*FilterState*) – Filter state of the columns.

Returns List of sorted columns.

Return type List

`InvManage.scripts.filters.paginate(queryset, filter, page_number)`

Paginates the list of objects.

This function takes the entire queryset and filters out only objects belonging to the request page

Parameters

- **queryset** (*QuerySet*) – QuerySet of the objects retrieved from the database.
- **filter** (*FilterState*) – Filter state of the columns.
- **page_number** (*int*) – Page number requested.

3.1.1.4 `InvManage.views`

3.1.1.4.1 Submodules

3.1.1.4.1.1 `InvManage.views.company_views`

3.1.1.4.1.2 Module Contents

3.1.1.4.1.3 Functions

<code>create_company_view(request)</code>	Creates a company on POST request and returns a company creation form GET request.
<code>update_company_view(request)</code>	Updates a company on POST request and returns the company update form GET request.

continues on next page

Table 10 – continued from previous page

<code>delete_company_view</code> (request, pk)	Deletes the company with primary key pk on POST request.
<code>display_companies_view</code> (request)	Retrieves the list of companies on GET request.

`InvManage.views.company_views.create_company_view(request)`

Creates a company on POST request and returns a company creation form GET request.

GET /company

Gets the company creation form.

Example request:

```
GET /company/ HTTP/1.1
Host: localhost:8000
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: text/html; charset=utf-8
```

Request Headers

- **Accept** – text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9

Status Codes

- **200 OK** – Company creation form received successfully.

POST /company

Creates a company.

Example request:

```
POST /company/ HTTP/1.1
Host: localhost:8000
Content-Type: multipart/form-data;
```

Form Parameters

- **comp-name** – Fringillami
- **comp-owner** – Ivor Barnett
- **comp-gstin** – 89AAC056465468
- **comp-phone** – 332 220-7026
- **comp-address** – Ap #849-6241 Euismod Av., 677598, Carinthia, Belgium
- **comp-email** – est.tempor@fringillami.org
- **comp-location** – Belgium
- **ship-title** – FingDocks
- **ship-name** – Harding Gross

- **ship-phone** – 936 651-4847
- **ship-address** – 8798 At, St., 7639
- **ship-city** – Rome
- **ship-state** – Lazio
- **ship-country** – Italy
- **ship-website** – fringdocs.com
- **ship-post** – 300326

Response Headers

- **Content-Type** – multipart/form-data; boundary=—WebKitFormBoundaryLTR88aZAnBUSE7mv

Status Codes

- **302 Found** – Redirects to /company.

InvManage.views.company_views.**update_company_view**(*request*)

Updates a company on POST request and returns the company update form GET request.

GET /company/update

Gets the company update form whose primary key matches the query parameter pk.

Example request:

```
GET /company/update HTTP/1.1
Host: localhost:8000
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/
webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
```

Query Parameters

- **pk** – The primary key of the company.

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: text/html; charset=utf-8
```

Request Headers

- **Accept** – text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9

Status Codes

- **200 OK** – Company update form received successfully.

POST /company/update

Updates a company.

Example request:

```
POST /company/ HTTP/1.1
Host: localhost:8000
Content-Type: multipart/form-data;
```

Form Parameters

- **pk** – 4
- **comp-name** – Fringillami
- **comp-owner** – Ivor Barnett
- **comp-gstin** – 89AAC056465468
- **comp-phone** – 332 220-7026
- **comp-address** – Ap #849-6241 Euismod Av., 677598, Carinthia, Belgium
- **comp-email** – est.tempor@fringillami.org
- **comp-location** – Belgium
- **ship-title** – FingDocks
- **ship-name** – Harding Gross
- **ship-phone** – 936 651-4847
- **ship-address** – 8798 At, St., 7639
- **ship-city** – Rome
- **ship-state** – Lazio
- **ship-country** – Italy
- **ship-website** – fringdocs.com
- **ship-post** – 300326

Response Headers

- **Content-Type** – multipart/form-data; boundary=—WebKitFormBoundaryLTR88aZAnBUSE7mv

Status Codes

- **302 Found** – Redirects to /company.

`InvManage.views.company_views.delete_company_view(request, pk)`
 Deletes the company with primary key `pk` on POST request.

POST /company/<str:object_id>/delete

Deletes the company represented by the primary key `object_id`.

Example request:

```
POST /company/32/delete HTTP/1.1
Host: localhost:8000
Content-Type: application/x-www-form-urlencoded
```

Parameters

- **object_id** – Company primary key.

Response Headers

- **Content-Type** – application/x-www-form-urlencoded

Status Codes

- **302 Found** – Redirects to /company.

- **500 Internal Server Error** – Company matching query does not exist.

`InvManage.views.company_views.display_companies_view(request)`

Retrieves the list of companies on GET request.

GET `/companies/`

Gets the list of all companies.

Example request:

```
GET /companies/ HTTP/1.1
Host: localhost:8000
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
```

Form Parameters

- **page** – The page number of the companies list.

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: text/html; charset=utf-8
```

Request Headers

- **Accept** – text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9

Status Codes

- **200 OK** – List of companies received successfully.

3.1.1.4.1.4 `InvManage.views.consumer_views`

3.1.1.4.1.5 Module Contents

3.1.1.4.1.6 Functions

<code>create_consumer_view(request)</code>	Creates a consumer on POST request, and returns a consumer creation form on GET request.
<code>update_consumer_view(request)</code>	Updates a consumer on POST request and returns the consumer update form for GET request.
<code>delete_consumer_view(request, pk)</code>	Deletes the consumer with primary key pk on POST request.
<code>display_consumers_view(request)</code>	Retrieves the list of consumers on GET request.
<code>get_consumer(request)</code>	Returns the JSON serialized data of the requested consumer on GET request.

`InvManage.views.consumer_views.create_consumer_view(request)`

Creates a consumer on POST request, and returns a consumer creation form on GET request.

GET `/consumer`

Gets the consumer creation form.

Example request:

```
GET /consumer/ HTTP/1.1
Host: localhost:8000
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: text/html; charset=utf-8
```

Request Headers

- **Accept** – text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9

Status Codes

- **200 OK** – Consumer creation form received successfully.

POST /consumer

Creates a consumer.

Example request:

```
POST /consumer/ HTTP/1.1
Host: localhost:8000
Content-Type: multipart/form-data;
```

Form Parameters

- **consumer-name** – The Music Store
- **consumer-identifier** – CONS1256
- **consumer-gstin** – 89ACC654654335
- **consumer-phone** – +91 6543525422
- **consumer-address** – Plot no. 958, N-4, Neo Complex, Barh, Wokha, Nagaland, 797111
- **consumer-email** – johnDoe@themusic.store
- **consumer-location** – Wokha

Response Headers

- **Content-Type** – multipart/form-data; boundary=—WebKitFormBoundaryLTR88aZAnBUSE7mv

Status Codes

- **302 Found** – Redirects to /consumer.

`InvManage.views.consumer_views.update_consumer_view(request)`

Updates a consumer on POST request and returns the consumer update form for GET request.

GET /consumer/update

Gets the consumer update form whose primary key matches the query parameter pk.

Example request:

```
GET /consumer/update HTTP/1.1
Host: localhost:8000
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
```

Query Parameters

- **pk** – The primary key of the consumer.

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: text/html; charset=utf-8
```

Request Headers

- **Accept** – text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9

Status Codes

- **200 OK** – Consumer update form received successfully.

POST /consumer/update

Updates the consumer.

Example request:

```
POST /company/update HTTP/1.1
Host: localhost:8000
Content-Type: multipart/form-data;
```

Form Parameters

- **pk** – 7
- **consumer-name** – The Music Store
- **consumer-identifier** – CONS1256
- **consumer-gstin** – 89ACC654654335
- **consumer-phone** – +91 6543525422
- **consumer-address** – Plot no. 958, N-4, Neo Complex, Barh, Wokha, Nagaland, 797111
- **consumer-email** – johnDoe@themusic.store
- **consumer-location** – Wokha

Response Headers

- **Content-Type** – multipart/form-data; boundary=—WebKitFormBoundaryLTR88aZAnBUSE7mv

Status Codes

- **302 Found** – Redirects to /consumer.

InvManage.views.consumer_views.delete_consumer_view(request, pk)

Deletes the consumer with primary key pk on POST request.

POST /consumer/<str:object_id>/delete

Deletes the consumer represented by the primary key object_id.

Example request:

```
POST /consumer/5/delete HTTP/1.1
Host: localhost:8000
Content-Type: application/x-www-form-urlencoded
```

Parameters

- **object_id** – Consumer primary key.

Response Headers

- **Content-Type** – application/x-www-form-urlencoded

Status Codes

- **302 Found** – Redirects to /consumer.
- **500 Internal Server Error** – Consumer matching query does not exist.

InvManage.views.consumer_views.display_consumers_view(request)

Retrieves the list of consumers on GET request.

GET /consumers/

Gets the list of all consumers.

Example request:

```
GET /companies/ HTTP/1.1
Host: localhost:8000
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
```

Form Parameters

- **page** – The page number of the consumers list.

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: text/html; charset=utf-8
```

Request Headers

- **Accept** – text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9

Status Codes

- **200 OK** – List of consumers received successfully.

InvManage.views.consumer_views.get_consumer(request)
Returns the JSON serialized data of the requested consumer on GET request.

GET /get_consumer/
Gets the JSON serialized data of the requested consumer.

Example request:

```
GET /get_consumer/ HTTP/1.1
Host: localhost:8000
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/
webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json; charset=utf-8

[
  {
    "name": "The Music Store",
    "identifier": "CONS1256",
    "gsttin": "89AAC463353643",
    "phone": "+91 8325642358",
    "address": "Plot no 958, N- 4, Neo Complex, Barh, Wokha, Nagaland, 797111",
    "email": "JohnDoe@themusic.store",
    "location": "Wokha"
  }
]
```

Response Headers

- Content-Type – application/json

Status Codes

- 200 OK – List of consumers received successfully.
- 400 Bad Request – Bad request version
- 500 Internal Server Error – Consumer matching query does not exist.

3.1.1.4.1.7 InvManage.views.goods_receipt_note_views

3.1.1.4.1.8 Module Contents

3.1.1.4.1.9 Functions

create_grn_view(request)	Creates a goods receipt note (GRN) on POST request, and returns a GRN creation form on GET request.
update_grn_view(request)	Updates the goods receipt note (GRN) on POST request, and returns a GRN update form on GET request.

continues on next page

Table 12 – continued from previous page

<code>delete_grn_view(request, pk)</code>	Deletes the GRN with primary key <code>object_id</code> on POST request.
<code>display_grns_view(request)</code>	Retrieves the list of GRNs on GET request.
<code>print_grn_view(request, pk)</code>	Opens the GRN with primary key <code>grn_id</code> for printing on GET request.

`InvManage.views.goods_receipt_note_views.create_grn_view(request)`

Creates a goods receipt note (GRN) on POST request, and returns a GRN creation form on GET request.

GET /grn

Gets the GRN creation form.

Example request:

```
GET /grn/ HTTP/1.1
Host: localhost:8000
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: text/html; charset=utf-8
```

Request Headers

- **Accept** – text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9

Status Codes

- **200 OK** – GRN creation form received successfully.

POST /grn

Creates a GRN. There are two modes: - auto : Setting `grn-grnType` to auto creates a GRN with PO reference. - manual : Setting `grn-grnType` to manual creates a GRN without PO reference. In this case the `grn-poRef` is not required.

Example request:

```
POST /consumer/ HTTP/1.1
Host: localhost:8000
Content-Type: multipart/form-data;
```

Form Parameters

- **grn-grnType** – auto
- **grn-vendor** – 4
- **grn-poRef** – 182
- **grn-amendNumber** – 546
- **grn-amendDate** – 2021-09-29
- **grn-identifier** – 846

- **grn-date** – 2021-09-29
- **grn-transporter** – TeraTransport
- **grn-vehicleNumber** – GH-646358
- **grn-gateInwardNumber** – 864353
- **grn-preparedBy** – KJL
- **grn-checkedBy** – KJH
- **grn-inspectedBy** – GIO
- **grn-approvedBy** – BHI
- **form-TOTAL_FORMS** – 3
- **form-INITIAL_FORMS** – 0
- **form-MIN_NUM_FORMS** –
- **form-MAX_NUM_FORMS** –
- **form-0-product** – 637
- **form-0-quantity** – 100
- **form-0-receivedQty** – 50
- **form-0-acceptedQty** – 50
- **form-0-rejectedQty** – 0
- **form-0-remark** – OK
- **form-0-DELETE** –
- **form-0-grne_id** – -1
- **form-0-ppe_id** – 324
- **form-1-product** – 645
- **form-1-quantity** – 250
- **form-1-receivedQty** – 200
- **form-1-acceptedQty** – 180
- **form-1-rejectedQty** – 20
- **form-1-remark** – 20 pieces faulty
- **form-1-DELETE** –
- **form-1-grne_id** – -1
- **form-1-ppe_id** – 325
- **form-2-product** – 638
- **form-2-quantity** – 200
- **form-2-receivedQty** – 0
- **form-2-acceptedQty** – 0
- **form-2-rejectedQty** – 0
- **form-2-remark** –

- **form-2-DELETE** –
- **form-2-grne_id** – -1
- **form-2-ppe_id** – 326

Response Headers

- **Content-Type** – multipart/form-data; boundary=——WebKitFormBoundaryLTR88aZAnBUSE7mv

Status Codes

- **302 Found** – Redirects to /grn.

InvManage.views.goods_receipt_note_views.**update_grn_view**(request)

Updates the goods receipt note (GRN) on POST request, and returns a GRN update form on GET request.

GET /grn/update

Gets the GRN update form.

Example request:

```
GET /grn/update HTTP/1.1
Host: localhost:8000
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: text/html; charset=utf-8
```

Request Headers

- **Accept** – text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9

Status Codes

- **200 OK** – GRN update form received successfully.

POST /grn/update

Updates a GRN. There are two modes: - auto : Setting grn-grnType to auto creates a GRN with PO reference. - manual : Setting grn-grnType to manual creates a GRN without PO reference. In this case

the grn-poRef is not required.

Example request:

```
POST /consumer/ HTTP/1.1
Host: localhost:8000
Content-Type: multipart/form-data;
```

Form Parameters

- **grn-grnType** – auto
- **grn-vendor** – 4
- **grn-poRef** – 182

- `grn-amendNumber` – 546
- `grn-amendDate` – 2021-09-29
- `grn-identifier` – 846
- `grn-date` – 2021-09-29
- `grn-transporter` – TeraTransport
- `grn-vehicleNumber` – GH-646358
- `grn-gateInwardNumber` – 864353
- `grn-preparedBy` – KJL
- `grn-checkedBy` – KJH
- `grn-inspectedBy` – GIO
- `grn-approvedBy` – BHI
- `form-TOTAL_FORMS` – 3
- `form-INITIAL_FORMS` – 0
- `form-MIN_NUM_FORMS` –
- `form-MAX_NUM_FORMS` –
- `form-0-product` – 637
- `form-0-quantity` – 100
- `form-0-receivedQty` – 50
- `form-0-acceptedQty` – 50
- `form-0-rejectedQty` – 0
- `form-0-remark` – OK
- `form-0-DELETE` –
- `form-0-grne_id` – -1
- `form-0-ppe_id` – 324
- `form-1-product` – 645
- `form-1-quantity` – 250
- `form-1-receivedQty` – 200
- `form-1-acceptedQty` – 180
- `form-1-rejectedQty` – 20
- `form-1-remark` – 20 pieces faulty
- `form-1-DELETE` –
- `form-1-grne_id` – -1
- `form-1-ppe_id` – 325
- `form-2-product` – 638
- `form-2-quantity` – 200
- `form-2-receivedQty` – 0

- **form-2-acceptedQty** – 0
- **form-2-rejectedQty** – 0
- **form-2-remark** –
- **form-2-DELETE** –
- **form-2-grne_id** – -1
- **form-2-ppe_id** – 326

Response Headers

- **Content-Type** – multipart/form-data; boundary=—WebKitFormBoundaryLTR88aZAnBUSE7mv

Status Codes

- **302 Found** – Redirects to /grn.

`InvManage.views.goods_receipt_note_views.delete_grn_view(request, pk)`

Deletes the GRN with primary key `object_id` on POST request.

POST /grn/<str:object_id>/delete

Deletes the GRN represented by the primary key `object_id`.

Example request:

```
POST /grn/104/delete HTTP/1.1
Host: localhost:8000
Content-Type: application/x-www-form-urlencoded
```

Parameters

- **object_id** – GRN primary key.

Response Headers

- **Content-Type** – application/x-www-form-urlencoded

Status Codes

- **302 Found** – Redirects to /grn.
- **500 Internal Server Error** – GRN matching query does not exist.

`InvManage.views.goods_receipt_note_views.display_grns_view(request)`

Retrieves the list of GRNs on GET request.

GET /grns/

Gets the list of all GRNs.

Example request:

```
GET /grns/ HTTP/1.1
Host: localhost:8000
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
```

Form Parameters

- **page** – The page number of the GRN list.

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: text/html; charset=utf-8
```

Request Headers

- **Accept** – text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9

Status Codes

- **200 OK** – List of GRNs received successfully.

InvManage.views.goods_receipt_note_views.**print_grn_view**(request, pk)

Opens the GRN with primary key grn_id for printing on GET request.

POST /grn/<str:grn_id>/print

Opens the GRN represented by the primary key grn_id.

Example request:

```
POST /grn/103/print HTTP/1.1
Host: localhost:8000
Content-Type: application/x-www-form-urlencoded
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json; charset=utf-8

[
  {
    "company": {
      "name": "Fringillami",
      "owner": "Ivor Barnett",
      "gstn": "89AAC056465468",
      "phone": "332 220-7026",
      "address": "Ap #849-6241 Euismod Av., 677598, Carinthia, Belgium",
      "email": "est.tempor@fringillami.org",
      "location": "Belgium",
      "image": "/media/images/hyperlink-green_x91WW5n.png"
    },
    "grn": {
      "grnes": [
        {
          "grn": 103,
          "grne_id": 117,
          "ppe_id": 324,
          "po_id": 293,
          "product": {
            "pk": 637,
            "name": "piano",
            "category": "Ultricies PC",

```

(continues on next page)

(continued from previous page)

```

        "quantity": 23921,
        "identifier": "PROD9",
        "location": "Musselburgh",
        "description": "88-key, Tri-sensor Scaled Hammer Action_
Keyboard II, Simulated ebony and ivory keys ",
        "prod_id": 637
    },
    {
        "quantity": 100,
        "receivedQty": 50,
        "acceptedQty": 50,
        "rejectedQty": 0,
        "remark": "OK"
    },
    {
        "grn": 103,
        "grne_id": 118,
        "ppe_id": 325,
        "po_id": 293,
        "product": {
            "pk": 645,
            "name": "Tabl",
            "category": "Sociis Natoque Company",
            "quantity": 38276,
            "identifier": "PROD17",
            "location": "Schagen",
            "description": "aldgjlkjlkasdj",
            "prod_id": 645
        },
        "quantity": 250,
        "receivedQty": 200,
        "acceptedQty": 180,
        "rejectedQty": 20,
        "remark": "20 pieces faulty"
    }
],
"date": "29-Sep-2021",
"vendor": {
    "name": "Girish",
    "identifier": "GJ",
    "gstIn": "GSTIN002",
    "address": {
        "name": "alsf",
        "address": "jas;k",
        "city": ";sdalkf",
        "phone": "alsf",
        "state": "kjdfk",
        "country": "ljflkj",
        "post": "54545"
    }
},
"poRef": [
    182

```

(continues on next page)

(continued from previous page)

```

    ],
    "identifier": 846,
    "grnType": "auto",
    "amendDate": "2021-09-29T00:00:00Z",
    "transporter": "TeraTransport",
    "vehicleNumber": "GH-646358",
    "gateInwardNumber": "864353",
    "preparedBy": "KJL",
    "checkedBy": "KJH",
    "inspectedBy": "GIO",
    "approvedBy": "BHI"
  },
  "shippingaddress": {
    "name": "Harding Gross",
    "address": "8798 At, St., 7639",
    "city": "Rome",
    "phone": "936 651-4847",
    "state": "Lazio",
    "country": "Italy",
    "post": "300326"
  },
  "communication": {
    "email": "asfs@aflk.com",
    "phone": "6546432"
  }
}
]

```

Parameters

- **grn_id** – GRN primary key.

Response Headers

- **Content-Type** – application/json

Status Codes

- **200 OK** – GRN print request successful.
- **500 Internal Server Error** – GRN matching query does not exist.

3.1.1.4.1.10 InvManage.views.history_views**3.1.1.4.1.11 Module Contents****3.1.1.4.1.12 Functions**

`display_history_view(request)`

Retrieves the list of events on GET request. The create, update, and delete events are registered for each model.

InvManage.views.history_views.display_history_view(request)

Retrieves the list of events on GET request. The create, update, and delete events are registered for each model.

GET /history/

Gets the list of all history items.

Example request:

```
GET /history/ HTTP/1.1
Host: localhost:8000
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9

[
  {
    "objname": {
      ""
    },
    "operation": {
      "Created",
      "Updated",
      "Deleted"
    },
    "objmodel": {
      "Company",
      "Vendor",
      "PurchaseOrder",
      "Product",
      "Consumer",
      "SalesOrder",
      "GoodsReceiptNote"
    },
    "history-qlen": {
      "10"
    },
    "date__gt": {
      "11/01/2020"
    },
    "date__lt": {
      "09/26/2021"
    }
  }
]
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: text/html; charset=utf-8
```

Request Headers

- **Accept** – text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9

Status Codes

- **200 OK** – List of filtered history events received successfully.

3.1.1.4.1.13 InvManage.views.product_views**3.1.1.4.1.14 Module Contents****3.1.1.4.1.15 Functions**

<code>create_product_view(request)</code>	Creates a product on POST request, and returns a product creation form on GET request.
<code>update_product_view(request)</code>	Updates a product on POST request and returns the product update form on GET request.
<code>delete_product_view(request, pk)</code>	Deletes the product with primary key <code>product_id</code> on POST request.
<code>display_products_view(request)</code>	Retrieves the list of products on GET request.
<code>uploadCSV(request, data)</code>	Imports products into database from a .csv file on POST request.

`InvManage.views.product_views.create_product_view(request)`

Creates a product on POST request, and returns a product creation form on GET request.

GET /product

Gets the product creation form.

Example request:

```
GET /product/ HTTP/1.1
Host: localhost:8000
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: text/html; charset=utf-8
```

Request Headers

- **Accept** – `text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9`

Status Codes

- **200 OK** – Product creation form received successfully.

POST /product

Creates a product.

Example request:

```
POST /product/ HTTP/1.1
Host: localhost:8000
Content-Type: multipart/form-data;
```

Form Parameters

- **basic-name** – Yamaha FS100C Acoustic Guitar
- **basic-item_type** – Black Acoustic Guitar
- **basic-category** – Musical Instrument
- **basic-description** – Top: Spruce back/sides, Mahogany neck, Nato fret board, Rosewood bridge, Snorkeling frets: 20, Open chrome tuners, Finishes: Black
- **detailed-length** – 1056
- **detailed-width** – 376
- **detailed-height** – 140
- **detailed-weight** – 3.2
- **storage-barcode** – 036000029145
- **storage-expiry** – 2040-09-29
- **status-quantity** – 20
- **status-identifier** – FS100C
- **status-location** – BAY004
- **pricing-price** – 9990
- **pricing-discount** – 7

Response Headers

- **Content-Type** – multipart/form-data; boundary=—WebKitFormBoundaryLTR88aZAnBUSE7mv

Status Codes

- **302 Found** – Redirects to /product.

`InvManage.views.product_views.update_product_view(request)`

Updates a product on POST request and returns the product update form on GET request.

GET /product/update

Gets the product update form whose primary key matches the query parameter `pk`.

Example request:

```
GET /product/update HTTP/1.1
Host: localhost:8000
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
```

Query Parameters

- **pk** – The primary key of the product.

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: text/html; charset=utf-8
```

Request Headers

- **Accept** – text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9

Status Codes

- **200 OK** – Product update form received successfully.

POST /product/update

Updates the product.

Example request:

```
POST /product/update HTTP/1.1
Host: localhost:8000
Content-Type: multipart/form-data;
```

Form Parameters

- **pk** – 778
- **basic-name** – Yamaha FS100C Acoustic Guitar
- **basic-item_type** – Black Acoustic Guitar
- **basic-category** – Musical Instrument
- **basic-description** – Top: Spruce back/sides, Mahogany neck, Nato fret board, Rosewood bridge, Snorkeling frets: 20, Open chrome tuners, Finishes: Black
- **detailed-length** – 1056
- **detailed-width** – 376
- **detailed-height** – 140
- **detailed-weight** – 3.2
- **storage-barcode** – 036000029145
- **storage-expiry** – 2040-09-29
- **status-quantity** – 20
- **status-identifier** – FS100C
- **status-location** – BAY004
- **pricing-price** – 9990
- **pricing-discount** – 7

Response Headers

- **Content-Type** – multipart/form-data; boundary=---WebKitFormBoundaryLTR88aZAnBUSE7mv

Status Codes

- **302 Found** – Redirects to /product.

`InvManage.views.product_views.delete_product_view(request, pk)`

Deletes the product with primary key `product_id` on POST request.

POST `/product/<str:product_id>/delete`

Deletes the consumer represented by the primary key `product_id`.

Example request:

```
POST /product/778/delete HTTP/1.1
Host: localhost:8000
Content-Type: application/x-www-form-urlencoded
```

Parameters

- **product_id** – Product primary key.

Response Headers

- **Content-Type** – application/x-www-form-urlencoded

Status Codes

- **302 Found** – Redirects to `/product`.
- **500 Internal Server Error** – Product matching query does not exist.

`InvManage.views.product_views.display_products_view(request)`

Retrieves the list of products on GET request.

GET `/products/`

Gets the list of all products.

Example request:

```
GET /products/ HTTP/1.1
Host: localhost:8000
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
```

Form Parameters

- **page** – The page number of the products list.

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: text/html; charset=utf-8
```

Request Headers

- **Accept** – text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9

Status Codes

- **200 OK** – List of products received successfully.

`InvManage.views.product_views.uploadCSV(request, data)`

Imports products into database from a .csv file on POST request. CSV template columns: name, category, item_type, description, price, quantity, identifier, location, length, width, height, weight, discount, barcode, expiry.

POST /upload/<data>/

Imports product data.

Example request:

```
POST /product/778/delete HTTP/1.1
Host: localhost:8000
```

Status Codes

- 302 Found – Redirects to /product.

3.1.1.4.1.16 `InvManage.views.purchase_order_views`

3.1.1.4.1.17 Module Contents

3.1.1.4.1.18 Functions

<code>create_purchase_order_view(request)</code>	Creates a purchase order (PO) on POST request, and returns a PO creation form on GET request.
<code>update_purchase_order_view(request)</code>	Updates a purchase order on POST request and returns the purchase order update form for GET request.
<code>display_purchase_orders_view(request)</code>	Retrieves the list of purchase orders on GET request.
<code>delete_purchase_order_view(request, pk)</code>	Deletes the purchase order with primary key <code>object_id</code> on POST request.
<code>print_purchase_order_view(request, pk)</code>	Opens the purchase order with primary key <code>po_id</code> for printing on GET request.
<code>get_product_purchase_entries_view(request)</code>	Returns the JSON serialized data of product purchase entries on GET request.

`InvManage.views.purchase_order_views.create_purchase_order_view(request)`

Creates a purchase order (PO) on POST request, and returns a PO creation form on GET request.

GET /purchase_order

Gets the purchase order creation form.

Example request:

```
GET /purchase_order/ HTTP/1.1
Host: localhost:8000
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: text/html; charset=utf-8
```

Request Headers

- **Accept** – text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9

Status Codes

- **200 OK** – Purchase order creation form received successfully.

POST /purchase_order

Creates a purchase order.

Example request:

```
POST /purchase_order/ HTTP/1.1
Host: localhost:8000
Content-Type: multipart/form-data;
```

Form Parameters

- **po-vendor** – 4
- **vend-name** – Girish
- **vend-identifier** – GJ
- **po-po** – 69
- **po-date** – 2021-09-29
- **po-tax** – 12
- **po-discount** – 8
- **po-paid** – 4500
- **po-balance** – 1200
- **ship-title** – FingDocks
- **ship-name** – Harding Gross
- **ship-phone** – 936 651-4847
- **ship-address** – 8798 At, St., 7639
- **ship-city** – Rome
- **ship-state** – Lazio
- **ship-country** – Italy
- **ship-website** – http://fringdocs.com
- **ship-post** – 300326
- **po-subtotal** – 2595000.00
- **po-taxtotal** – 207600.00
- **po-ordertotal** – 2802600.00
- **form-TOTAL_FORMS** – 2
- **form-INITIAL_FORMS** – 0
- **form-MIN_NUM_FORMS** –

- **form-MAX_NUM_FORMS** –
- **form-0-product** – 649
- **form-0-quantity** – 300
- **form-0-price** – 4500
- **form-0-discount** – 10
- **form-0-DELETE** –
- **form-0-ppe_id** – -1
- **form-1-product** – 664
- **form-1-quantity** – 250
- **form-1-price** – 6000
- **form-1-discount** – 8
- **form-1-DELETE** –
- **form-1-ppe_id** – -1

Response Headers

- **Content-Type** – multipart/form-data; boundary=——WebKitFormBoundaryLTR88aZAnBUSE7mv

Status Codes

- **302 Found** – Redirects to /purchase_order.

`InvManage.views.purchase_order_views.update_purchase_order_view(request)`

Updates a purchase order on POST request and returns the purchase order update form for GET request.

GET /purchaes_order/update

Gets the purchase order update form whose primary key matches the query parameter pk.

Example request:

```
GET /purchase_order/update HTTP/1.1
Host: localhost:8000
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/
webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
```

Query Parameters

- **pk** – The primary key of the purchase order.

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: text/html; charset=utf-8
```

Request Headers

- **Accept** – text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9

Status Codes

- **200 OK** – Purchase order update form received successfully.

POST /purchase_order/update

Updates the purchase order.

Example request:

```
POST /purchase_order/update HTTP/1.1
Host: localhost:8000
Content-Type: multipart/form-data;
```

Form Parameters

- **pk** – 187
- **po-vendor** – 4
- **vend-name** – Girish
- **vend-identifier** – GJ
- **po-po** – 69
- **po-date** – 2021-09-29
- **po-tax** – 12
- **po-discount** – 8.0
- **po-paid** – 4500.0
- **po-balance** – 1200.0
- **ship-title** – FingDocks
- **ship-name** – Harding Gross
- **ship-phone** – 936 651-4847
- **ship-address** – 8798 At, St., 7639
- **ship-city** – Rome
- **ship-state** – Lazio
- **ship-country** – Italy
- **ship-website** – http://fringdocs.com
- **ship-post** – 300326
- **po-subtotal** – 2595000.0
- **po-taxtotal** – 207600.0
- **po-ordertotal** – 2802600.0
- **form-TOTAL_FORMS** – 2
- **form-INITIAL_FORMS** – 2
- **form-MIN_NUM_FORMS** –
- **form-MAX_NUM_FORMS** –
- **form-0-product** – 649
- **form-0-quantity** – 300

- **form-0-price** – 4500
- **form-0-discount** – 10
- **form-0-DELETE** –
- **form-0-ppe_id** – 339
- **form-1-product** – 664
- **form-1-quantity** – 250
- **form-1-price** – 6000
- **form-1-discount** – 8
- **form-1-DELETE** –
- **form-1-ppe_id** – 340

Response Headers

- **Content-Type** – multipart/form-data; boundary=—WebKitFormBoundaryLTR88aZAnBUSE7mv

Status Codes

- **302 Found** – Redirects to /consumer.

`InvManage.views.purchase_order_views.display_purchase_orders_view(request)`

Retrieves the list of purchase orders on GET request.

GET /purchase_orders/

Gets the list of all purchase orders.

Example request:

```
GET /purchase_orders/ HTTP/1.1
Host: localhost:8000
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
```

Form Parameters

- **page** – The page number of the purchase order list.

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: text/html; charset=utf-8
```

Request Headers

- **Accept** – text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9

Status Codes

- **200 OK** – List of purchase orders received successfully.

`InvManage.views.purchase_order_views.delete_purchase_order_view(request, pk)`

Deletes the purchase order with primary key `object_id` on POST request.

POST /purchase_order/<str:object_id>/delete

Deletes the consumer represented by the primary key `object_id`.

Example request:

```
POST /purchase_order/187/delete HTTP/1.1
Host: localhost:8000
Content-Type: application/x-www-form-urlencoded
```

Parameters

- **object_id** – Purchase order primary key.

Response Headers

- **Content-Type** – application/x-www-form-urlencoded

Status Codes

- **302 Found** – Redirects to `/purchase_order`.
- **500 Internal Server Error** – Purchase order matching query does not exist.

`InvManage.views.purchase_order_views.print_purchase_order_view(request, pk)`

Opens the purchase order with primary key `po_id` for printing on GET request.

POST /purchase_order/<str:po_id>/print

Opens the purchase order represented by the primary key `po_id`.

Example request:

```
POST /purchase_order/182/print HTTP/1.1
Host: localhost:8000
Content-Type: application/x-www-form-urlencoded
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json; charset=utf-8

[
  {
    "company": {
      "name": "Fringillami",
      "owner": "Ivor Barnett",
      "gstn": "89AAC056465468",
      "phone": "332 220-7026",
      "address": "Ap #849-6241 Euismod Av., 677598, Carinthia, Belgium",
      "email": "est.tempor@fringillami.org",
      "location": "Belgium",
      "image": "/media/images/hyperlink-green_x91WW5n.png"
    },
    "po": {
      "vendor": {
        "name": "Girish",
        "identifier": "GJ",
        "gstn": "GSTIN002",
```

(continues on next page)

(continued from previous page)

```

        "address": {
            "name": "alsf",
            "address": "jas;k",
            "city": ";sdalkf",
            "phone": "alsf",
            "state": "kjdfkl",
            "country": "ljflkj",
            "post": "54545"
        },
        "date": "25-Sep-2021",
        "po": 293,
        "subtotal": 279975.0,
        "taxtotal": 22398.0,
        "ordertotal": 302373.0,
        "ppes": [
            {
                "ppe_id": 324,
                "product": {
                    "pk": 637,
                    "name": "piano",
                    "category": "Ultricies PC",
                    "quantity": 23921,
                    "identifier": "PROD9",
                    "location": "Musselburgh",
                    "description": "88-key, Tri-sensor Scaled Hammer Action",
                    "prod_id": 637
                },
                "quantity": 100,
                "price": 10.0,
                "discount": 0.0,
                "order": 182,
                "pendingQty": 50
            },
            {
                "ppe_id": 325,
                "product": {
                    "pk": 645,
                    "name": "Tabl",
                    "category": "Sociis Natoque Company",
                    "quantity": 38276,
                    "identifier": "PROD17",
                    "location": "Schagen",
                    "description": "aldgjlakjlkasdj",
                    "prod_id": 645
                },
                "quantity": 250,
                "price": 90.0,
                "discount": 8.0,
                "order": 182,
                "pendingQty": 70
            }
        ]
    }

```

(continues on next page)

(continued from previous page)

```

    },
    {
      "ppe_id": 326,
      "product": {
        "pk": 638,
        "name": "Goblet drum",
        "category": "Est Congue Consulting",
        "quantity": 46076,
        "identifier": "PROD10",
        "location": "Kaluga",
        "description": "aldgjlakjlkasdj",
        "prod_id": 638
      },
      "quantity": 200,
      "price": 150.0,
      "discount": 12.0,
      "order": 182,
      "pendingQty": 200
    },
    {
      "ppe_id": 327,
      "product": {
        "pk": 643,
        "name": "quinto",
        "category": "Enim Suspendisse Associates",
        "quantity": 51099,
        "identifier": "PROD15",
        "location": "Nagpur",
        "description": "aldgjlakjlkasdj",
        "prod_id": 643
      },
      "quantity": 350,
      "price": 150.0,
      "discount": 10.0,
      "order": 182,
      "pendingQty": 350
    },
    {
      "ppe_id": 328,
      "product": {
        "pk": 651,
        "name": "Igihumurizo",
        "category": "Curabitur Massa Vestibulum Consulting",
        "quantity": 48553,
        "identifier": "PROD23",
        "location": "Yeovil",
        "description": "aldgjlakjlkasdj",
        "prod_id": 651
      },
      "quantity": 500,
      "price": 50.0,
      "discount": 6.0,

```

(continues on next page)

(continued from previous page)

```

        "order": 182,
        "pendingQty": 500
    },
    {
        "ppe_id": 329,
        "product": {
            "pk": 653,
            "name": "Balsié",
            "category": "Eu Foundation",
            "quantity": 29988,
            "identifier": "PROD25",
            "location": "Sudbury",
            "description": "aldgjlakjlkasdj",
            "prod_id": 653
        },
        "quantity": 200,
        "price": 500.0,
        "discount": 15.0,
        "order": 182,
        "pendingQty": 200
    },
    {
        "ppe_id": 330,
        "product": {
            "pk": 656,
            "name": "Padayani thappu",
            "category": "Posuere LLP",
            "quantity": 6358,
            "identifier": "PROD28",
            "location": "Ruvo del Monte",
            "description": "aldgjlakjlkasdj",
            "prod_id": 656
        },
        "quantity": 350,
        "price": 250.0,
        "discount": 13.0,
        "order": 182,
        "pendingQty": 350
    }
]
},
"shippingaddress": {
    "name": "Harding Gross",
    "address": "8798 At, St., 7639",
    "city": "Rome",
    "phone": "936 651-4847",
    "state": "Lazio",
    "country": "Italy",
    "post": "300326"
},
"communication": {
    "email": "asfs@aflk.com",

```

(continues on next page)

(continued from previous page)

```

    "phone": "6546432"
  }
}
]

```

Parameters

- **po_id** – Purchase order primary key.

Response Headers

- **Content-Type** – application/json

Status Codes

- **200 OK** – Purchase order print request successful.
- **500 Internal Server Error** – Purchase order matching query does not exist.

InvManage.views.purchase_order_views.get_product_purchase_entries_view(request)

Returns the JSON serialized data of product purchase entries on GET request.

GET /product_purchase_entries/

Gets the JSON serialized data of the requested product purchase entries.

Example request:

```

GET /product_purchase_entries/ HTTP/1.1
Host: localhost:8000
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/
↪webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9

```

Example response:

```

HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json; charset=utf-8

[
  {
    "ppe_id": 324,
    "product": {
      "pk": 637,
      "name": "piano",
      "category": "Ultricies PC",
      "quantity": 23921,
      "identifier": "PROD9",
      "location": "Musselburgh",
      "description": "88-key, Tri-sensor Scaled Hammer Action Keyboard II,
↪ Simulated ebony and ivory keys ",
      "prod_id": 637
    },
    "quantity": 100,
    "price": 10.0,
    "discount": 0.0,
    "order": 182,
  }
]

```

(continues on next page)

(continued from previous page)

```

    "pendingQty": 50,
    "po": 293
  },
  {
    "ppe_id": 325,
    "product": {
      "pk": 645,
      "name": "Tabl",
      "category": "Sociis Natoque Company",
      "quantity": 38276,
      "identifier": "PROD17",
      "location": "Schagen",
      "description": "aldgjlakjlkasdj",
      "prod_id": 645
    },
    "quantity": 250,
    "price": 90.0,
    "discount": 8.0,
    "order": 182,
    "pendingQty": 70,
    "po": 293
  },
  {
    "ppe_id": 326,
    "product": {
      "pk": 638,
      "name": "Goblet drum",
      "category": "Est Congue Consulting",
      "quantity": 46076,
      "identifier": "PROD10",
      "location": "Kaluga",
      "description": "aldgjlakjlkasdj",
      "prod_id": 638
    },
    "quantity": 200,
    "price": 150.0,
    "discount": 12.0,
    "order": 182,
    "pendingQty": 200,
    "po": 293
  },
  {
    "ppe_id": 327,
    "product": {
      "pk": 643,
      "name": "quinto",
      "category": "Enim Suspendisse Associates",
      "quantity": 51099,
      "identifier": "PROD15",
      "location": "Nagpur",
      "description": "aldgjlakjlkasdj",
      "prod_id": 643
    }
  }

```

(continues on next page)

(continued from previous page)

```

    },
    "quantity": 350,
    "price": 150.0,
    "discount": 10.0,
    "order": 182,
    "pendingQty": 350,
    "po": 293
  },
  {
    "ppe_id": 328,
    "product": {
      "pk": 651,
      "name": "Igihumurizo",
      "category": "Curabitur Massa Vestibulum Consulting",
      "quantity": 48553,
      "identifier": "PROD23",
      "location": "Yeovil",
      "description": "aldgjlakjlkasdj",
      "prod_id": 651
    },
    "quantity": 500,
    "price": 50.0,
    "discount": 6.0,
    "order": 182,
    "pendingQty": 500,
    "po": 293
  },
  {
    "ppe_id": 329,
    "product": {
      "pk": 653,
      "name": "Balsié",
      "category": "Eu Foundation",
      "quantity": 29988,
      "identifier": "PROD25",
      "location": "Sudbury",
      "description": "aldgjlakjlkasdj",
      "prod_id": 653
    },
    "quantity": 200,
    "price": 500.0,
    "discount": 15.0,
    "order": 182,
    "pendingQty": 200,
    "po": 293
  },
  {
    "ppe_id": 330,
    "product": {
      "pk": 656,
      "name": "Padayani thappu",
      "category": "Posuere LLP",

```

(continues on next page)

(continued from previous page)

```

        "quantity": 6358,
        "identifier": "PROD28",
        "location": "Ruvo del Monte",
        "description": "aldgjlakjlkasdj",
        "prod_id": 656
    },
    "quantity": 350,
    "price": 250.0,
    "discount": 13.0,
    "order": 182,
    "pendingQty": 350,
    "po": 293
}
]

```

Request Headers

- **Accept** – text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9

Status Codes

- **200 OK** – List of product purchase entries received successfully.
- **400 Bad Request** – Bad request version
- **500 Internal Server Error** – Product purchase entries matching query does not exist.

3.1.1.4.1.19 InvManage.views.sales_order_views**3.1.1.4.1.20 Module Contents****3.1.1.4.1.21 Functions**

<code>create_sales_order_view(request)</code>	Creates a sales order (SO) on POST request, and returns a SO creation form on GET request.
<code>update_sales_order_view(request)</code>	Updates a sales order on POST request and returns the sales order update form for GET request.
<code>delete_sales_order_view(request, pk)</code>	Deletes the sales order with primary key object_id on POST request.
<code>display_sales_orders_view(request)</code>	Retrieves the list of sales orders on GET request.
<code>print_sales_order_view(request, pk)</code>	Opens the sales order with primary key so_id for printing on GET request.

`InvManage.views.sales_order_views.create_sales_order_view(request)`

Creates a sales order (SO) on POST request, and returns a SO creation form on GET request.

GET /sales_order

Gets the sales order creation form.

Example request:

```
GET /sales_order/ HTTP/1.1
Host: localhost:8000
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: text/html; charset=utf-8
```

Request Headers

- **Accept** – text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9

Status Codes

- **200 OK** – Sales order creation form received successfully.

POST /sales_order

Creates a sales order.

Example request:

```
POST /sales_order/ HTTP/1.1
Host: localhost:8000
Content-Type: multipart/form-data;
```

Form Parameters

- **so-consumer** – 5
- **consumer-name** – The Music Store
- **consumer-identifier** – CONS1256
- **consumer-gstin** – 89ACC654654335
- **consumer-phone** – +91 8325642358
- **consumer-address** – Plot no 958, N- 4, Neo Complex, Barh, Wokha, Nagaland, 797111
- **consumer-email** – JohnDoe@themusic.store
- **consumer-location** – Wokha
- **so-so** – 465
- **so-date** – 2021-09-30
- **so-tax** – 12
- **so-discount** – 8
- **so-paid** – 4500
- **so-balance** – 1200
- **ship-title** – FingDocks
- **ship-name** – Harding Gross

- **ship-phone** – 936 651-4847
- **ship-address** – 8798 At, St., 7639
- **ship-city** – Rome
- **ship-state** – Lazio
- **ship-country** – Italy
- **ship-website** – <http://fringdocs.com>
- **ship-post** – 300326
- **so-subtotal** – 94600.00
- **so-taxtotal** – 7568.00
- **so-ordertotal** – 102168.00
- **form-TOTAL_FORMS** – 2
- **form-INITIAL_FORMS** – 0
- **form-MIN_NUM_FORMS** –
- **form-MAX_NUM_FORMS** –
- **form-0-product** – 645
- **form-0-quantity** – 450
- **form-0-price** – 120
- **form-0-discount** – 10
- **form-0-DELETE** –
- **form-0-pse_id** – -1
- **form-1-product** – 654
- **form-1-quantity** – 500
- **form-1-price** – 100
- **form-1-discount** – 8
- **form-1-DELETE** –
- **form-1-pse_id** – -1

Response Headers

- **Content-Type** – multipart/form-data; boundary=—WebKitFormBoundaryLTR88aZAnBUSE7mv

Status Codes

- **302 Found** – Redirects to `/sales_order`.

`InvManage.views.sales_order_views.update_sales_order_view(request)`

Updates a sales order on POST request and returns the sales order update form for GET request.

GET `/sales_order/update`

Gets the sales order update form whose primary key matches the query parameter `pk`.

Example request:

```
GET /sales_order/update HTTP/1.1
Host: localhost:8000
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
```

Query Parameters

- **pk** – The primary key of the sales order.

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: text/html; charset=utf-8
```

Request Headers

- **Accept** – text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9

Status Codes

- **200 OK** – Sales order update form received successfully.

POST /sales_order/update

Updates the sales order.

Example request:

```
POST /sales_order/update HTTP/1.1
Host: localhost:8000
Content-Type: multipart/form-data;
```

Form Parameters

- **pk** – 30
- **so-consumer** – 7
- **consumer-name** – The Music Store
- **consumer-identifier** – CONS1256
- **consumer-gstin** – 89AAC4633353643
- **consumer-phone** – +91 8325642358
- **consumer-address** – Plot no 958, N- 4, Neo Complex, Barh, Wokha, Nagaland, 797111
- **consumer-email** – JohnDoe@themusic.store
- **consumer-location** – Wokha
- **so-so** – 465
- **so-date** – 2021-09-30
- **so-tax** – 10
- **so-discount** – 8.0

- **so-paid** – 4500.0
- **so-balance** – 1200.0
- **ship-title** – FingDocks
- **ship-name** – Harding Gross
- **ship-phone** – 936 651-4847
- **ship-address** – 8798 At, St., 7639
- **ship-city** – Rome
- **ship-state** – Lazio
- **ship-country** – Italy
- **ship-website** – <http://fringdocs.com>
- **ship-post** – 300326
- **so-subtotal** – 94600.0
- **so-taxtotal** – 7568.0
- **so-ordertotal** – 102168.0
- **form-TOTAL_FORMS** – 2
- **form-INITIAL_FORMS** – 2
- **form-MIN_NUM_FORMS** –
- **form-MAX_NUM_FORMS** –
- **form-0-product** – 645
- **form-0-quantity** – 450
- **form-0-price** – 120
- **form-0-discount** – 10
- **form-0-DELETE** –
- **form-0-pse_id** – 67
- **form-1-product** – 654
- **form-1-quantity** – 500
- **form-1-price** – 100
- **form-1-discount** – 8
- **form-1-DELETE** –
- **form-1-pse_id** – 68

Response Headers

- **Content-Type** – multipart/form-data; boundary=—WebKitFormBoundaryLTR88aZAnBUSE7mv

Status Codes

- **302 Found** – Redirects to /sales_order.

`InvManage.views.sales_order_views.delete_sales_order_view(request, pk)`

Deletes the sales order with primary key `object_id` on POST request.

POST /sales_order/<str:object_id>/delete

Deletes the sales order represented by the primary key `object_id`.

Example request:

```
POST /sales_order/30/delete HTTP/1.1
Host: localhost:8000
Content-Type: application/x-www-form-urlencoded
```

Parameters

- **object_id** – Sales order primary key.

Response Headers

- **Content-Type** – application/x-www-form-urlencoded

Status Codes

- **302 Found** – Redirects to `/sales_order`.
- **500 Internal Server Error** – Sales order matching query does not exist.

`InvManage.views.sales_order_views.display_sales_orders_view(request)`

Retrieves the list of sales orders on GET request.

GET /sales_orders/

Gets the list of all sales orders.

Example request:

```
GET /sales_orders/ HTTP/1.1
Host: localhost:8000
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
```

Form Parameters

- **page** – The page number of the sales order list.

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: text/html; charset=utf-8
```

Request Headers

- **Accept** – text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9

Status Codes

- **200 OK** – List of sales orders received successfully.

`InvManage.views.sales_order_views.print_sales_order_view(request, pk)`

Opens the sales order with primary key `so_id` for printing on GET request.

POST /purchase_order/<str:so_id>/print

Opens the sales order represented by the primary key so_id.

Example request:

```
POST /sales_order/182/print HTTP/1.1
Host: localhost:8000
Content-Type: application/x-www-form-urlencoded
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json; charset=utf-8

[
  {
    "company": {
      "name": "Fringillami",
      "owner": "Ivor Barnett",
      "gstn": "89AAC056465468",
      "phone": "332 220-7026",
      "address": "Ap #849-6241 Euismod Av., 677598, Carinthia, Belgium",
      "email": "est.tempor@fringillami.org",
      "location": "Belgium",
      "image": "/media/images/hyperlink-green_x91WW5n.png"
    },
    "so": {
      "consumer": {
        "name": "The Music Store",
        "identifier": "CONS1256",
        "gstn": "89AAC4633353643",
        "phone": "+91 8325642358",
        "address": "Plot no 958, N- 4, Neo Complex, Barh, Wokha, Nagaland, 797111",
        "email": "JohnDoe@themusic.store",
        "location": "Wokha"
      },
      "date": "2021-09-25T00:00:00Z",
      "so": 89,
      "subtotal": 744900.0,
      "taxtotal": 59592.0,
      "ordertotal": 804492.0,
      "pscs": [
        {
          "pse_id": 64,
          "product": {
            "pk": 637,
            "name": "piano",
            "category": "Ultricies PC",
            "quantity": 23921,
            "identifier": "PROD9",
            "location": "Musselburgh",
            "description": "88-key, Tri-sensor Scaled Hammer Action Keyboard II, Simulated ebony and ivory keys",
            (continues on next page)
```


(continued from previous page)

```

        "prod_id": 637
      },
      "quantity": 50,
      "price": 9900.0,
      "discount": 10.0,
      "order": 29
    },
    {
      "pse_id": 65,
      "product": {
        "pk": 649,
        "name": "Sabar",
        "category": "Amet Consulting",
        "quantity": 3903,
        "identifier": "PROD21",
        "location": "Serang",
        "description": "High Sierra Sabar, Travel bag, Blue,
← Grey, Zipper, 36.5 L, 51.5 cm, 26 cm ",
        "prod_id": 649
      },
      "quantity": 25,
      "price": 4800.0,
      "discount": 8.0,
      "order": 29
    },
    {
      "pse_id": 66,
      "product": {
        "pk": 654,
        "name": "Parai",
        "category": "Aliquet Lobortis Ltd",
        "quantity": 8534,
        "identifier": "PROD26",
        "location": "Burhaniye",
        "description": "Kannan musical instruments Parai 15"
← inch (Baffallow skin) Daf Instrument",
        "prod_id": 654
      },
      "quantity": 35,
      "price": 6000.0,
      "discount": 10.0,
      "order": 29
    }
  ]
},
"shippingaddress": {
  "name": "Harding Gross",
  "address": "8798 At, St., 7639",
  "city": "Rome",
  "phone": "936 651-4847",
  "state": "Lazio",
  "country": "Italy",

```

(continues on next page)

(continued from previous page)

```

    "post": "300326"
  }
}
]

```

Parameters

- **so_id** – Sales order primary key.

Response Headers

- **Content-Type** – application/json

Status Codes

- **200 OK** – Sales order print request successful.
- **500 Internal Server Error** – Sales order matching query does not exist.

3.1.1.4.1.22 InvManage.views.vendor_views**3.1.1.4.1.23 Module Contents****3.1.1.4.1.24 Functions**

<code>create_vendor_view(request)</code>	Creates a vendor on POST request, and returns a vendor creation form on GET request.
<code>update_vendor_view(request)</code>	Updates a vendor on POST request and returns the vendor update form for GET request.
<code>delete_vendor_view(request, pk)</code>	Deletes the vendor with primary key pk on POST request.
<code>display_vendors_view(request)</code>	Retrieves the list of vendors on GET request.
<code>get_vendor(request)</code>	Returns the JSON serialized data of the requested vendor on GET request.

InvManage.views.vendor_views.create_vendor_view(request)

Creates a vendor on POST request, and returns a vendor creation form on GET request.

GET /vendor

Gets the vendor creation form.

Example request:

```

GET /vendor/ HTTP/1.1
Host: localhost:8000
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9

```

Example response:

```

HTTP/1.1 200 OK
Vary: Accept
Content-Type: text/html; charset=utf-8

```

Request Headers

- **Accept** – text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9

Status Codes

- **200 OK** – Vendor creation form received successfully.

POST /vendor

Creates a vendor.

Example request:

```
POST /vendor/ HTTP/1.1
Host: localhost:8000
Content-Type: multipart/form-data;
```

Form Parameters

- **vend-name** – Lug Vendor
- **vend-identifier** – TBPB-02692
- **vend-gstin** – 89AAC254254F2
- **ship-title** – AKATSUKI
- **ship-name** – Kuame Burns
- **ship-phone** – 679 166-3127
- **ship-address** – Nonummy Avenue
- **ship-city** – Chung Cheong
- **ship-state** – Guanacaste
- **ship-country** – tellusidnunc.net
- **ship-website** – Germany.protitor@tellusid.net
- **ship-post** – 8949
- **pdform-currency** – DEM
- **pdform-minorder** – 2000
- **pdform-contactperson** – Harding Gross
- **pdform-refcode** – CUST000124
- **pdform-transportmode** – Express
- **com-language** – German
- **com-phone** – 936 651-4817
- **com-email** – non.sollicitudin@uttincidunt.org
- **com-fax** – 323 555 1234
- **bank-name** – FIRST FLORIDA INTEGRITY BANK
- **bank-branch** – Bavaria
- **bank-region** – Bayem

- **bank-route** – 67016325
- **bank-number** – 42543251393
- **bank-acctype** – Current
- **bank-iban** – DE6233542
- **bank-code** – BA54354354
- **bank-branchcode** – BA35435823

Response Headers

- **Content-Type** – multipart/form-data; boundary=—WebKitFormBoundaryLTR88aZAnBUSE7mv

Status Codes

- **302 Found** – Redirects to /vendor.

`InvManage.views.vendor_views.update_vendor_view(request)`

Updates a vendor on POST request and returns the vendor update form for GET request.

GET /vendor/update

Gets the vendor update form whose primary key matches the query parameter pk.

Example request:

```
GET /vendor/update HTTP/1.1
Host: localhost:8000
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
```

Query Parameters

- **pk** – The primary key of the vendor.

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: text/html; charset=utf-8
```

Request Headers

- **Accept** – text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9

Status Codes

- **200 OK** – Vendor update form received successfully.

POST /vendor/update

Updates the vendor.

Example request:

```
POST /vendor/update HTTP/1.1
Host: localhost:8000
Content-Type: multipart/form-data;
```

Form Parameters

- **vend-name** – Lug Vendor
- **vend-identifier** – TBPB-02692
- **vend-gstin** – 89AAC254254F2
- **ship-title** – AKATSUKI
- **ship-name** – Kuame Burns
- **ship-phone** – 679 166-3127
- **ship-address** – Nonummy Avenue
- **ship-city** – Chung Cheong
- **ship-state** – Guanacaste
- **ship-country** – tellusidnunc.net
- **ship-website** – Germany.protitor@tellusid.net
- **ship-post** – 8949
- **pdform-currency** – DEM
- **pdform-minorder** – 1000
- **pdform-contactperson** – Harding Gross
- **pdform-refcode** – CUST000124
- **pdform-transportmode** – Express
- **com-language** – German
- **com-phone** – 936 651-4817
- **com-email** – non.sollicitudin@uttincidunt.org
- **com-fax** – 323 555 1234
- **bank-name** – FIRST FLORIDA INTEGRITY BANK
- **bank-branch** – Bavaria
- **bank-region** – Bayem
- **bank-route** – 67016325
- **bank-number** – 42543251393
- **bank-acctype** – Current
- **bank-iban** – DE6233542
- **bank-code** – BA54354354
- **bank-branchcode** – BA35435823

Response Headers

- **Content-Type** – multipart/form-data; boundary=—WebKitFormBoundaryLTR88aZAnBUSE7mv

Status Codes

- **302 Found** – Redirects to /consumer.

`InvManage.views.vendor_views.delete_vendor_view(request, pk)`

Deletes the vendor with primary key `pk` on POST request.

POST `/vendor/<str:object_id>/delete`

Deletes the vendor represented by the primary key `object_id`.

Example request:

```
POST /vendor/5/delete HTTP/1.1
Host: localhost:8000
Content-Type: application/x-www-form-urlencoded
```

Parameters

- **object_id** – Vendor primary key.

Response Headers

- **Content-Type** – application/x-www-form-urlencoded

Status Codes

- **302 Found** – Redirects to `/vendors`.
- **500 Internal Server Error** – Vendor matching query does not exist.

`InvManage.views.vendor_views.display_vendors_view(request)`

Retrieves the list of vendors on GET request.

GET `/vendors/`

Gets the list of all vendors.

Example request:

```
GET /vendors/ HTTP/1.1
Host: localhost:8000
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
```

Form Parameters

- **page** – The page number of the vendors list.

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: text/html; charset=utf-8
```

Request Headers

- **Accept** – text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9

Status Codes

- **200 OK** – List of vendors received successfully.

`InvManage.views.vendor_views.get_vendor(request)`

Returns the JSON serialized data of the requested vendor on GET request.

GET /get_vendor/

Gets the JSON serialized data of the requested vendor.

Example request:

```
GET /get_vendor/ HTTP/1.1
Host: localhost:8000
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
```

Parameters

- **vendor_id** – Vendor primary key.

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json; charset=utf-8

[
  {
    "name": "Lug Vendor",
    "identifier": "TBPB-02692",
    "gstn": "89AAC4683897343",
    "address": {
      "name": "Kuame Burns",
      "address": "Nonummy Avenue",
      "city": "Chung Cheong",
      "phone": "679 166-3127",
      "state": "Guanacaste",
      "country": "tellusidnunc.net",
      "post": "8949"
    }
  }
]
```

Response Headers

- **Content-Type** – application/json

Status Codes

- **200 OK** – List of vendors received successfully.
- **400 Bad Request** – Bad request version
- **500 Internal Server Error** – Vendor matching query does not exist.

3.1.1.4.2 Package Contents

3.1.1.4.2.1 Classes

<i>ProductFilter</i>	
<i>ProductBasicInfoForm</i>	Form for basic product information.
<i>ProductDetailedInfoForm</i>	Form for product physical details
<i>ProductPricingForm</i>	Form for pricing information of the product.
<i>ProductPurchaseEntryForm</i>	Form for product purchase entry.
<i>ProductSalesEntryForm</i>	Form for product sales entry.
<i>ProductStatusForm</i>	Form for locating and identifying the product.
<i>ProductStorageInfoForm</i>	Form for storage information.
<i>PurchaseOrderBasicInfo</i>	Form for basic information of the purchase order.
<i>GRNInfo</i>	Form for goods receipt note (GRN).
<i>GRNEntryForm</i>	Form for goods receipt note entry (GRNE).
<i>ThumbnailForm</i>	For for product thumbnail image.
<i>SalesOrderBasicInfo</i>	Form for basic information on sales order.
<i>ShippingAddressForm</i>	For for shipping address.
<i>VendorForm</i>	Form for basic informaion of the vendor.
<i>ConsumerForm</i>	Form for consumer.
<i>CompanyForm</i>	Form for company.
<i>HistoryForm</i>	Form for history view.
<i>CommunicationForm</i>	Form for communication information.
<i>PurchaseDataForm</i>	Form for purchase related data.
<i>BankAccountForm</i>	Form for bank account details.
<i>Product</i>	Model of the product.
<i>ProductPurchaseEntry</i>	Model of the product purchase entry.
<i>GRNEntry</i>	Model of the goods receipt note entry (GRNE).
<i>ProductSalesEntry</i>	Model of the product sales entry.
<i>SalesOrder</i>	Model of the sales order.
<i>PurchaseOrder</i>	Model of the purchase order.
<i>GoodsReceiptNote</i>	Model of the goods receipt note (GRN).
<i>Consumer</i>	Model of the consumer.
<i>Vendor</i>	Model of the vendor.
<i>Company</i>	Model of the company.
<i>PurchaseInvoice</i>	Model for the purchase invoice.
<i>SalesInvoice</i>	Model for the sales order invoice.
<i>FilterState</i>	Model of the filter state.
<i>FilterColumn</i>	Model of the column state.
<i>Object</i>	Model for a generic object.
<i>EventCard</i>	Model for event card to be displayed in the history view.
<i>EventType</i>	Model for event type.
<i>ObjectModel</i>	Model associated with an object.
<i>HistoryFilterState</i>	Model of filter state of the history.
<i>ShippingAddress</i>	Model of the shipping address.
<i>PurchaseData</i>	Model of the purchase data.
<i>Communication</i>	Model of the communication information.
<i>BankAccount</i>	Model of the bank account details.

continues on next page

Table 18 – continued from previous page

<i>GoodsReceiptNoteFilter</i>	
<i>GRNEntrySerializer</i>	Serializer for GRNEntry class instance.
<i>GRNEntryWithPORefSerializer</i>	Serializer for GRNEntry class instance if the GRN references a purchase order.
<i>GRNInvoiceSerializer</i>	Serializer for GRNInvoice class instance.
<i>GoodsReceiptNoteSerializer</i>	Serializer for GoodsReceiptNote class instance.
<i>PPEntrySerializer</i>	Serializer for ProductPurchaseEntry class instance.
<i>ProductSerializer</i>	Serializer for Product class instance.
<i>PurchaseInvoiceSerializer</i>	Serializer for PurchaseInvoice class instance.
<i>GRNInvoice</i>	Model for the goods receipt note invoice.
<i>ProductBasicInfoForm</i>	Form for basic product information.
<i>ProductDetailedInfoForm</i>	Form for product physical details
<i>ProductPricingForm</i>	Form for pricing information of the product.
<i>ProductPurchaseEntryForm</i>	Form for product purchase entry.
<i>ProductSalesEntryForm</i>	Form for product sales entry.
<i>ProductStatusForm</i>	Form for locating and identifying the product.
<i>ProductStorageInfoForm</i>	Form for storage information.
<i>PurchaseOrderBasicInfo</i>	Form for basic information of the purchase order.
<i>GRNInfo</i>	Form for goods receipt note (GRN).
<i>GRNEntryForm</i>	Form for goods receipt note entry (GRNE).
<i>ThumbnailForm</i>	For for product thumbnail image.
<i>SalesOrderBasicInfo</i>	Form for basic information on sales order.
<i>ShippingAddressForm</i>	For for shipping address.
<i>VendorForm</i>	Form for basic informaion of the vendor.
<i>ConsumerForm</i>	Form for consumer.
<i>CompanyForm</i>	Form for company.
<i>HistoryForm</i>	Form for history view.
<i>CommunicationForm</i>	Form for communication information.
<i>PurchaseDataForm</i>	Form for purchase related data.
<i>BankAccountForm</i>	Form for bank account details.
<i>Product</i>	Model of the product.
<i>ProductPurchaseEntry</i>	Model of the product purchase entry.
<i>GRNEntry</i>	Model of the goods receipt note entry (GRNE).
<i>ProductSalesEntry</i>	Model of the product sales entry.
<i>SalesOrder</i>	Model of the sales order.
<i>PurchaseOrder</i>	Model of the purchase order.
<i>GoodsReceiptNote</i>	Model of the goods receipt note (GRN).
<i>Consumer</i>	Model of the consumer.
<i>Vendor</i>	Model of the vendor.
<i>Company</i>	Model of the company.
<i>PurchaseInvoice</i>	Model for the purchase invoice.
<i>SalesInvoice</i>	Model for the sales order invoice.
<i>FilterState</i>	Model of the filter state.
<i>FilterColumn</i>	Model of the column state.
<i>Object</i>	Model for a generic object.
<i>EventCard</i>	Model for event card to be displayed in the history view.
<i>EventType</i>	Model for event type.
<i>ObjectModel</i>	Model associated with an object.
<i>HistoryFilterState</i>	Model of filter state of the history.

continues on next page

Table 18 – continued from previous page

<i>ShippingAddress</i>	Model of the shipping address.
<i>PurchaseData</i>	Model of the purchase data.
<i>Communication</i>	Model of the communication information.
<i>BankAccount</i>	Model of the bank account details.
<i>PurchaseOrderFilter</i>	
<i>ProductSerializer</i>	Serializer for <i>Product</i> class instance.
<i>PPEntrySerializer</i>	Serializer for <i>ProductPurchaseEntry</i> class instance.
<i>PurchaseInvoiceSerializer</i>	Serializer for <i>PurchaseInvoice</i> class instance.
<i>ProductBasicInfoForm</i>	Form for basic product information.
<i>ProductDetailedInfoForm</i>	Form for product physical details
<i>ProductPricingForm</i>	Form for pricing information of the product.
<i>ProductPurchaseEntryForm</i>	Form for product purchase entry.
<i>ProductSalesEntryForm</i>	Form for product sales entry.
<i>ProductStatusForm</i>	Form for locating and identifying the product.
<i>ProductStorageInfoForm</i>	Form for storage information.
<i>PurchaseOrderBasicInfo</i>	Form for basic information of the purchase order.
<i>GRNInfo</i>	Form for goods receipt note (GRN).
<i>GRNEntryForm</i>	Form for goods receipt note entry (GRNE).
<i>ThumbnailForm</i>	For for product thumbnail image.
<i>SalesOrderBasicInfo</i>	Form for basic information on sales order.
<i>ShippingAddressForm</i>	For for shipping address.
<i>VendorForm</i>	Form for basic informaion of the vendor.
<i>ConsumerForm</i>	Form for consumer.
<i>CompanyForm</i>	Form for company.
<i>HistoryForm</i>	Form for history view.
<i>CommunicationForm</i>	Form for communication information.
<i>PurchaseDataForm</i>	Form for purchase related data.
<i>BankAccountForm</i>	Form for bank account details.
<i>Product</i>	Model of the product.
<i>ProductPurchaseEntry</i>	Model of the product purchase entry.
<i>GRNEntry</i>	Model of the goods receipt note entry (GRNE).
<i>ProductSalesEntry</i>	Model of the product sales entry.
<i>SalesOrder</i>	Model of the sales order.
<i>PurchaseOrder</i>	Model of the purchase order.
<i>GoodsReceiptNote</i>	Model of the goods receipt note (GRN).
<i>Consumer</i>	Model of the consumer.
<i>Vendor</i>	Model of the vendor.
<i>Company</i>	Model of the company.
<i>PurchaseInvoice</i>	Model for the purchase invoice.
<i>SalesInvoice</i>	Model for the sales order invoice.
<i>FilterState</i>	Model of the filter state.
<i>FilterColumn</i>	Model of the column state.
<i>Object</i>	Model for a generic object.
<i>EventCard</i>	Model for event card to be displayed in the history view.
<i>EventType</i>	Model for event type.
<i>ObjectModel</i>	Model associated with an object.
<i>HistoryFilterState</i>	Model of filter state of the history.
<i>ShippingAddress</i>	Model of the shipping address.
<i>PurchaseData</i>	Model of the purchase data.

continues on next page

Table 18 – continued from previous page

<i>Communication</i>	Model of the communication information.
<i>BankAccount</i>	Model of the bank account details.
<i>CompanySerializer</i>	Serializer for <i>Company</i> class instance.
<i>ShippingAddressSerializer</i>	Serializer for <i>ShippingAddress</i> class instance.
<i>CompanyFilter</i>	
<i>ProductBasicInfoForm</i>	Form for basic product information.
<i>ProductDetailedInfoForm</i>	Form for product physical details
<i>ProductPricingForm</i>	Form for pricing information of the product.
<i>ProductPurchaseEntryForm</i>	Form for product purchase entry.
<i>ProductSalesEntryForm</i>	Form for product sales entry.
<i>ProductStatusForm</i>	Form for locating and identifying the product.
<i>ProductStorageInfoForm</i>	Form for storage information.
<i>PurchaseOrderBasicInfo</i>	Form for basic information of the purchase order.
<i>GRNInfo</i>	Form for goods receipt note (GRN).
<i>GRNEntryForm</i>	Form for goods receipt note entry (GRNE).
<i>ThumbnailForm</i>	For for product thumbnail image.
<i>SalesOrderBasicInfo</i>	Form for basic information on sales order.
<i>ShippingAddressForm</i>	For for shipping address.
<i>VendorForm</i>	Form for basic informaion of the vendor.
<i>ConsumerForm</i>	Form for consumer.
<i>CompanyForm</i>	Form for company.
<i>HistoryForm</i>	Form for history view.
<i>CommunicationForm</i>	Form for communication information.
<i>PurchaseDataForm</i>	Form for purchase related data.
<i>BankAccountForm</i>	Form for bank account details.
<i>Product</i>	Model of the product.
<i>ProductPurchaseEntry</i>	Model of the product purchase entry.
<i>GRNEntry</i>	Model of the goods receipt note entry (GRNE).
<i>ProductSalesEntry</i>	Model of the product sales entry.
<i>SalesOrder</i>	Model of the sales order.
<i>PurchaseOrder</i>	Model of the purchase order.
<i>GoodsReceiptNote</i>	Model of the goods receipt note (GRN).
<i>Consumer</i>	Model of the consumer.
<i>Vendor</i>	Model of the vendor.
<i>Company</i>	Model of the company.
<i>PurchaseInvoice</i>	Model for the purchase invoice.
<i>SalesInvoice</i>	Model for the sales order invoice.
<i>FilterState</i>	Model of the filter state.
<i>FilterColumn</i>	Model of the column state.
<i>Object</i>	Model for a generic object.
<i>EventCard</i>	Model for event card to be displayed in the history view.
<i>EventType</i>	Model for event type.
<i>ObjectModel</i>	Model associated with an object.
<i>HistoryFilterState</i>	Model of filter state of the history.
<i>ShippingAddress</i>	Model of the shipping address.
<i>PurchaseData</i>	Model of the purchase data.
<i>Communication</i>	Model of the communication information.
<i>BankAccount</i>	Model of the bank account details.

continues on next page

Table 18 – continued from previous page

<i>VendorFilter</i>	
<i>VendorSerializer</i>	Serializer for <i>Vendor</i> class instance.
<i>ProductBasicInfoForm</i>	Form for basic product information.
<i>ProductDetailedInfoForm</i>	Form for product physical details
<i>ProductPricingForm</i>	Form for pricing information of the product.
<i>ProductPurchaseEntryForm</i>	Form for product purchase entry.
<i>ProductSalesEntryForm</i>	Form for product sales entry.
<i>ProductStatusForm</i>	Form for locating and identifying the product.
<i>ProductStorageInfoForm</i>	Form for storage information.
<i>PurchaseOrderBasicInfo</i>	Form for basic information of the purchase order.
<i>GRNInfo</i>	Form for goods receipt note (GRN).
<i>GRNEntryForm</i>	Form for goods receipt note entry (GRNE).
<i>ThumbnailForm</i>	For for product thumbnail image.
<i>SalesOrderBasicInfo</i>	Form for basic information on sales order.
<i>ShippingAddressForm</i>	For for shipping address.
<i>VendorForm</i>	Form for basic informaion of the vendor.
<i>ConsumerForm</i>	Form for consumer.
<i>CompanyForm</i>	Form for company.
<i>HistoryForm</i>	Form for history view.
<i>CommunicationForm</i>	Form for communication information.
<i>PurchaseDataForm</i>	Form for purchase related data.
<i>BankAccountForm</i>	Form for bank account details.
<i>Product</i>	Model of the product.
<i>ProductPurchaseEntry</i>	Model of the product purchase entry.
<i>GRNEntry</i>	Model of the goods receipt note entry (GRNE).
<i>ProductSalesEntry</i>	Model of the product sales entry.
<i>SalesOrder</i>	Model of the sales order.
<i>PurchaseOrder</i>	Model of the purchase order.
<i>GoodsReceiptNote</i>	Model of the goods receipt note (GRN).
<i>Consumer</i>	Model of the consumer.
<i>Vendor</i>	Model of the vendor.
<i>Company</i>	Model of the company.
<i>PurchaseInvoice</i>	Model for the purchase invoice.
<i>SalesInvoice</i>	Model for the sales order invoice.
<i>FilterState</i>	Model of the filter state.
<i>FilterColumn</i>	Model of the column state.
<i>Object</i>	Model for a generic object.
<i>EventCard</i>	Model for event card to be displayed in the history view.
<i>EventType</i>	Model for event type.
<i>ObjectModel</i>	Model associated with an object.
<i>HistoryFilterState</i>	Model of filter state of the history.
<i>ShippingAddress</i>	Model of the shipping address.
<i>PurchaseData</i>	Model of the purchase data.
<i>Communication</i>	Model of the communication information.
<i>BankAccount</i>	Model of the bank account details.
<i>ConsumerFilter</i>	
<i>ConsumerSerializer</i>	Serializer for <i>Consumer</i> class instance.
<i>ProductBasicInfoForm</i>	Form for basic product information.

continues on next page

Table 18 – continued from previous page

<i>ProductDetailedInfoForm</i>	Form for product physical details
<i>ProductPricingForm</i>	Form for pricing information of the product.
<i>ProductPurchaseEntryForm</i>	Form for product purchase entry.
<i>ProductSalesEntryForm</i>	Form for product sales entry.
<i>ProductStatusForm</i>	Form for locating and identifying the product.
<i>ProductStorageInfoForm</i>	Form for storage information.
<i>PurchaseOrderBasicInfo</i>	Form for basic information of the purchase order.
<i>GRNInfo</i>	Form for goods receipt note (GRN).
<i>GRNEntryForm</i>	Form for goods receipt note entry (GRNE).
<i>ThumbnailForm</i>	For for product thumbnail image.
<i>SalesOrderBasicInfo</i>	Form for basic information on sales order.
<i>ShippingAddressForm</i>	For for shipping address.
<i>VendorForm</i>	Form for basic informaion of the vendor.
<i>ConsumerForm</i>	Form for consumer.
<i>CompanyForm</i>	Form for company.
<i>HistoryForm</i>	Form for history view.
<i>CommunicationForm</i>	Form for communication information.
<i>PurchaseDataForm</i>	Form for purchase related data.
<i>BankAccountForm</i>	Form for bank account details.
<i>Product</i>	Model of the product.
<i>ProductPurchaseEntry</i>	Model of the product purchase entry.
<i>GRNEntry</i>	Model of the goods receipt note entry (GRNE).
<i>ProductSalesEntry</i>	Model of the product sales entry.
<i>SalesOrder</i>	Model of the sales order.
<i>PurchaseOrder</i>	Model of the purchase order.
<i>GoodsReceiptNote</i>	Model of the goods receipt note (GRN).
<i>Consumer</i>	Model of the consumer.
<i>Vendor</i>	Model of the vendor.
<i>Company</i>	Model of the company.
<i>PurchaseInvoice</i>	Model for the purchase invoice.
<i>SalesInvoice</i>	Model for the sales order invoice.
<i>FilterState</i>	Model of the filter state.
<i>FilterColumn</i>	Model of the column state.
<i>Object</i>	Model for a generic object.
<i>EventCard</i>	Model for event card to be displayed in the history view.
<i>EventType</i>	Model for event type.
<i>ObjectModel</i>	Model associated with an object.
<i>HistoryFilterState</i>	Model of filter state of the history.
<i>ShippingAddress</i>	Model of the shipping address.
<i>PurchaseData</i>	Model of the purchase data.
<i>Communication</i>	Model of the communication information.
<i>BankAccount</i>	Model of the bank account details.
<i>SalesOrderFilter</i>	
<i>ProductSerializer</i>	Serializer for Product class instance.
<i>PSEntrySerializer</i>	Serializer for ProductSalesEntry class instance.
<i>SalesInvoiceSerializer</i>	Serializer for SalesInvoice class instance.
<i>ProductBasicInfoForm</i>	Form for basic product information.
<i>ProductDetailedInfoForm</i>	Form for product physical details
<i>ProductPricingForm</i>	Form for pricing information of the product.

continues on next page

Table 18 – continued from previous page

<i>ProductPurchaseEntryForm</i>	Form for product purchase entry.
<i>ProductSalesEntryForm</i>	Form for product sales entry.
<i>ProductStatusForm</i>	Form for locating and identifying the product.
<i>ProductStorageInfoForm</i>	Form for storage information.
<i>PurchaseOrderBasicInfo</i>	Form for basic information of the purchase order.
<i>GRNInfo</i>	Form for goods receipt note (GRN).
<i>GRNEntryForm</i>	Form for goods receipt note entry (GRNE).
<i>ThumbnailForm</i>	Form for product thumbnail image.
<i>SalesOrderBasicInfo</i>	Form for basic information on sales order.
<i>ShippingAddressForm</i>	Form for shipping address.
<i>VendorForm</i>	Form for basic information of the vendor.
<i>ConsumerForm</i>	Form for consumer.
<i>CompanyForm</i>	Form for company.
<i>HistoryForm</i>	Form for history view.
<i>CommunicationForm</i>	Form for communication information.
<i>PurchaseDataForm</i>	Form for purchase related data.
<i>BankAccountForm</i>	Form for bank account details.
<i>Product</i>	Model of the product.
<i>ProductPurchaseEntry</i>	Model of the product purchase entry.
<i>GRNEntry</i>	Model of the goods receipt note entry (GRNE).
<i>ProductSalesEntry</i>	Model of the product sales entry.
<i>SalesOrder</i>	Model of the sales order.
<i>PurchaseOrder</i>	Model of the purchase order.
<i>GoodsReceiptNote</i>	Model of the goods receipt note (GRN).
<i>Consumer</i>	Model of the consumer.
<i>Vendor</i>	Model of the vendor.
<i>Company</i>	Model of the company.
<i>PurchaseInvoice</i>	Model for the purchase invoice.
<i>SalesInvoice</i>	Model for the sales order invoice.
<i>FilterState</i>	Model of the filter state.
<i>FilterColumn</i>	Model of the column state.
<i>Object</i>	Model for a generic object.
<i>EventCard</i>	Model for event card to be displayed in the history view.
<i>EventType</i>	Model for event type.
<i>ObjectModel</i>	Model associated with an object.
<i>HistoryFilterState</i>	Model of filter state of the history.
<i>ShippingAddress</i>	Model of the shipping address.
<i>PurchaseData</i>	Model of the purchase data.
<i>Communication</i>	Model of the communication information.
<i>BankAccount</i>	Model of the bank account details.
<i>EventCard</i>	Model for event card to be displayed in the history view.
<i>HistoryFilterState</i>	Model of filter state of the history.
<i>EventCardFilter</i>	
<i>HistoryFilterStateSerializer</i>	Serializer for <i>HistoryFilterState</i> class instance.
<i>HistoryForm</i>	Form for history view.

3.1.1.4.2.2 Functions

<i>create_product_view</i> (request)	Creates a product on POST request, and returns a product creation form on GET request.
<i>update_product_view</i> (request)	Updates a product on POST request and returns the product update form on GET request.
<i>delete_product_view</i> (request, pk)	Deletes the product with primary key <code>product_id</code> on POST request.
<i>display_products_view</i> (request)	Retrieves the list of products on GET request.
<i>uploadCSV</i> (request, data)	Imports products into database from a .csv file on POST request.
<i>get_columns</i> (state)	Gets the columns of the display table.
<i>sort_ascending_descending</i> (request, model)	Sorts and saves the state of the column.
<i>change_column_position</i> (request, state)	Modifies the column states depending on the request.
<i>paginate</i> (queryset, filter, page_number)	Paginates the list of objects.
<i>create_grn_view</i> (request)	Creates a goods receipt note (GRN) on POST request, and returns a GRN creation form on GET request.
<i>update_grn_view</i> (request)	Updates the goods receipt note (GRN) on POST request, and returns a GRN update form on GET request.
<i>delete_grn_view</i> (request, pk)	Deletes the GRN with primary key <code>object_id</code> on POST request.
<i>display_grns_view</i> (request)	Retrieves the list of GRNs on GET request.
<i>print_grn_view</i> (request, pk)	Opens the GRN with primary key <code>grn_id</code> for printing on GET request.
<i>get_columns</i> (state)	Gets the columns of the display table.
<i>sort_ascending_descending</i> (request, model)	Sorts and saves the state of the column.
<i>change_column_position</i> (request, state)	Modifies the column states depending on the request.
<i>paginate</i> (queryset, filter, page_number)	Paginates the list of objects.
<i>create_purchase_order_view</i> (request)	Creates a purchase order (PO) on POST request, and returns a PO creation form on GET request.
<i>update_purchase_order_view</i> (request)	Updates a purchase order on POST request and returns the purchase order update form for GET request.
<i>display_purchase_orders_view</i> (request)	Retrieves the list of purchase orders on GET request.
<i>delete_purchase_order_view</i> (request, pk)	Deletes the purchase order with primary key <code>object_id</code> on POST request.
<i>print_purchase_order_view</i> (request, pk)	Opens the purchase order with primary key <code>po_id</code> for printing on GET request.
<i>get_product_purchase_entries_view</i> (request)	Returns the JSON serialized data of product purchase entries on GET request.
<i>get_columns</i> (state)	Gets the columns of the display table.
<i>sort_ascending_descending</i> (request, model)	Sorts and saves the state of the column.
<i>change_column_position</i> (request, state)	Modifies the column states depending on the request.
<i>paginate</i> (queryset, filter, page_number)	Paginates the list of objects.
<i>create_company_view</i> (request)	Creates a company on POST request and returns a company creation form GET request.
<i>update_company_view</i> (request)	Updates a company on POST request and returns the company update form GET request.
<i>delete_company_view</i> (request, pk)	Deletes the company with primary key <code>pk</code> on POST request.
<i>display_companies_view</i> (request)	Retrieves the list of companies on GET request.
<i>get_columns</i> (state)	Gets the columns of the display table.

continues on next page

Table 19 – continued from previous page

<i>sort_ascending_descending</i> (request, model)	Sorts and saves the state of the column.
<i>change_column_position</i> (request, state)	Modifies the column states depending on the request.
<i>paginate</i> (queryset, filter, page_number)	Paginates the list of objects.
<i>create_vendor_view</i> (request)	Creates a vendor on POST request, and returns a vendor creation form on GET request.
<i>update_vendor_view</i> (request)	Updates a vendor on POST request and returns the vendor update form for GET request.
<i>delete_vendor_view</i> (request, pk)	Deletes the vendor with primary key pk on POST request.
<i>display_vendors_view</i> (request)	Retrieves the list of vendors on GET request.
<i>get_vendor</i> (request)	Returns the JSON serialized data of the requested vendor on GET request.
<i>get_columns</i> (state)	Gets the columns of the display table.
<i>sort_ascending_descending</i> (request, model)	Sorts and saves the state of the column.
<i>change_column_position</i> (request, state)	Modifies the column states depending on the request.
<i>paginate</i> (queryset, filter, page_number)	Paginates the list of objects.
<i>create_consumer_view</i> (request)	Creates a consumer on POST request, and returns a consumer creation form on GET request.
<i>update_consumer_view</i> (request)	Updates a consumer on POST request and returns the consumer update form for GET request.
<i>delete_consumer_view</i> (request, pk)	Deletes the consumer with primary key pk on POST request.
<i>display_consumers_view</i> (request)	Retrieves the list of consumers on GET request.
<i>get_consumer</i> (request)	Returns the JSON serialized data of the requested consumer on GET request.
<i>get_columns</i> (state)	Gets the columns of the display table.
<i>sort_ascending_descending</i> (request, model)	Sorts and saves the state of the column.
<i>change_column_position</i> (request, state)	Modifies the column states depending on the request.
<i>paginate</i> (queryset, filter, page_number)	Paginates the list of objects.
<i>create_sales_order_view</i> (request)	Creates a sales order (SO) on POST request, and returns a SO creation form on GET request.
<i>update_sales_order_view</i> (request)	Updates a sales order on POST request and returns the sales order update form for GET request.
<i>delete_sales_order_view</i> (request, pk)	Deletes the sales order with primary key object_id on POST request.
<i>display_sales_orders_view</i> (request)	Retrieves the list of sales orders on GET request.
<i>print_sales_order_view</i> (request, pk)	Opens the sales order with primary key so_id for printing on GET request.
<i>get_columns</i> (state)	Gets the columns of the display table.
<i>sort_ascending_descending</i> (request, model)	Sorts and saves the state of the column.
<i>change_column_position</i> (request, state)	Modifies the column states depending on the request.
<i>paginate</i> (queryset, filter, page_number)	Paginates the list of objects.
<i>display_history_view</i> (request)	Retrieves the list of events on GET request. The create, update, and delete events are registered for each model.

```

class InvManage.views.ProductFilter
    Bases: django_filters.FilterSet

    class Meta

        model

```


fields
context
name
category
quantity
price
identifier
location
order_by_field = ordering
ordering

InvManage.views.**create_product_view**(*request*)

Creates a product on POST request, and returns a product creation form on GET request.

GET /product

Gets the product creation form.

Example request:

```
GET /product/ HTTP/1.1
Host: localhost:8000
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: text/html; charset=utf-8
```

Request Headers

- **Accept** – text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9

Status Codes

- **200 OK** – Product creation form received successfully.

POST /product

Creates a product.

Example request:

```
POST /product/ HTTP/1.1
Host: localhost:8000
Content-Type: multipart/form-data;
```

Form Parameters

- **basic-name** – Yamaha FS100C Acoustic Guitar
- **basic-item_type** – Black Acoustic Guitar

- **basic-category** – Musical Instrument
- **basic-description** – Top: Spruce back/sides, Mahogany neck, Nato fret board, Rosewood bridge, Snorkeling frets: 20, Open chrome tuners, Finishes: Black
- **detailed-length** – 1056
- **detailed-width** – 376
- **detailed-height** – 140
- **detailed-weight** – 3.2
- **storage-barcode** – 036000029145
- **storage-expiry** – 2040-09-29
- **status-quantity** – 20
- **status-identifier** – FS100C
- **status-location** – BAY004
- **pricing-price** – 9990
- **pricing-discount** – 7

Response Headers

- **Content-Type** – multipart/form-data; boundary=—WebKitFormBoundaryLTR88aZAnBUSE7mv

Status Codes

- **302 Found** – Redirects to /product.

`InvManage.views.update_product_view(request)`

Updates a product on POST request and returns the product update form on GET request.

GET /product/update

Gets the product update form whose primary key matches the query parameter pk.

Example request:

```
GET /product/update HTTP/1.1
Host: localhost:8000
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
```

Query Parameters

- **pk** – The primary key of the product.

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: text/html; charset=utf-8
```

Request Headers

- **Accept** – text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9

Status Codes

- **200 OK** – Product update form received successfully.

POST /product/update

Updates the product.

Example request:

```
POST /product/update HTTP/1.1
Host: localhost:8000
Content-Type: multipart/form-data;
```

Form Parameters

- **pk** – 778
- **basic-name** – Yamaha FS100C Acoustic Guitar
- **basic-item_type** – Black Acoustic Guitar
- **basic-category** – Musical Instrument
- **basic-description** – Top: Spruce back/sides, Mahogany neck, Nato fret board, Rosewood bridge, Snorkeling frets: 20, Open chrome tuners, Finishes: Black
- **detailed-length** – 1056
- **detailed-width** – 376
- **detailed-height** – 140
- **detailed-weight** – 3.2
- **storage-barcode** – 036000029145
- **storage-expiry** – 2040-09-29
- **status-quantity** – 20
- **status-identifier** – FS100C
- **status-location** – BAY004
- **pricing-price** – 9990
- **pricing-discount** – 7

Response Headers

- **Content-Type** – multipart/form-data; boundary=—WebKitFormBoundaryLTR88aZAnBUSE7mv

Status Codes

- **302 Found** – Redirects to /product.

`InvManage.views.delete_product_view(request, pk)`

Deletes the product with primary key `product_id` on POST request.

POST /product/<str:product_id>/delete

Deletes the consumer represented by the primary key `product_id`.

Example request:

```
POST /product/778/delete HTTP/1.1
Host: localhost:8000
Content-Type: application/x-www-form-urlencoded
```

Parameters

- **product_id** – Product primary key.

Response Headers

- **Content-Type** – application/x-www-form-urlencoded

Status Codes

- **302 Found** – Redirects to /product.
- **500 Internal Server Error** – Product matching query does not exist.

`InvManage.views.display_products_view(request)`
Retrieves the list of products on GET request.

GET /products/
Gets the list of all products.

Example request:

```
GET /products/ HTTP/1.1
Host: localhost:8000
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
```

Form Parameters

- **page** – The page number of the products list.

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: text/html; charset=utf-8
```

Request Headers

- **Accept** – text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9

Status Codes

- **200 OK** – List of products received successfully.

`InvManage.views.uploadCSV(request, data)`

Imports products into database from a .csv file on POST request. CSV template columns: name, category, item_type, description, price, quantity, identifier, location, length, width, height, weight, discount, barcode, expiry.

POST /upload/<data>/
Imports product data.

Example request:

```
POST /product/778/delete HTTP/1.1
Host: localhost:8000
```

Status Codes

- 302 Found – Redirects to /product.

```
class InvManage.views.ProductBasicInfoForm
    Bases: django.forms.Form
    Form for basic product information.

    name
        Name of the product.
        Type str

    item_type
        Type of the product.
        Type str

    category
        Product category.
        Type str

    description
        Short description of the product.
        Type str

    prefix = basic
    context
    name
    item_type
    category
    description

class InvManage.views.ProductDetailedInfoForm
    Bases: django.forms.Form
    Form for product physical details

    length
        Length of the product.
        Type str

    width
        Width of the product.
        Type str

    height
        Height of the product.
        Type str
```

weight

Weight of the product.

Type str

prefix = detailed**context****length****width****height****weight****class** InvManage.views.ProductPricingForm

Bases: django.forms.Form

Form for pricing information of the product.

price

Price of the product.

Type float

discount

Default discount percentage on the product.

Type float

prefix = pricing**context****price****discount****class** InvManage.views.ProductPurchaseEntryForm

Bases: django.forms.Form

Form for product purchase entry.

ppe_id

Unique identifier of the product purchase entry.

Type int

product

Product associated with the product purchase entry.

Type *Product*

quantity

Quantity of the product to be ordered.

Type int

price

Price of the product.

Type float

discount

Percentage discount on the product purchase.

```

        Type float
    prefix = form
    context
    ppe_id
    product
    quantity
    price
    discount

class InvManage.views.ProductSalesEntryForm
    Bases: django.forms.Form

    Form for product sales entry.

    pse_id
        Unique identifier of the product sales entry.

        Type int

    product
        Product associated with the product sales entry.

        Type Product

    quantity
        Quantity of the product.

        Type int

    price
        Price of the product.

        Type float

    discount
        Percentage discount on the product.

        Type float

    prefix = form
    context
    pse_id
    product
    quantity
    price
    discount

class InvManage.views.ProductStatusForm
    Bases: django.forms.Form

    Form for locating and identifying the product.

    quantity
        Stock quantity of the product.

        Type int

```

identifier

Unique identifier of the product.

Type str

location

Physical location of the product.

Type str

context

prefix = status

quantity

identifier

location

class InvManage.views.ProductStorageInfoForm

Bases: django.forms.Form

Form for storage information.

barcode

Barcode of the product.

Type str

expiry

Expiry date of the product.

Type DateField

prefix = storage

barcode

context

expiry

class InvManage.views.PurchaseOrderBasicInfo

Bases: django.forms.Form

Form for basic information of the purchase order.

vendor

Vendor associated with the purchase order.

Type *Vendor*

date

Date of the purchase order creation.

Type DateField

po

Purchase order number.

Type int

discount

Percentage of overall discount.

Type float

tax
 Percentage of tax.
Type float

paid
 Amount paid to the vendor.
Type float

balance
 Balance amount to be paid to the vendor.
Type float

subtotal
 Total of all the product purchase entries associated with the purchase order.
Type float

taxtotal
 Total tax applicable on the *subtotal*.
Type float

ordertotal
 Total price of the purchase order including *taxtotal*.
Type float

prefix = po

context

product_choices

vendor_choices

vendor

date

po

discount

tax

paid

balance

subtotal

taxtotal

ordertotal

class InvManage.views.GRNInfo

Bases: django.forms.Form

Form for goods receipt note (GRN).

vendor

Vendor associated with the goods receipt note.

Type *Vendor*

poRef

List of identifiers of the purchase orders from which the goods receipt note is derived.

Type *PurchaseOrder*

identifier

Unique identifier of the goods receipt note.

Type str

date

Date of GRN creation.

Type DateField

grnType

Type of GRN (auto or manual).

Type str

amendNumber

Amendment number of the GRN.

Type int

amendDate

Amendment date.

Type DateField

transporter

Name of the transport/shipping service.

Type str

vehicleNumber

Vehicle number using which the products are shipped.

Type str

gateInwardNumber

Gate inward number of the vehicle.

Type str

preparedBy

Name/identifier of the person who created the goods receipt note.

Type str

checkedBy

Name/identifier of the person who validated the goods receipt note.

Type str

checkedBy

Name/identifier of the person who inspected the physical products in the goods receipt note.

Type str

approvedBy

Name/identifier of the authority who approved the goods receipt note.

Type str

prefix = grn

```

    context
    product_choices
    vendor_choices
    vendor
    TYPE_CHOICES = [['manual', 'Blank'], ['auto', 'PO Reference']]
    poRef
    identifier
    date
    grnType
    amendNumber
    amendDate
    transporter
    vehicleNumber
    gateInwardNumber
    preparedBy
    checkedBy
    inspectedBy
    approvedBy
class InvManage.views.GRNEntryForm
    Bases: django.forms.Form
    Form for goods receipt note entry (GRNE).
    product
        Product associated with the goods receipt note entry.
        Type Product
    grne_id
        Unique identifier of the GRNE.
        Type int
    ppe_id
        Unique identifier of the product purchase entry associated with the GRNE.
        Type int
    quantity
        Ordered quantity of product with reference to product purchase entry.
        Type int
    remark
        Remarks of the quality engineer or the GRN creator about status of products received.
        Type str
    receivedQty
        Quantity of product received against the ordered quantity.

```

```

        Type str
acceptedQty
    Quantity of product accepted as OK.
        Type str
rejectedQty
    Quantity of product rejected (not OK, on HOLD, extra delivery, etc.)
        Type str
prefix = form
context
product
grne_id
ppe_id
quantity
remark
receivedQty
acceptedQty
rejectedQty
class InvManage.views.ThumbnailForm
    Bases: django.forms.Form
    For for product thumbnail image.
    image
        Image of the product.
        Type ImageField
    prefix = thumbnail
    image
class InvManage.views.SalesOrderBasicInfo
    Bases: django.forms.Form
    Form for basic information on sales order.
    consumer
        Consumer associated with the sales order.
        Type Consumer
    date
        Date of the sales order creation.
        Type DateField
    so
        Sales order number.
        Type int
    discount
        Percentage discount associated with the sales order.
```

Type float
tax
 Percentage of tax applicable.
Type float
paid
 Amount received from the consumer.
Type float
balance
 Amount balance with the consumer.
Type float
subtotal
 Total of all the product sales entries associated with the sales order.
Type float
taxtotal
 Total tax applicable on the *subtotal*.
Type float
ordertotal
 Total price of the sales order including *taxtotal*.
Type float
prefix = so
context
product_choices
consumer_choices
consumer
date
so
discount
tax
paid
balance
subtotal
taxtotal
ordertotal
class InvManage.views.ShippingAddressForm
 Bases: django.forms.Form
 For for shipping address.
title
 Title of the firm.
Type str

name

Name of the firm.

Type str**phone**

Contact number of the firm.

Type str**address**

Postal address.

Type str**city**

City.

Type str**state**

State.

Type str**country**

Country.

Type str**website**

Official website of the firm.

Type URLField**post**

Postal code of the firm.

Type str**prefix = ship****context****title****name****phone****address****city****state****country****website****post****class** InvManage.views.**VendorForm**

Bases: django.forms.Form

Form for basic informaion of the vendor.

name

Name of the vendor.

Type str**identifier**

Unique identifier of the vendor.

Type str**gstn**

GSTIN number of the vendor.

Type str**prefix = vend****context****name****identifier****gstn****class** InvManage.views.ConsumerForm

Bases: django.forms.Form

Form for consumer.

name

Name of the consumer.

Type str**identifier**

Unique identifier of the consumer.

Type str**gstn**

GSTIN number of the consumer.

Type str**phone**

Contact number.

Type str**address**

Address of the consumer.

Type str**email**

E-mail address of the consumer.

Type str**location**

City of the consumer.

Type str**prefix = consumer****context**

name
identifier
gstn
phone
address
email
location

class InvManage.views.CompanyForm

Bases: django.forms.Form

Form for company.

name
Name of the company.

Type str

owner
Name of the owner of the company.

Type str

gstn
GSTIN number of the company.

Type str

phone
Contact number of the company.

Type str

address
Postal address.

Type str

email
E-mail address of the contact person.

Type str

location
City of the company.

Type str

prefix = comp

context

name

owner

gstn

phone

address

email

location

class InvManage.views.**HistoryForm**

Bases: django.forms.Form

Form for history view.

qlen

Count of the events visible on the history view.

Type int

prefix = **history**

context

qlen

class InvManage.views.**CommunicationForm**

Bases: django.forms.Form

Form for communication information.

language

Language of communication.

Type str

phone

Contact number of the firm.

Type str

email

E-mail of the contact person.

Type EmailField

fax

Fax number.

Type str

prefix = **com**

context

language

phone

email

fax

class InvManage.views.**PurchaseDataForm**

Bases: django.forms.Form

Form for purchase related data.

currency

Currency of purchase.

Type str

minorder

Minimum order quantity.

Type int

contactperson
Name of the contact person.

Type str

refcode
Reference code.

Type str

transportmode
Mode of transport of the shipment.

Type str

prefix = **pdform**

context

currency

minorder

contactperson

refcode

transportmode

class InvManage.views.**BankAccountForm**
Bases: django.forms.Form
Form for bank account details.

name
Name of the bank.

Type str

branch
Branch number of the bank.

Type str

region
City in which the branch is located.

Type str

route
Transit number.

Type str

number
Bank account number.

Type int

accttype
Type of bank account.

Type str

iban
IBAN number.

```

        Type str
code
    Bank code.
        Type str
branchcode
    Branch code.
        Type str
prefix = bank
context
name
branch
region
route
number
acctype
iban
code
branchcode
class InvManage.views.Product
    Bases: django.db.models.Model
    Model of the product.
    name
        Name of the product.
        Type str
    category
        Product category.
        Type str
    item_type
        Type of the product.
        Type str
    description
        Short description of the product.
        Type str
    price
        Price of the product.
        Type float
    quantity
        Stock quantity of the product.
        Type int

```

identifier

Unique identifier of the product.

Type str

location

Physical location of the product.

Type str

length

Length of the product.

Type str

width

Width of the product.

Type str

height

Height of the product.

Type str

weight

Weight of the product.

Type str

discount

Default discount percentage on the product.

Type float

barcode

Barcode of the product.

Type str

expiry

Expiry date of the product.

Type DateField

image

Image of the product.

Type ImageField

name**category****item_type****description****price****quantity****identifier****location****length**

width
height
weight
discount
barcode
expiry
image
__str__(*self*)

class InvManage.views.**ProductPurchaseEntry**

Bases: `django.db.models.Model`

Model of the product purchase entry.

product

Product associated wit the product purchase entry.

Type `ModelChoiceField`

quantity

Quantity of the product to be ordered.

Type `int`

price

Price of the product.

Type `float`

discount

Percentage discount on the product purchase.

Type `float`

order

Referenced purchase order.

Type `PurchaseOrder`

receivedQty

Quantity of product received against the ordered quantity.

Type `str`

acceptedQty

Quantity of product accepted as OK.

Type `str`

rejectedQty

Quantity of product rejected (not OK, on HOLD, extra delivery, etc.)

Type `str`

is_complete()

Returns the completion status (boolean).

pending_quantity()

Returns the pending quantity against the entry (int).

product

quantity
price
discount
order
receivedQty
acceptedQty
rejectedQty
is_complete(*self*)
property pending_quantity(*self*)

class InvManage.views.GRNEntry

Bases: `django.db.models.Model`

Model of the goods receipt note entry (GRNE).

product

Primary key of the Product instance associated with the goods receipt note entry.

Type *Product*

quantity

Ordered quantity of product with reference to product purchase entry.

Type `int`

grne

Primary key of the GoodsReceiptNote instance.

Type `int`

ppe

Primary key of the ProductPurchaseEntry instance associated with the GRNE.

Type `int`

remark

Remarks of the quality engineer or the GRN creator about status of products received.

Type `str`

receivedQty

Quantity of product received against the ordered quantity.

Type `str`

acceptedQty

Quantity of product accepted as OK.

Type `str`

rejectedQty

Quantity of product rejected (not OK, on HOLD, extra delivery, etc.)

Type `str`

product

quantity

grn

ppes
remark
receivedQty
acceptedQty
rejectedQty

class InvManage.views.ProductSalesEntry
 Bases: django.db.models.Model
 Model of the product sales entry.

product
 Primary key of the Product associated with the product sales entry.
 Type *Product*

quantity
 Quantity of the product.
 Type *int*

price
 Price of the product.
 Type *float*

discount
 Percentage discount on the product.
 Type *float*

order
 Primary key of the SalesOrder referenced by the entry.
 Type *SalesOrder*

product
quantity
price
discount
order

class InvManage.views.SalesOrder
 Bases: django.db.models.Model
 Model of the sales order.

consumer
 Consumer associated with the sales order.
 Type *ModelChoiceField*

date
 Date of the sales order creation.
 Type *DateField*

so
 Sales order number.

Type int

discount

Percentage discount associated with the sales order.

Type float

tax

Percentage of tax applicable.

Type float

paid

Amount received from the consumer.

Type float

balance

Amount balance with the consumer.

Type float

subtotal

Total of all the product sales entries associated with the sales order.

Type float

taxtotal

Total tax applicable on the *subtotal*.

Type float

ordertotal

Total price of the sales order including *taxtotal*.

Type float

consumer

date

so

discount

tax

paid

balance

subtotal

taxtotal

ordertotal

class InvManage.views.**PurchaseOrder**

Bases: django.db.models.Model

Model of the purchase order.

vendor

Vendor associated with the purchase order.

Type *Vendor*

date
Date and time of the purchase order creation.
Type DateField

po
Purchase order number.
Type int

discount
Percentage discount.
Type float

tax
Percentage of tax applicable for the purchase.
Type float

paid
Amount paid against the PO.
Type float

balance
Amount balance which remains to be paid.
Type float

subtotal
Total of all the product purchase entries associated with the purchase order.
Type float

taxtotal
Total tax applicable on the *subtotal*.
Type float

ordertotal
Total price of the purchase order including *taxtotal*.
Type float

is_complete()
Returns the completion status (boolean).

pending_ppes()
Returns list of product purchase entries which are not completed.

vendor

date

po

discount

tax

paid

balance

subtotal

taxtotal

```
ordertotal
is_complete(self)
pending_ppes(self)
class InvManage.views.GoodsReceiptNote
    Bases: django.db.models.Model
    Model of the goods receipt note (GRN).
    vendor
        Vendor associated with the goods receipt note.
        Type ModelChoiceField
    poRef
        List of identifiers of the purchase orders from which the goods receipt note is derived.
        Type MultipleChoiceField
    identifier
        Unique identifier of the goods receipt note.
        Type str
    date
        Date of GRN creation.
        Type DateField
    grnType
        Type of GRN (auto or manual).
        Type ChoiceField
    amendNumber
        Amendment number of the GRN.
        Type int
    amendDate
        Amendment date.
        Type DateField
    transporter
        Name of the transport/shipping service.
        Type str
    vehicleNumber
        Vehicle number using which the products are shipped.
        Type str
    gateInwardNumber
        Gate inward number of the vehicle.
        Type str
    preparedBy
        Name/identifier of the person who created the goods receipt note.
        Type str
```

checkedBy

Name/identifier of the person who validated the goods receipt note.

Type str

checkedBy

Name/identifier of the person who inspected the physical products in the goods receipt note.

Type str

approvedBy

Name/identifier of the authority who approved the goods receipt note.

Type str

TYPE_CHOICES = [['manual', 'Blank'], ['auto', 'PO Reference']]

vendor**poRef****identifier****date****grnType****amendNumber****amendDate****transporter****vehicleNumber****gateInwardNumber****preparedBy****checkedBy****inspectedBy****approvedBy**

class InvManage.views.Consumer

Bases: django.db.models.Model

Model of the consumer.

name

Name of the consumer.

Type str

identifier

Unique identifier of the consumer.

Type str

gstIn

GSTIN number of the consumer.

Type str

phone

Contact number.

Type str

address

Address of the consumer.

Type str

email

E-mail address of the consumer.

Type str

location

City of the consumer.

Type str

Returns [description]

Return type [type]

name**identifier****gstin****phone****address****email****location**

__str__ (*self*)

class InvManage.views.**Vendor**

Bases: django.db.models.Model

Model of the vendor.

name

Name of the vendor.

Type str

identifier

Unique identifier of the vendor.

Type str

gstin

GSTIN number of the vendor.

Type str

address

Primary key of the `ShippingAddress` instance associated with the vendor.

Type *ShippingAddress*

communication

Primary key of the `Communication` instance associated with the vendor.

Type *Communication*

backaccount

Primary key of the `BankAccount` instance associated with the vendor.

Type *BankAccount*

purchasedata

Primary key of the PurchaseData instance associated with the vendor.

Type *PurchaseData*

Returns String representation of the vendor name.

Return type str

name

identifier

gstn

address

communication

bankaccount

purchasedata

__str__(self)

class InvManage.views.**Company**

Bases: django.db.models.Model

Model of the company.

name

Name of the company.

Type str

owner

Owner of the company.

Type str

gstn

GSTIN number of the company.

Type str

phone

Contact number.

Type str

address

Postal address.

Type str

email

E-mail address of the contact person.

Type str

location

City of the company.

Type str

image

Photo/logo of the company.

Type `ImageField`

shippingaddress

Primary key of the `ShippingAddress` instance associated with the company.

Type `ShippingAddress`

Returns String representation of the company name.

Return type `str`

name**owner****gsttin****phone****address****email****location****image****shippingaddress****__str__**(*self*)

class `InvManage.views.PurchaseInvoice`

Bases: `django.db.models.Model`

Model for the purchase invoice.

company

Primary key of the `Company`.

Type `Company`

po

Primary key of the `PurchaseOrder`.

Type `PurchaseOrder`

shippingaddress

Primary key of the `ShippingAddress`.

Type `ShippingAddress`

communication

Primary key of the `Communication`.

Type `Communication`

company**po****shippingaddress****communication**

class InvManage.views.SalesInvoice

Bases: django.db.models.Model

Model for the sales order invoice.

company

Primary key of the Company.

Type *Company*

so

Primary key of the SalesOrder.

Type *SalesOrder*

shippingaddress

Primary key of the ShippingAddress.

Type *ShippingAddress*

company

so

shippingaddress

class InvManage.views.FilterState

Bases: django.db.models.Model

Model of the filter state.

name

Name of the filter state.

Type str

name

class InvManage.views.FilterColumn

Bases: django.db.models.Model

Model of the column state.

name

Name of the column.

Type str

label

Label of the column.

Type str

visible

Visibility flag.

Type bool

position

Index of the column.

Type int

state

Primary key of the FilterState instance being referenced by the column instance.

Type *FilterState*

name

label

visible

position

state

class `InvManage.views.Object`

Bases: `django.db.models.Model`

Model for a generic object.

identifier

Unique identifier of the object.

Type `str`

name

Name of the object.

Type `str`

model

Model associated with the object.

Type `str`

identifier

name

model

class `InvManage.views.EventCard`

Bases: `django.db.models.Model`

Model for event card to be displayed in the history view.

obj

Object associated with the event.

Type `OneToOneField`

objId

Unique identifier of the object.

Type `str`

objname

Name of the object.

Type `str`

objmodel

Model of the object.

Type `str`

date

Timestamp of the event.

Type `DateField`

operation

Type of event.


```

        Type str

    obj
    objId
    objname
    objmodel
    date
    operation

class InvManage.views.EventType
    Bases: django.db.models.Model

    Model for event type.

    name
        Name of the event being triggered.

        Type str

    label
        Label associated with the event.

        Type str

    Returns Returns the string representation of the label when the event type is queried.

    Return type Label

    name
    label
    __str__(self)

class InvManage.views.ObjectModel
    Bases: django.db.models.Model

    Model associated with an object.

    name
        Name of the object.

        Type str

    label
        Label string of the object.

        Type str

    modName
        Name of the model associated with the object.

        Type str

    Returns Returns string representation of the label when object model is queried.

    Return type Label

    name
    label

```

modName

__str__(*self*)

class InvManage.views.**HistoryFilterState**

Bases: django.db.models.Model

Model of filter state of the history.

params

Paramameters of the configured filters.

Type str

params

class InvManage.views.**ShippingAddress**

Bases: django.db.models.Model

Model of the shipping address.

title

Title of the firm.

Type str

name

Name of the firm.

Type str

phone

Contact number of the firm.

Type str

address

Postal address.

Type str

city

City.

Type str

state

State.

Type str

country

Country.

Type str

website

Official website of the firm.

Type URLField

post

Postal code of the firm.

Type str

title

name
phone
address
city
state
country
website
post

class InvManage.views.**PurchaseData**

Bases: django.db.models.Model

Model of the purchase data.

currency

Currency of purchase.

Type str

minorder

Minimum order quantity.

Type int

contactperson

Name of the contact person.

Type str

refcode

Reference code.

Type str

transportmode

Mode of transport of the shipment.

Type str

currency

minorder

contactperson

refcode

transportmode

class InvManage.views.**Communication**

Bases: django.db.models.Model

Model of the communication information.

language

Language of communication.

Type str

phone

Contact number of the firm.

```
        Type str
email
    E-mail of the contact person.
        Type EmailField
fax
    Fax number.
        Type str
language
phone
email
fax
class InvManage.views.BankAccount
    Bases: django.db.models.Model
    Model of the bank account details.
    name
        Name of the bank.
        Type str
    branch
        Branch number of the bank.
        Type str
    region
        City in which the branch is located.
        Type str
    route
        Transit number.
        Type str
    number
        Bank account number.
        Type int
    accttype
        Type of bank account.
        Type str
    iban
        IBAN number.
        Type str
    code
        Bank code.
        Type str
    branchcode
        Branch code.
```

Type str

name
branch
region
route
number
accttype
iban
code
branchcode

`InvManage.views.get_columns(state)`

Gets the columns of the display table.

This function takes a filter state and returns an array of column names that are ordered depending on the position attribute of the columns. This order is important as the queryset will be rendered in the display table according to the column order in this returned array.

Parameters `state` (`FilterState`) – Filter state of the columns.

Returns A list that dictates the order of the columns.

Return type List

`InvManage.views.sort_ascending_descending(request, model)`

Sorts and saves the state of the column.

This function takes a request, finds which column needs to be sorted in ascending/descending order, and returns the sorted queryset.

Parameters

- **request** (`HttpRequest`) – GET request received for retrieving the list of model instances.
- **model** (`Model`) – Model of the objects being retrieved.

Returns List of sorted object instances.

Return type QuerySet

`InvManage.views.change_column_position(request, state)`

Modifies the column states depending on the request.

This function:

- Takes a filter state
- Extracts the column names into an order array
- Modifies the column order in the array depending on user input direction (left/right)
- Saves this modified state of columns into the database
- Returns the modified column names array

Parameters

- **request** (`HttpRequest`) – GET request for retrieving the state of the columns.
- **state** (`FilterState`) – Filter state of the columns.

Returns List of sorted columns.

Return type List

`InvManage.views.paginate(queryset, filter, page_number)`

Paginates the list of objects.

This function takes the entire queryset and filters out only objects belonging to the request page

Parameters

- **queryset** (*QuerySet*) – QuerySet of the objects retrieved from the database.
- **filter** (*FilterState*) – Filter state of the columns.
- **page_number** (*int*) – Page number requested.

class `InvManage.views.GoodsReceiptNoteFilter`

Bases: `django_filters.FilterSet`

class `Meta`

model

fields

context

identifier

vendor

date

poRef

grnType

amendNumber

amendDate

vehicleNumber

gateInwardNumber

preparedBy

checkedBy

inspectedBy

approvedBy

ordering

class `InvManage.views.GRNEntrySerializer`

Bases: `rest_framework.serializers.ModelSerializer`

Serializer for GRNEntry class instance.

The `GRNEntrySerializer.data` attribute gives the JSON serialized data of the GRNEntry instance:

```
{
  "grn": 103,
  "grne_id": 117,
```

(continues on next page)

(continued from previous page)

```

    "product": {
        "pk": 637,
        "name": "piano",
        "category": "Ultricies PC",
        "quantity": 23921,
        "identifier": "PROD9",
        "location": "Musselburgh",
        "description": "88-key, Tri-sensor Scaled Hammer Action Keyboard II, ↵
↳ Simulated ebony and ivory keys ",
        "prod_id": 637
    },
    "quantity": 100,
    "receivedQty": 50,
    "acceptedQty": 50,
    "rejectedQty": 0,
    "remark": "OK"
}

```

```
class Meta
```

```
    model
```

```
    fields = ['grn', 'grne_id', 'product', 'quantity', 'receivedQty', 'acceptedQty',
              'rejectedQty', 'remark']
```

```
grne_id
```

```
product
```

```
grn
```

```
to_representation(self, instance)
```

```
create(self, validated_data)
```

```
update(self, instance, validated_data)
```

```
class InvManage.views.GRNEntryWithPORefSerializer
```

```
    Bases: rest_framework.serializers.ModelSerializer
```

Serializer for GRNEntry class instance if the GRN references a purchase order.

The GRNEntryWithPORefSerializer.data attribute gives the JSON serialized data of the GRNEntry instance:

```

{
    "grn": 103,
    "grne_id": 117,
    "ppe_id": 324,
    "po_id": 293,
    "product": {
        "pk": 637,
        "name": "piano",
        "category": "Ultricies PC",
        "quantity": 23921,
        "identifier": "PROD9",
        "location": "Musselburgh",

```

(continues on next page)

(continued from previous page)

```

        "description": "88-key, Tri-sensor Scaled Hammer Action Keyboard II, ↵
↵Simulated ebony and ivory keys ",
        "prod_id": 637
    },
    "quantity": 100,
    "receivedQty": 50,
    "acceptedQty": 50,
    "rejectedQty": 0,
    "remark": "OK"
}

```

class Meta

model

fields = ['grn', 'grne_id', 'ppe_id', 'po_id', 'product', 'quantity',
'receivedQty', 'acceptedQty',...]

grne_id

ppe_id

po_id

product

grn

to_representation(self, instance)

create(self, validated_data)

update(self, instance, validated_data)

class InvManage.views.GRNInvoiceSerializer

Bases: rest_framework.serializers.ModelSerializer

Serializer for GRNInvoice class instance.

The GRNInvoiceSerializer.data attribute gives the JSON serialized data of the GRNInvoice instance:

```

{
  "company": {
    "name": "Fringillami",
    "owner": "Ivor Barnett",
    "gstin": "89AAC056465468",
    "phone": "332 220-7026",
    "address": "Ap #849-6241 Euismod Av., 677598, Carinthia, Belgium",
    "email": "est.tempor@fringillami.org",
    "location": "Belgium",
    "image": "/media/images/logo.png"
  },
  "grn": {
    "grnes": [
      {
        "grn": 103,
        "grne_id": 117,

```

(continues on next page)

(continued from previous page)

```

        "ppe_id": 324,
        "po_id": 293,
        "product": {
            "pk": 637,
            "name": "piano",
            "category": "Ultricies PC",
            "quantity": 23921,
            "identifier": "PROD9",
            "location": "Musselburgh",
            "description": "88-key, Tri-sensor Scaled Hammer Action_
↪Keyboard II, Simulated ebony and ivory keys ",
            "prod_id": 637
        },
        "quantity": 100,
        "receivedQty": 50,
        "acceptedQty": 50,
        "rejectedQty": 0,
        "remark": "OK"
    },
    {
        "grn": 103,
        "grne_id": 118,
        "ppe_id": 325,
        "po_id": 293,
        "product": {
            "pk": 645,
            "name": "Tabl",
            "category": "Sociis Natoque Company",
            "quantity": 38276,
            "identifier": "PROD17",
            "location": "Schagen",
            "description": "aldgjlakjlkasdj",
            "prod_id": 645
        },
        "quantity": 250,
        "receivedQty": 200,
        "acceptedQty": 180,
        "rejectedQty": 20,
        "remark": "20 pieces faulty"
    }
],
"date": "29-Sep-2021",
"vendor": {
    "name": "Girish",
    "identifier": "GJ",
    "gstn": "GSTIN002",
    "address": {
        "name": "alsf",
        "address": "jas;k",
        "city": ";sdalkf",
        "phone": "alsf",
        "state": "kjdfmk",

```

(continues on next page)

(continued from previous page)

```

        "country": "ljflkj",
        "post": "54545"
    },
    "poRef": [
        182
    ],
    "identifier": 846,
    "grnType": "auto",
    "amendDate": "2021-09-29T00:00:00Z",
    "transporter": "TeraTransport",
    "vehicleNumber": "GH-646358",
    "gateInwardNumber": "864353",
    "preparedBy": "KJL",
    "checkedBy": "KJH",
    "inspectedBy": "GIO",
    "approvedBy": "BHI"
},
"shippingaddress": {
    "name": "Harding Gross",
    "address": "8798 At, St., 7639",
    "city": "Rome",
    "phone": "936 651-4847",
    "state": "Lazio",
    "country": "Italy",
    "post": "300326"
},
"communication": {
    "email": "asfs@aflk.com",
    "phone": "6546432"
}
}

```

```
class Meta
```

```
    model
```

```
    fields = ['company', 'grn', 'shippingaddress', 'communication']
```

```
company
```

```
grn
```

```
shippingaddress
```

```
communication
```

```
class InvManage.views.GoodsReceiptNoteSerializer
```

```
    Bases: rest_framework.serializers.ModelSerializer
```

```
    Serializer for GoodsReceiptNote class instance.
```

```
    The GoodsReceiptNoteSerializer.data attribute gives the JSON serialized data of the GoodsReceiptNote instance:
```

```

{
  "grnes": [
    {
      "grn": 103,
      "grne_id": 117,
      "ppe_id": 324,
      "po_id": 293,
      "product": {
        "pk": 637,
        "name": "piano",
        "category": "Ultricies PC",
        "quantity": 23921,
        "identifier": "PROD9",
        "location": "Musselburgh",
        "description": "88-key, Tri-sensor Scaled Hammer Action Keyboard II,
→ Simulated ebony and ivory keys ",
        "prod_id": 637
      },
      "quantity": 100,
      "receivedQty": 50,
      "acceptedQty": 50,
      "rejectedQty": 0,
      "remark": "OK"
    },
    {
      "grn": 103,
      "grne_id": 118,
      "ppe_id": 325,
      "po_id": 293,
      "product": {
        "pk": 645,
        "name": "Tabl",
        "category": "Sociis Natoque Company",
        "quantity": 38276,
        "identifier": "PROD17",
        "location": "Schagen",
        "description": "aldgjlakjlkasdj",
        "prod_id": 645
      },
      "quantity": 250,
      "receivedQty": 200,
      "acceptedQty": 180,
      "rejectedQty": 20,
      "remark": "20 pieces faulty"
    }
  ],
  "date": "29-Sep-2021",
  "vendor": {
    "name": "Girish",
    "identifier": "GJ",
    "gstn": "GSTIN002",
    "address": {
      "name": "alsf",

```

(continues on next page)

(continued from previous page)

```

        "address": "jas;k",
        "city": ";sdalkf",
        "phone": "alsf",
        "state": "kjdfk",
        "country": "ljflkj",
        "post": "54545"
    },
    "poRef": [
        182
    ],
    "identifier": 846,
    "grnType": "auto",
    "amendDate": "2021-09-29T00:00:00Z",
    "transporter": "TeraTransport",
    "vehicleNumber": "GH-646358",
    "gateInwardNumber": "864353",
    "preparedBy": "KJL",
    "checkedBy": "KJH",
    "inspectedBy": "GIO",
    "approvedBy": "BHI"
}

```

class Meta

model

fields = ['grnes', 'date', 'vendor', 'poRef', 'identifier', 'grnType',
'amendDate', 'transporter',...]

grnes

date

vendor

to_representation(self, instance)

get_grnes(self, instance)

class InvManage.views.PPEntrySerializer

Bases: rest_framework.serializers.ModelSerializer

Serializer for ProductPurchaseEntry class instance.

The PPEntrySerializer.data attribute gives the JSON serialized data of the ProductPurchaseEntry instance:

```

{
    "ppe_id": 324,
    "product": {
        "pk": 637,
        "name": "Piano",
        "category": "Ultricies PC",
        "quantity": 23921,
        "identifier": "PROD9",
    }
}

```

(continues on next page)

(continued from previous page)

```

        "location": "Musselburgh",
        "description": "88-key, Tri-sensor Scaled Hammer Action Keyboard II, ↵
↵Simulated ebony and ivory keys ",
        "prod_id": 637
    },
    "quantity": 100,
    "price": 10.0,
    "discount": 0.0,
    "order": 182,
    "pendingQty": 50
}

```

class Meta

model

fields = ['ppe_id', 'product', 'quantity', 'price', 'discount', 'order',
'pendingQty']

ppe_id

product

pendingQty

to_representation(self, instance)

create(self, validated_data)

update(self, instance, validated_data)

class InvManage.views.ProductSerializer

Bases: rest_framework.serializers.ModelSerializer

Serializer for Product class instance.

The ProductSerializer.data attribute gives the JSON serialized data of the Product instance:

```

{
    "pk": 637,
    "name": "Piano",
    "category": "Ultricies PC",
    "quantity": 23921,
    "identifier": "PROD9",
    "location": "Musselburgh",
    "description": "88-key, Tri-sensor Scaled Hammer Action Keyboard II, ↵
↵Simulated ebony and ivory keys ",
    "prod_id": 637
}

```

class Meta

model

fields = ['pk', 'name', 'category', 'quantity', 'identifier', 'location',
'description', 'prod_id']

prod_id

`to_representation(self, instance)`

class `InvManage.views.PurchaseInvoiceSerializer`

Bases: `rest_framework.serializers.ModelSerializer`

Serializer for `PurchaseInvoice` class instance.

The `PurchaseInvoiceSerializer.data` attribute gives the JSON serialized data of the `PurchaseInvoice` instance:

```
{
  "company": {
    "name": "Fringillami",
    "owner": "Ivor Barnett",
    "gstn": "89AAC056465468",
    "phone": "332 220-7026",
    "address": "Ap #849-6241 Euismod Av., 677598, Carinthia, Belgium",
    "email": "est.tempor@fringillami.org",
    "location": "Belgium",
    "image": "/media/images/logo.png"
  },
  "po": {
    "vendor": {
      "name": "Girish",
      "identifier": "GJ",
      "gstn": "GSTIN002",
      "address": {
        "name": "alsf",
        "address": "jas;k",
        "city": ";sdalkf",
        "phone": "alsf",
        "state": "kjdfk",
        "country": "ljflkj",
        "post": "54545"
      }
    },
    "date": "25-Sep-2021",
    "po": 293,
    "subtotal": 279975.0,
    "taxtotal": 22398.0,
    "ordertotal": 302373.0,
    "ppes": [
      {
        "ppe_id": 324,
        "product": {
          "pk": 637,
          "name": "piano",
          "category": "Ultricies PC",
          "quantity": 23921,
          "identifier": "PROD9",
          "location": "Musselburgh",
          "description": "88-key, Tri-sensor Scaled Hammer Action ↵
↵Keyboard II, Simulated ebony and ivory keys ",
          "prod_id": 637
        }
      }
    ]
  }
}
```

(continues on next page)

(continued from previous page)

```

    "quantity": 100,
    "price": 10.0,
    "discount": 0.0,
    "order": 182,
    "pendingQty": 50
  },
  {
    "ppe_id": 325,
    "product": {
      "pk": 645,
      "name": "Tabl",
      "category": "Sociis Natoque Company",
      "quantity": 38276,
      "identifier": "PROD17",
      "location": "Schagen",
      "description": "aldgjlakjlkasdj",
      "prod_id": 645
    },
    "quantity": 250,
    "price": 90.0,
    "discount": 8.0,
    "order": 182,
    "pendingQty": 70
  },
  {
    "ppe_id": 326,
    "product": {
      "pk": 638,
      "name": "Goblet drum",
      "category": "Est Congue Consulting",
      "quantity": 46076,
      "identifier": "PROD10",
      "location": "Kaluga",
      "description": "aldgjlakjlkasdj",
      "prod_id": 638
    },
    "quantity": 200,
    "price": 150.0,
    "discount": 12.0,
    "order": 182,
    "pendingQty": 200
  },
  {
    "ppe_id": 327,
    "product": {
      "pk": 643,
      "name": "quinto",
      "category": "Enim Suspendisse Associates",
      "quantity": 51099,
      "identifier": "PROD15",
      "location": "Nagpur",
      "description": "aldgjlakjlkasdj",

```

(continues on next page)

(continued from previous page)

```
        "prod_id": 643
    },
    "quantity": 350,
    "price": 150.0,
    "discount": 10.0,
    "order": 182,
    "pendingQty": 350
},
{
    "ppe_id": 328,
    "product": {
        "pk": 651,
        "name": "Igihumurizo",
        "category": "Curabitur Massa Vestibulum Consulting",
        "quantity": 48553,
        "identifier": "PROD23",
        "location": "Yeovil",
        "description": "aldgjlakjlkasdj",
        "prod_id": 651
    },
    "quantity": 500,
    "price": 50.0,
    "discount": 6.0,
    "order": 182,
    "pendingQty": 500
},
{
    "ppe_id": 329,
    "product": {
        "pk": 653,
        "name": "Balsié",
        "category": "Eu Foundation",
        "quantity": 29988,
        "identifier": "PROD25",
        "location": "Sudbury",
        "description": "aldgjlakjlkasdj",
        "prod_id": 653
    },
    "quantity": 200,
    "price": 500.0,
    "discount": 15.0,
    "order": 182,
    "pendingQty": 200
},
{
    "ppe_id": 330,
    "product": {
        "pk": 656,
        "name": "Padayani thappu",
        "category": "Posuere LLP",
        "quantity": 6358,
        "identifier": "PROD28",
```

(continues on next page)

(continued from previous page)

```

        "location": "Ruvo del Monte",
        "description": "aldgjlakjlkasdj",
        "prod_id": 656
    },
    "quantity": 350,
    "price": 250.0,
    "discount": 13.0,
    "order": 182,
    "pendingQty": 350
}
]
},
"shippingaddress": {
    "name": "Harding Gross",
    "address": "8798 At, St., 7639",
    "city": "Rome",
    "phone": "936 651-4847",
    "state": "Lazio",
    "country": "Italy",
    "post": "300326"
},
"communication": {
    "email": "asfs@aflk.com",
    "phone": "6546432"
}
}

```

class Meta

model

fields = ['company', 'po', 'shippingaddress', 'communication']

company

po

shippingaddress

communication

class InvManage.views.GRNInvoice

Bases: django.db.models.Model

Model for the goods receipt note invoice.

company

Primary key of the Company.

Type *Company*

grn

Primary key of the GoodsReceiptNote.

Type *GoodsReceiptNote*

shippingaddress

Primary key of the ShippingAddress.

Type *ShippingAddress*

communication

Primary key of the Communication.

Type *Communication*

company

grn

shippingaddress

communication

`InvManage.views.create_grn_view(request)`

Creates a goods receipt note (GRN) on POST request, and returns a GRN creation form on GET request.

GET /grn

Gets the GRN creation form.

Example request:

```
GET /grn/ HTTP/1.1
Host: localhost:8000
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: text/html; charset=utf-8
```

Request Headers

- **Accept** – text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9

Status Codes

- **200 OK** – GRN creation form received successfully.

POST /grn

Creates a GRN. There are two modes: - auto : Setting `grn-grnType` to auto creates a GRN with PO reference. - manual : Setting `grn-grnType` to manual creates a GRN without PO reference. In this case

the `grn-poRef` is not required.

Example request:

```
POST /consumer/ HTTP/1.1
Host: localhost:8000
Content-Type: multipart/form-data;
```

Form Parameters

- **grn-grnType** – auto
- **grn-vendor** – 4
- **grn-poRef** – 182

- grn-amendNumber – 546
- grn-amendDate – 2021-09-29
- grn-identifier – 846
- grn-date – 2021-09-29
- grn-transporter – TeraTransport
- grn-vehicleNumber – GH-646358
- grn-gateInwardNumber – 864353
- grn-preparedBy – KJL
- grn-checkedBy – KJH
- grn-inspectedBy – GIO
- grn-approvedBy – BHI
- form-TOTAL_FORMS – 3
- form-INITIAL_FORMS – 0
- form-MIN_NUM_FORMS –
- form-MAX_NUM_FORMS –
- form-0-product – 637
- form-0-quantity – 100
- form-0-receivedQty – 50
- form-0-acceptedQty – 50
- form-0-rejectedQty – 0
- form-0-remark – OK
- form-0-DELETE –
- form-0-grne_id – -1
- form-0-ppe_id – 324
- form-1-product – 645
- form-1-quantity – 250
- form-1-receivedQty – 200
- form-1-acceptedQty – 180
- form-1-rejectedQty – 20
- form-1-remark – 20 pieces faulty
- form-1-DELETE –
- form-1-grne_id – -1
- form-1-ppe_id – 325
- form-2-product – 638
- form-2-quantity – 200
- form-2-receivedQty – 0

- **form-2-acceptedQty** – 0
- **form-2-rejectedQty** – 0
- **form-2-remark** –
- **form-2-DELETE** –
- **form-2-grne_id** – -1
- **form-2-ppe_id** – 326

Response Headers

- **Content-Type** – multipart/form-data; boundary=—WebKitFormBoundaryLTR88aZAnBUSE7mv

Status Codes

- **302 Found** – Redirects to /grn.

`InvManage.views.update_grn_view(request)`

Updates the goods receipt note (GRN) on POST request, and returns a GRN update form on GET request.

GET /grn/update

Gets the GRN update form.

Example request:

```
GET /grn/update HTTP/1.1
Host: localhost:8000
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: text/html; charset=utf-8
```

Request Headers

- **Accept** – text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9

Status Codes

- **200 OK** – GRN update form received successfully.

POST /grn/update

Updates a GRN. There are two modes: - auto : Setting `grn-grnType` to auto creates a GRN with PO reference. - manual : Setting `grn-grnType` to manual creates a GRN without PO reference. In this case

the `grn-poRef` is not required.

Example request:

```
POST /consumer/ HTTP/1.1
Host: localhost:8000
Content-Type: multipart/form-data;
```

Form Parameters

- grn-grnType – auto
- grn-vendor – 4
- grn-poRef – 182
- grn-amendNumber – 546
- grn-amendDate – 2021-09-29
- grn-identifier – 846
- grn-date – 2021-09-29
- grn-transporter – TeraTransport
- grn-vehicleNumber – GH-646358
- grn-gateInwardNumber – 864353
- grn-preparedBy – KJL
- grn-checkedBy – KJH
- grn-inspectedBy – GIO
- grn-approvedBy – BHI
- form-TOTAL_FORMS – 3
- form-INITIAL_FORMS – 0
- form-MIN_NUM_FORMS –
- form-MAX_NUM_FORMS –
- form-0-product – 637
- form-0-quantity – 100
- form-0-receivedQty – 50
- form-0-acceptedQty – 50
- form-0-rejectedQty – 0
- form-0-remark – OK
- form-0-DELETE –
- form-0-grne_id – -1
- form-0-ppe_id – 324
- form-1-product – 645
- form-1-quantity – 250
- form-1-receivedQty – 200
- form-1-acceptedQty – 180
- form-1-rejectedQty – 20
- form-1-remark – 20 pieces faulty
- form-1-DELETE –
- form-1-grne_id – -1
- form-1-ppe_id – 325

- **form-2-product** – 638
- **form-2-quantity** – 200
- **form-2-receivedQty** – 0
- **form-2-acceptedQty** – 0
- **form-2-rejectedQty** – 0
- **form-2-remark** –
- **form-2-DELETE** –
- **form-2-grne_id** – -1
- **form-2-ppe_id** – 326

Response Headers

- **Content-Type** – multipart/form-data; boundary=——WebKitFormBoundaryLTR88aZAnBUSE7mv

Status Codes

- **302 Found** – Redirects to /grn.

`InvManage.views.delete_grn_view(request, pk)`

Deletes the GRN with primary key `object_id` on POST request.

POST /grn/<str:object_id>/delete

Deletes the GRN represented by the primary key `object_id`.

Example request:

```
POST /grn/104/delete HTTP/1.1
Host: localhost:8000
Content-Type: application/x-www-form-urlencoded
```

Parameters

- **object_id** – GRN primary key.

Response Headers

- **Content-Type** – application/x-www-form-urlencoded

Status Codes

- **302 Found** – Redirects to /grn.
- **500 Internal Server Error** – GRN matching query does not exist.

`InvManage.views.display_grns_view(request)`

Retrieves the list of GRNs on GET request.

GET /grns/

Gets the list of all GRNs.

Example request:

```
GET /grns/ HTTP/1.1
Host: localhost:8000
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
```

Form Parameters

- **page** – The page number of the GRN list.

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: text/html; charset=utf-8
```

Request Headers

- **Accept** – text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9

Status Codes

- **200 OK** – List of GRNs received successfully.

`InvManage.views.print_grn_view(request, pk)`

Opens the GRN with primary key `grn_id` for printing on GET request.

POST /grn/<str:grn_id>/print

Opens the GRN represented by the primary key `grn_id`.

Example request:

```
POST /grn/103/print HTTP/1.1
Host: localhost:8000
Content-Type: application/x-www-form-urlencoded
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json; charset=utf-8

[
  {
    "company": {
      "name": "Fringillami",
      "owner": "Ivor Barnett",
      "gst_in": "89AAC056465468",
      "phone": "332 220-7026",
      "address": "Ap #849-6241 Euismod Av., 677598, Carinthia, Belgium",
      "email": "est.tempor@fringillami.org",
      "location": "Belgium",
      "image": "/media/images/hyperlink-green_x91WW5n.png"
    },
    "grn": {
      "grnes": [
        {
          "grn": 103,
          "grne_id": 117,
          "ppe_id": 324,
          "po_id": 293,
```

(continues on next page)

(continued from previous page)

```

        "product": {
          "pk": 637,
          "name": "piano",
          "category": "Ultricies PC",
          "quantity": 23921,
          "identifier": "PROD9",
          "location": "Musselburgh",
          "description": "88-key, Tri-sensor Scaled Hammer Action_
Keyboard II, Simulated ebony and ivory keys ",
          "prod_id": 637
        },
        "quantity": 100,
        "receivedQty": 50,
        "acceptedQty": 50,
        "rejectedQty": 0,
        "remark": "OK"
      },
      {
        "grn": 103,
        "grne_id": 118,
        "ppe_id": 325,
        "po_id": 293,
        "product": {
          "pk": 645,
          "name": "Tabl",
          "category": "Sociis Natoque Company",
          "quantity": 38276,
          "identifier": "PROD17",
          "location": "Schagen",
          "description": "aldgjllakjlkasdj",
          "prod_id": 645
        },
        "quantity": 250,
        "receivedQty": 200,
        "acceptedQty": 180,
        "rejectedQty": 20,
        "remark": "20 pieces faulty"
      }
    ],
    "date": "29-Sep-2021",
    "vendor": {
      "name": "Girish",
      "identifier": "GJ",
      "gstn": "GSTIN002",
      "address": {
        "name": "alsf",
        "address": "jas;k",
        "city": ";sdalkf",
        "phone": "alsf",
        "state": "kjdfk",
        "country": "ljflkj",
        "post": "54545"
      }
    }
  }

```

(continues on next page)

(continued from previous page)

```

    },
    "poRef": [
        182
    ],
    "identifier": 846,
    "grnType": "auto",
    "amendDate": "2021-09-29T00:00:00Z",
    "transporter": "TeraTransport",
    "vehicleNumber": "GH-646358",
    "gateInwardNumber": "864353",
    "preparedBy": "KJL",
    "checkedBy": "KJH",
    "inspectedBy": "GIO",
    "approvedBy": "BHI"
},
"shippingaddress": {
    "name": "Harding Gross",
    "address": "8798 At, St., 7639",
    "city": "Rome",
    "phone": "936 651-4847",
    "state": "Lazio",
    "country": "Italy",
    "post": "300326"
},
"communication": {
    "email": "asfs@aflk.com",
    "phone": "6546432"
}
}
]

```

Parameters

- **grn_id** – GRN primary key.

Response Headers

- **Content-Type** – application/json

Status Codes

- **200 OK** – GRN print request successful.
- **500 Internal Server Error** – GRN matching query does not exist.

class InvManage.views.ProductBasicInfoForm

Bases: django.forms.Form

Form for basic product information.

name

Name of the product.

Type str

item_type

Type of the product.

Type str**category**

Product category.

Type str**description**

Short description of the product.

Type str**prefix = basic****context****name****item_type****category****description****class** InvManage.views.ProductDetailedInfoForm

Bases: django.forms.Form

Form for product physical details

length

Length of the product.

Type str**width**

Width of the product.

Type str**height**

Height of the product.

Type str**weight**

Weight of the product.

Type str**prefix = detailed****context****length****width****height****weight****class** InvManage.views.ProductPricingForm

Bases: django.forms.Form

Form for pricing information of the product.

price
Price of the product.

Type float

discount
Default discount percentage on the product.

Type float

prefix = pricing

context

price

discount

class InvManage.views.ProductPurchaseEntryForm

Bases: django.forms.Form

Form for product purchase entry.

ppe_id
Unique identifier of the product purchase entry.

Type int

product
Product associated wit the product purchase entry.

Type *Product*

quantity
Quantity of the product to be ordered.

Type int

price
Price of the product.

Type float

discount
Percentage discount on the product purchase.

Type float

prefix = form

context

ppe_id

product

quantity

price

discount

class InvManage.views.ProductSalesEntryForm

Bases: django.forms.Form

Form for product sales entry.

pse_id

Unique identifier of the product sales entry.

Type int

product

Product associated with the product sales entry.

Type *Product*

quantity

Quantity of the product.

Type int

price

Price of the product.

Type float

discount

Percentage discount on the product.

Type float

prefix = form

context

pse_id

product

quantity

price

discount

class InvManage.views.ProductStatusForm

Bases: django.forms.Form

Form for locating and identifying the product.

quantity

Stock quantity of the product.

Type int

identifier

Unique identifier of the product.

Type str

location

Physical location of the product.

Type str

context

prefix = status

quantity

identifier

location

```

class InvManage.views.ProductStorageInfoForm
    Bases: django.forms.Form

    Form for storage information.

    barcode
        Barcode of the product.

        Type str

    expiry
        Expiry date of the product.

        Type DateField

    prefix = storage

    barcode
    context
    expiry

class InvManage.views.PurchaseOrderBasicInfo
    Bases: django.forms.Form

    Form for basic information of the purchase order.

    vendor
        Vendor associated with the purchahse order.

        Type Vendor

    date
        Date of the purchase order creation.

        Type DateField

    po
        Purchase order number.

        Type int

    discount
        Percentage of overall discount.

        Type float

    tax
        Percentage of tax.

        Type float

    paid
        Amount paid to the vendor.

        Type float

    balance
        Balance amount to be paid to the vendor.

        Type float

    subtotal
        Total of all the product purchase entries associated with the purchase order.

        Type float

```

taxtotal

Total tax applicable on the *subtotal*.

Type float

ordertotal

Total price of the purchase order including *taxtotal*.

Type float

prefix = po**context****product_choices****vendor_choices****vendor****date****po****discount****tax****paid****balance****subtotal****taxtotal****ordertotal****class** InvManage.views.GRNInfo

Bases: django.forms.Form

Form for goods receipt note (GRN).

vendor

Vendor associated with the goods receipt note.

Type *Vendor*

poRef

List of identifiers of the purchase orders from which the goods receipt note is derived.

Type *PurchaseOrder*

identifier

Unique identifier of the goods receipt note.

Type str

date

Date of GRN creation.

Type DateField

grnType

Type of GRN (auto or manual).

Type str

amendNumber

Amendment number of the GRN.

Type int

amendDate

Amendment date.

Type DateField

transporter

Name of the transport/shipping service.

Type str

vehicleNumber

Vehicle number using which the products are shipped.

Type str

gateInwardNumber

Gate inward number of the vehicle.

Type str

preparedBy

Name/identifier of the person who created the goods receipt note.

Type str

checkedBy

Name/identifier of the person who validated the goods receipt note.

Type str

checkedBy

Name/identifier of the person who inspected the physical products in the goods receipt note.

Type str

approvedBy

Name/identifier of the authority who approved the goods receipt note.

Type str

prefix = grn**context****product_choices****vendor_choices****vendor**

TYPE_CHOICES = [['manual', 'Blank'], ['auto', 'PO Reference']]

poRef**identifier****date****grnType****amendNumber****amendDate**

transporter

vehicleNumber

gateInwardNumber

preparedBy

checkedBy

inspectedBy

approvedBy

class `InvManage.views.GRNEnterForm`

Bases: `django.forms.Form`

Form for goods receipt note entry (GRNE).

product

Product associated with the goods receipt note entry.

Type *Product*

grne_id

Unique identifier of the GRNE.

Type `int`

ppe_id

Unique identifier of the product purchase entry associated with the GRNE.

Type `int`

quantity

Ordered quantity of product with reference to product purchase entry.

Type `int`

remark

Remarks of the quality engineer or the GRN creator about status of products received.

Type `str`

receivedQty

Quantity of product received against the ordered quantity.

Type `str`

acceptedQty

Quantity of product accepted as OK.

Type `str`

rejectedQty

Quantity of product rejected (not OK, on HOLD, extra delivery, etc.)

Type `str`

prefix = `form`

context

product

grne_id

ppe_id


```

    quantity
    remark
    receivedQty
    acceptedQty
    rejectedQty
class InvManage.views.ThumbnailForm
    Bases: django.forms.Form
    For for product thumbnail image.
    image
        Image of the product.
        Type ImageField
    prefix = thumbnail
    image
class InvManage.views.SalesOrderBasicInfo
    Bases: django.forms.Form
    Form for basic information on sales order.
    consumer
        Consumer associated with the sales order.
        Type Consumer
    date
        Date of the sales order creation.
        Type DateField
    so
        Sales order number.
        Type int
    discount
        Percentage discount associated with the sales order.
        Type float
    tax
        Percentage of tax applicable.
        Type float
    paid
        Amount received from the consumer.
        Type float
    balance
        Amount balance with the consumer.
        Type float
    subtotal
        Total of all the product sales entries associated with the sales order.
        Type float

```

taxtotalTotal tax applicable on the *subtotal*.**Type** float**ordertotal**Total price of the sales order including *taxtotal*.**Type** float**prefix = so****context****product_choices****consumer_choices****consumer****date****so****discount****tax****paid****balance****subtotal****taxtotal****ordertotal****class** InvManage.views.ShippingAddressForm

Bases: django.forms.Form

For for shipping address.

title

Title of the firm.

Type str**name**

Name of the firm.

Type str**phone**

Contact number of the firm.

Type str**address**

Postal address.

Type str**city**

City.

Type str

```

    state
        State.

        Type str
    country
        Country.

        Type str
    website
        Official website of the firm.

        Type URLField
    post
        Postal code of the firm.

        Type str
    prefix = ship
    context
    title
    name
    phone
    address
    city
    state
    country
    website
    post
class InvManage.views.VendorForm
    Bases: django.forms.Form
    Form for basic informaion of the vendor.
    name
        Name of the vendor.

        Type str
    identifier
        Unique identifier of the vendor.

        Type str
    gstln
        GSTIN number of the vendor.

        Type str
    prefix = vend
    context
    name
    identifier

```

gstn

class InvManage.views.ConsumerForm

Bases: django.forms.Form

Form for consumer.

name

Name of the consumer.

Type str

identifier

Unique identifier of the consumer.

Type str

gstn

GSTIN number of the consumer.

Type str

phone

Contact number.

Type str

address

Address of the consumer.

Type str

email

E-mail address of the consumer.

Type str

location

City of the consumer.

Type str

prefix = consumer

context

name

identifier

gstn

phone

address

email

location

class InvManage.views.CompanyForm

Bases: django.forms.Form

Form for company.

name

Name of the company.

```

        Type str
owner
    Name of the owner of the company.
        Type str
gstn
    GSTIN number of the company.
        Type str
phone
    Contact number of the company.
        Type str
address
    Postal address.
        Type str
email
    E-mail address of the contact person.
        Type str
location
    City of the company.
        Type str
prefix = comp
context
name
owner
gstn
phone
address
email
location
class InvManage.views.HistoryForm
    Bases: django.forms.Form
    Form for history view.
qlen
    Count of the events visible on the history view.
        Type int
prefix = history
context
qlen

```

class InvManage.views.CommunicationForm

Bases: django.forms.Form

Form for communication information.

language

Language of communication.

Type str

phone

Contact number of the firm.

Type str

email

E-mail of the contact person.

Type EmailField

fax

Fax number.

Type str

prefix = com

context

language

phone

email

fax

class InvManage.views.PurchaseDataForm

Bases: django.forms.Form

Form for purchase related data.

currency

Currency of purchase.

Type str

minorder

Minimum order quantity.

Type int

contactperson

Name of the contact person.

Type str

refcode

Reference code.

Type str

transportmode

Mode of transport of the shipment.

Type str

prefix = pdform

```

    context
    currency
    minorder
    contactperson
    refcode
    transportmode
class InvManage.views.BankAccountForm
    Bases: django.forms.Form
    Form for bank account details.
    name
        Name of the bank.
        Type str
    branch
        Branch number of the bank.
        Type str
    region
        City in which the branch is located.
        Type str
    route
        Transit number.
        Type str
    number
        Bank account number.
        Type int
    accttype
        Type of bank account.
        Type str
    iban
        IBAN number.
        Type str
    code
        Bank code.
        Type str
    branchcode
        Branch code.
        Type str
    prefix = bank
    context
    name

```

branch

region

route

number

accttype

iban

code

branchcode

class InvManage.views.**Product**
Bases: django.db.models.Model

Model of the product.

name

Name of the product.

Type str

category

Product category.

Type str

item_type

Type of the product.

Type str

description

Short description of the product.

Type str

price

Price of the product.

Type float

quantity

Stock quantity of the product.

Type int

identifier

Unique identifier of the product.

Type str

location

Physical location of the product.

Type str

length

Length of the product.

Type str

width

Width of the product.

Type str

height
Height of the product.

Type str

weight
Weight of the product.

Type str

discount
Default discount percentage on the product.

Type float

barcode
Barcode of the product.

Type str

expiry
Expiry date of the product.

Type DateField

image
Image of the product.

Type ImageField

name

category

item_type

description

price

quantity

identifier

location

length

width

height

weight

discount

barcode

expiry

image

__str__ (*self*)

```
class InvManage.views.ProductPurchaseEntry
    Bases: django.db.models.Model

    Model of the product purchase entry.

    product
        Product associated wit the product purchase entry.

        Type ModelChoiceField

    quantity
        Quantity of the product to be ordered.

        Type int

    price
        Price of the product.

        Type float

    discount
        Percentage discount on the product purchase.

        Type float

    order
        Referenced purchase order.

        Type PurchaseOrder

    receivedQty
        Quantity of product received against the ordered quantity.

        Type str

    acceptedQty
        Quantity of product accepted as OK.

        Type str

    rejectedQty
        Quantity of product rejected (not OK, on HOLD, extra delivery, etc.)

        Type str

    is_complete()
        Returns the completion status (boolean).

    pending_quantity()
        Returns the pending quantity against the entry (int).

    product
    quantity
    price
    discount
    order
    receivedQty
    acceptedQty
    rejectedQty
    is_complete(self)
```

```

    property pending_quantity(self)
class InvManage.views.GRNEntry
    Bases: django.db.models.Model

    Model of the goods receipt note entry (GRNE).

    product
        Primary key of the Product instance associated with the goods receipt note entry.
        Type Product

    quantity
        Ordered quantity of product with reference to product purchase entry.
        Type int

    grne
        Primary key of the GoodsReceiptNote instance.
        Type int

    ppe
        Primary key of the ProductPurchaseEntry instance associated with the GRNE.
        Type int

    remark
        Remarks of the quality engineer or the GRN creator about status of products received.
        Type str

    receivedQty
        Quantity of product received against the ordered quantity.
        Type str

    acceptedQty
        Quantity of product accepted as OK.
        Type str

    rejectedQty
        Quantity of product rejected (not OK, on HOLD, extra delivery, etc.)
        Type str

    product
    quantity
    grn
    ppes
    remark
    receivedQty
    acceptedQty
    rejectedQty
class InvManage.views.ProductSalesEntry
    Bases: django.db.models.Model

    Model of the product sales entry.

```

product

Primary key of the `Product` associated with the product sales entry.

Type *Product*

quantity

Quantity of the product.

Type `int`

price

Price of the product.

Type `float`

discount

Percentage discount on the product.

Type `float`

order

Primary key of the `SalesOrder` referenced by the entry.

Type *SalesOrder*

product**quantity****price****discount****order****class** `InvManage.views.SalesOrder`

Bases: `django.db.models.Model`

Model of the sales order.

consumer

Consumer associated with the sales order.

Type `ModelChoiceField`

date

Date of the sales order creation.

Type `DateField`

so

Sales order number.

Type `int`

discount

Percentage discount associated with the sales order.

Type `float`

tax

Percentage of tax applicable.

Type `float`

paid

Amount received from the consumer.

Type float

balance

Amount balance with the consumer.

Type float

subtotal

Total of all the product sales entries associated with the sales order.

Type float

taxtotal

Total tax applicable on the *subtotal*.

Type float

ordertotal

Total price of the sales order including *taxtotal*.

Type float

consumer

date

so

discount

tax

paid

balance

subtotal

taxtotal

ordertotal

class InvManage.views.**PurchaseOrder**

Bases: django.db.models.Model

Model of the purchase order.

vendor

Vendor associated with the purchase order.

Type *Vendor*

date

Date and time of the purchase order creation.

Type DateField

po

Purchase order number.

Type int

discount

Percentage discount.

Type float

tax

Percentage of tax applicable for the purchase.

Type float

paid

Amount paid against the PO.

Type float

balance

Amount balance which remains to be paid.

Type float

subtotal

Total of all the product purchase entries associated with the purchase order.

Type float

taxtotal

Total tax applicable on the *subtotal*.

Type float

ordertotal

Total price of the purchase order including *taxtotal*.

Type float

is_complete()

Returns the completion status (boolean).

pending_ppes()

Returns list of product purchase entries which are not completed.

vendor**date****po****discount****tax****paid****balance****subtotal****taxtotal****ordertotal****is_complete(*self*)****pending_ppes(*self*)****class** InvManage.views.GoodsReceiptNote

Bases: django.db.models.Model

Model of the goods receipt note (GRN).

vendor

Vendor associated with the goods receipt note.

Type ModelChoiceField

poRef

List of identifiers of the purchase orders from which the goods receipt note is derived.

Type MultipleChoiceField

identifier

Unique identifier of the goods receipt note.

Type str

date

Date of GRN creation.

Type DateField

grnType

Type of GRN (auto or manual).

Type ChoiceField

amendNumber

Amendment number of the GRN.

Type int

amendDate

Amendment date.

Type DateField

transporter

Name of the transport/shipping service.

Type str

vehicleNumber

Vehicle number using which the products are shipped.

Type str

gateInwardNumber

Gate inward number of the vehicle.

Type str

preparedBy

Name/identifier of the person who created the goods receipt note.

Type str

checkedBy

Name/identifier of the person who validated the goods receipt note.

Type str

checkedBy

Name/identifier of the person who inspected the physical products in the goods receipt note.

Type str

approvedBy

Name/identifier of the authority who approved the goods receipt note.

Type str

```
TYPE_CHOICES = [['manual', 'Blank'], ['auto', 'PO Reference']]

vendor
poRef
identifier
date
grnType
amendNumber
amendDate
transporter
vehicleNumber
gateInwardNumber
preparedBy
checkedBy
inspectedBy
approvedBy

class InvManage.views.Consumer
    Bases: django.db.models.Model

    Model of the consumer.

    name
        Name of the consumer.

        Type str

    identifier
        Unique identifier of the consumer.

        Type str

    gstin
        GSTIN number of the consumer.

        Type str

    phone
        Contact number.

        Type str

    address
        Address of the consumer.

        Type str

    email
        E-mail address of the consumer.

        Type str

    location
        City of the consumer.

        Type str
```


Returns [description]

Return type [type]

name

identifier

gstn

phone

address

email

location

__str__(*self*)

class InvManage.views.**Vendor**

Bases: django.db.models.Model

Model of the vendor.

name

Name of the vendor.

Type str

identifier

Unique identifier of the vendor.

Type str

gstn

GSTIN number of the vendor.

Type str

address

Primary key of the ShippingAddress instance associated with the vendor.

Type *ShippingAddress*

communication

Primary key of the Communication instance associated with the vendor.

Type *Communication*

backaccount

Primary key of the BankAccount instance associated with the vendor.

Type *BankAccount*

purchasedata

Primary key of the PurchaseData instance associated with the vendor.

Type *PurchaseData*

Returns String representation of the vendor name.

Return type str

name

identifier

```
    gstin
    address
    communication
    bankaccount
    purchasedata
    __str__(self)
class InvManage.views.Company
    Bases: django.db.models.Model

    Model of the company.

    name
        Name of the company.

        Type str

    owner
        Owner of the company.

        Type str

    gstin
        GSTIN number of the company.

        Type str

    phone
        Contact number.

        Type str

    address
        Postal address.

        Type str

    email
        E-mail address of the contact person.

        Type str

    location
        City of the company.

        Type str

    image
        Photo/logo of the company.

        Type ImageField

    shippingaddress
        Primary key of the ShippingAddress instance associated with the company.

        Type ShippingAddress

    Returns String representation of the company name.

    Return type str

name
```

```

    owner
    gstin
    phone
    address
    email
    location
    image
    shippingaddress
    __str__(self)

class InvManage.views.PurchaseInvoice
    Bases: django.db.models.Model

    Model for the purchase invoice.

    company
        Primary key of the Company.

        Type Company

    po
        Primary key of the PurchaseOrder.

        Type PurchaseOrder

    shippingaddress
        Primary key of the ShippingAddress.

        Type ShippingAddress

    communication
        Primary key of the Communication.

        Type Communication

    company
    po
    shippingaddress
    communication

class InvManage.views.SalesInvoice
    Bases: django.db.models.Model

    Model for the sales order invoice.

    company
        Primary key of the Company.

        Type Company

    so
        Primary key of the SalesOrder.

        Type SalesOrder

    shippingaddress
        Primary key of the ShippingAddress.

```

Type *ShippingAddress*

company

so

shippingaddress

class `InvManage.views.FilterState`

Bases: `django.db.models.Model`

Model of the filter state.

name

Name of the filter state.

Type `str`

name

class `InvManage.views.FilterColumn`

Bases: `django.db.models.Model`

Model of the column state.

name

Name of the column.

Type `str`

label

Label of the column.

Type `str`

visible

Visibility flag.

Type `bool`

position

Index of the column.

Type `int`

state

Primary key of the `FilterState` instance being referenced by the column instance.

Type *FilterState*

name

label

visible

position

state

class `InvManage.views.Object`

Bases: `django.db.models.Model`

Model for a generic object.

identifier

Unique identifier of the object.

```

        Type str
name
    Name of the object.
        Type str
model
    Model associated with the object.
        Type str
identifier
name
model
class InvManage.views.EventCard
    Bases: django.db.models.Model
    Model for event card to be displayed in the history view.
obj
    Object associated with the event.
        Type OneToOneField
objId
    Unique identifier of the object.
        Type str
objname
    Name of the object.
        Type str
objmodel
    Model of the object.
        Type str
date
    Timestamp of the event.
        Type DateField
operation
    Type of event.
        Type str
obj
objId
objname
objmodel
date
operation

```

```
class InvManage.views.EventType
    Bases: django.db.models.Model

    Model for event type.

    name
        Name of the event being triggered.

        Type str

    label
        Label associated with the event.

        Type str

    Returns Returns the string representation of the label when the event type is queried.

    Return type Label

    name
    label
    __str__(self)

class InvManage.views.ObjectModel
    Bases: django.db.models.Model

    Model associated with an object.

    name
        Name of the object.

        Type str

    label
        Label string of the object.

        Type str

    modName
        Name of the model associated with the object.

        Type str

    Returns Returns string representation of the label when object model is queried.

    Return type Label

    name
    label
    modName
    __str__(self)

class InvManage.views.HistoryFilterState
    Bases: django.db.models.Model

    Model of filter state of the history.

    params
        Paramameters of the configured filters.

        Type str
```

params**class** InvManage.views.ShippingAddress

Bases: django.db.models.Model

Model of the shipping address.

title

Title of the firm.

Type str**name**

Name of the firm.

Type str**phone**

Contact number of the firm.

Type str**address**

Postal address.

Type str**city**

City.

Type str**state**

State.

Type str**country**

Country.

Type str**website**

Official website of the firm.

Type URLField**post**

Postal code of the firm.

Type str**title****name****phone****address****city****state****country****website**

post

class InvManage.views.PurchaseData

Bases: django.db.models.Model

Model of the purchase data.

currency

Currency of purchase.

Type str

minorder

Minimum order quantity.

Type int

contactperson

Name of the contact person.

Type str

refcode

Reference code.

Type str

transportmode

Mode of transport of the shipment.

Type str

currency

minorder

contactperson

refcode

transportmode

class InvManage.views.Communication

Bases: django.db.models.Model

Model of the communication information.

language

Language of communication.

Type str

phone

Contact number of the firm.

Type str

email

E-mail of the contact person.

Type EmailField

fax

Fax number.

Type str

language


```
    phone
    email
    fax
class InvManage.views.BankAccount
    Bases: django.db.models.Model
    Model of the bank account details.
    name
        Name of the bank.
        Type str
    branch
        Branch number of the bank.
        Type str
    region
        City in which the branch is located.
        Type str
    route
        Transit number.
        Type str
    number
        Bank account number.
        Type int
    acctype
        Type of bank account.
        Type str
    iban
        IBAN number.
        Type str
    code
        Bank code.
        Type str
    branchcode
        Branch code.
        Type str
    name
    branch
    region
    route
    number
    acctype
```

iban

code

branchcode

`InvManage.views.get_columns(state)`

Gets the columns of the display table.

This function takes a filter state and returns an array of column names that are ordered depending on the position attribute of the columns. This order is important as the queryset will be rendered in the display table according to the column order in this returned array.

Parameters `state` (`FilterState`) – Filter state of the columns.

Returns A list that dictates the order of the columns.

Return type List

`InvManage.views.sort_ascending_descending(request, model)`

Sorts and saves the state of the column.

This function takes a request, finds which column needs to be sorted in ascending/descending order, and returns the sorted queryset.

Parameters

- **request** (`HttpRequest`) – GET request received for retrieving the list of model instances.
- **model** (`Model`) – Model of the objects being retrieved.

Returns List of sorted object instances.

Return type QuerySet

`InvManage.views.change_column_position(request, state)`

Modifies the column states depending on the request.

This function:

- Takes a filter state
- Extracts the column names into an order array
- Modifies the column order in the array depending on user input direction (left/right)
- Saves this modified state of columns into the database
- Returns the modified column names array

Parameters

- **request** (`HttpRequest`) – GET request for retrieving the state of the columns.
- **state** (`FilterState`) – Filter state of the columns.

Returns List of sorted columns.

Return type List

`InvManage.views.paginate(queryset, filter, page_number)`

Paginates the list of objects.

This function takes the entire queryset and filters out only objects belonging to the request page

Parameters

- **queryset** (`QuerySet`) – QuerySet of the objects retrieved from the database.

- **filter** (`FilterState`) – Filter state of the columns.
- **page_number** (`int`) – Page number requested.

```
class InvManage.views.PurchaseOrderFilter
```

```
    Bases: django_filters.FilterSet
```

```
    class Meta
```

```
        model
```

```
        fields
```

```
    context
```

```
    vendor
```

```
    date
```

```
    po
```

```
    ordertotal
```

```
    order_by_field = ordering
```

```
    ordering
```

```
class InvManage.views.ProductSerializer
```

```
    Bases: rest_framework.serializers.ModelSerializer
```

```
    Serializer for Product class instance.
```

The `ProductSerializer.data` attribute gives the JSON serialized data of the Product instance:

```
{
    "pk": 637,
    "name": "Piano",
    "category": "Ultricies PC",
    "quantity": 23921,
    "identifier": "PROD9",
    "location": "Musselburgh",
    "description": "88-key, Tri-sensor Scaled Hammer Action Keyboard II, Simulated_
↪ebony and ivory keys ",
    "prod_id": 637
}
```

```
    class Meta
```

```
        model
```

```
        fields = ['pk', 'name', 'category', 'quantity', 'identifier', 'location',
        'description', 'prod_id']
```

```
    prod_id
```

```
    to_representation(self, instance)
```

```
class InvManage.views.PPEntrySerializer
```

```
    Bases: rest_framework.serializers.ModelSerializer
```

```
    Serializer for ProductPurchaseEntry class instance.
```

The `PPEntrySerializer.data` attribute gives the JSON serialized data of the `ProductPurchaseEntry` instance:

```
{
  "ppe_id": 324,
  "product": {
    "pk": 637,
    "name": "Piano",
    "category": "Ultricies PC",
    "quantity": 23921,
    "identifier": "PROD9",
    "location": "Musselburgh",
    "description": "88-key, Tri-sensor Scaled Hammer Action Keyboard II, ↵
↳ Simulated ebony and ivory keys ",
    "prod_id": 637
  },
  "quantity": 100,
  "price": 10.0,
  "discount": 0.0,
  "order": 182,
  "pendingQty": 50
}
```

class Meta

model

fields = ['ppe_id', 'product', 'quantity', 'price', 'discount', 'order',
'pendingQty']

ppe_id

product

pendingQty

to_representation(self, instance)

create(self, validated_data)

update(self, instance, validated_data)

class `InvManage.views.PurchaseInvoiceSerializer`

Bases: `rest_framework.serializers.ModelSerializer`

Serializer for `PurchaseInvoice` class instance.

The `PurchaseInvoiceSerializer.data` attribute gives the JSON serialized data of the `PurchaseInvoice` instance:

```
{
  "company": {
    "name": "Fringillami",
    "owner": "Ivor Barnett",
    "gstIn": "89AAC056465468",
    "phone": "332 220-7026",
    "address": "Ap #849-6241 Euismod Av., 677598, Carinthia, Belgium",
    "email": "est.tempor@fringillami.org",
  }
}
```

(continues on next page)

(continued from previous page)

```

    "location": "Belgium",
    "image": "/media/images/logo.png"
  },
  "po": {
    "vendor": {
      "name": "Girish",
      "identifier": "GJ",
      "gstn": "GSTIN002",
      "address": {
        "name": "alsf",
        "address": "jas;k",
        "city": ";sdalkf",
        "phone": "alsf",
        "state": "kjdfk",
        "country": "ljflkj",
        "post": "54545"
      }
    },
    "date": "25-Sep-2021",
    "po": 293,
    "subtotal": 279975.0,
    "taxtotal": 22398.0,
    "ordertotal": 302373.0,
    "ppes": [
      {
        "ppe_id": 324,
        "product": {
          "pk": 637,
          "name": "piano",
          "category": "Ultricies PC",
          "quantity": 23921,
          "identifier": "PROD9",
          "location": "Musselburgh",
          "description": "88-key, Tri-sensor Scaled Hammer Action",
          "prod_id": 637
        },
        "quantity": 100,
        "price": 10.0,
        "discount": 0.0,
        "order": 182,
        "pendingQty": 50
      },
      {
        "ppe_id": 325,
        "product": {
          "pk": 645,
          "name": "Tabl",
          "category": "Sociis Natoque Company",
          "quantity": 38276,
          "identifier": "PROD17",
          "location": "Schagen",

```

(continues on next page)

(continued from previous page)

```

        "description": "aldgjlakjlkasdj",
        "prod_id": 645
    },
    "quantity": 250,
    "price": 90.0,
    "discount": 8.0,
    "order": 182,
    "pendingQty": 70
},
{
    "ppe_id": 326,
    "product": {
        "pk": 638,
        "name": "Goblet drum",
        "category": "Est Congue Consulting",
        "quantity": 46076,
        "identifier": "PROD10",
        "location": "Kaluga",
        "description": "aldgjlakjlkasdj",
        "prod_id": 638
    },
    "quantity": 200,
    "price": 150.0,
    "discount": 12.0,
    "order": 182,
    "pendingQty": 200
},
{
    "ppe_id": 327,
    "product": {
        "pk": 643,
        "name": "quinto",
        "category": "Enim Suspendisse Associates",
        "quantity": 51099,
        "identifier": "PROD15",
        "location": "Nagpur",
        "description": "aldgjlakjlkasdj",
        "prod_id": 643
    },
    "quantity": 350,
    "price": 150.0,
    "discount": 10.0,
    "order": 182,
    "pendingQty": 350
},
{
    "ppe_id": 328,
    "product": {
        "pk": 651,
        "name": "Igihumurizo",
        "category": "Curabitur Massa Vestibulum Consulting",
        "quantity": 48553,

```

(continues on next page)

(continued from previous page)

```

        "identifier": "PROD23",
        "location": "Yeovil",
        "description": "aldgjlakjlkasdj",
        "prod_id": 651
    },
    "quantity": 500,
    "price": 50.0,
    "discount": 6.0,
    "order": 182,
    "pendingQty": 500
},
{
    "ppe_id": 329,
    "product": {
        "pk": 653,
        "name": "Balsié",
        "category": "Eu Foundation",
        "quantity": 29988,
        "identifier": "PROD25",
        "location": "Sudbury",
        "description": "aldgjlakjlkasdj",
        "prod_id": 653
    },
    "quantity": 200,
    "price": 500.0,
    "discount": 15.0,
    "order": 182,
    "pendingQty": 200
},
{
    "ppe_id": 330,
    "product": {
        "pk": 656,
        "name": "Padayani thappu",
        "category": "Posuere LLP",
        "quantity": 6358,
        "identifier": "PROD28",
        "location": "Ruvo del Monte",
        "description": "aldgjlakjlkasdj",
        "prod_id": 656
    },
    "quantity": 350,
    "price": 250.0,
    "discount": 13.0,
    "order": 182,
    "pendingQty": 350
}
]
},
"shippingaddress": {
    "name": "Harding Gross",
    "address": "8798 At, St., 7639",

```

(continues on next page)

(continued from previous page)

```

        "city": "Rome",
        "phone": "936 651-4847",
        "state": "Lazio",
        "country": "Italy",
        "post": "300326"
    },
    "communication": {
        "email": "asfs@aflk.com",
        "phone": "6546432"
    }
}

```

class Meta

model

fields = ['company', 'po', 'shippingaddress', 'communication']

company

po

shippingaddress

communication

`InvManage.views.create_purchase_order_view(request)`

Creates a purchase order (PO) on POST request, and returns a PO creation form on GET request.

GET /purchase_order

Gets the purchase order creation form.

Example request:

```

GET /purchase_order/ HTTP/1.1
Host: localhost:8000
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/
webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9

```

Example response:

```

HTTP/1.1 200 OK
Vary: Accept
Content-Type: text/html; charset=utf-8

```

Request Headers

- **Accept** – text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9

Status Codes

- **200 OK** – Purchase order creation form received successfully.

POST /purchase_order

Creates a purchase order.

Example request:


```
POST /purchase_order/ HTTP/1.1
Host: localhost:8000
Content-Type: multipart/form-data;
```

Form Parameters

- po-vendor – 4
- vend-name – Girish
- vend-identifier – GJ
- po-po – 69
- po-date – 2021-09-29
- po-tax – 12
- po-discount – 8
- po-paid – 4500
- po-balance – 1200
- ship-title – FingDocks
- ship-name – Harding Gross
- ship-phone – 936 651-4847
- ship-address – 8798 At, St., 7639
- ship-city – Rome
- ship-state – Lazio
- ship-country – Italy
- ship-website – http://fringdocs.com
- ship-post – 300326
- po-subtotal – 2595000.00
- po-taxtotal – 207600.00
- po-ordertotal – 2802600.00
- form-TOTAL_FORMS – 2
- form-INITIAL_FORMS – 0
- form-MIN_NUM_FORMS –
- form-MAX_NUM_FORMS –
- form-0-product – 649
- form-0-quantity – 300
- form-0-price – 4500
- form-0-discount – 10
- form-0-DELETE –
- form-0-ppe_id – -1
- form-1-product – 664

- **form-1-quantity** – 250
- **form-1-price** – 6000
- **form-1-discount** – 8
- **form-1-DELETE** –
- **form-1-ppe_id** – -1

Response Headers

- **Content-Type** – multipart/form-data; boundary=—WebKitFormBoundaryLTR88aZAnBUSE7mv

Status Codes

- **302 Found** – Redirects to `/purchase_order`.

`InvManage.views.update_purchase_order_view(request)`

Updates a purchase order on POST request and returns the purchase order update form for GET request.

GET `/purchaes_order/update`

Gets the purchase order update form whose primary key matches the query parameter `pk`.

Example request:

```
GET /purchase_order/update HTTP/1.1
Host: localhost:8000
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
```

Query Parameters

- **pk** – The primary key of the purchase order.

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: text/html; charset=utf-8
```

Request Headers

- **Accept** – text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9

Status Codes

- **200 OK** – Purchase order update form received successfully.

POST `/purchase_order/update`

Updates the purchase order.

Example request:

```
POST /purchase_order/update HTTP/1.1
Host: localhost:8000
Content-Type: multipart/form-data;
```

Form Parameters

- **pk** – 187
- **po-vendor** – 4
- **vend-name** – Girish
- **vend-identifier** – GJ
- **po-po** – 69
- **po-date** – 2021-09-29
- **po-tax** – 12
- **po-discount** – 8.0
- **po-paid** – 4500.0
- **po-balance** – 1200.0
- **ship-title** – FingDocks
- **ship-name** – Harding Gross
- **ship-phone** – 936 651-4847
- **ship-address** – 8798 At, St., 7639
- **ship-city** – Rome
- **ship-state** – Lazio
- **ship-country** – Italy
- **ship-website** – <http://fringdocs.com>
- **ship-post** – 300326
- **po-subtotal** – 2595000.0
- **po-taxtotal** – 207600.0
- **po-ordertotal** – 2802600.0
- **form-TOTAL_FORMS** – 2
- **form-INITIAL_FORMS** – 2
- **form-MIN_NUM_FORMS** –
- **form-MAX_NUM_FORMS** –
- **form-0-product** – 649
- **form-0-quantity** – 300
- **form-0-price** – 4500
- **form-0-discount** – 10
- **form-0-DELETE** –
- **form-0-ppe_id** – 339
- **form-1-product** – 664
- **form-1-quantity** – 250
- **form-1-price** – 6000
- **form-1-discount** – 8

- **form-1-DELETE** –
- **form-1-ppe_id** – 340

Response Headers

- **Content-Type** – multipart/form-data; boundary=——WebKitFormBoundaryLTR88aZAnBUSE7mv

Status Codes

- **302 Found** – Redirects to /consumer.

`InvManage.views.display_purchase_orders_view(request)`

Retrieves the list of purchase orders on GET request.

GET /purchase_orders/

Gets the list of all purchase orders.

Example request:

```
GET /purchase_orders/ HTTP/1.1
Host: localhost:8000
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
```

Form Parameters

- **page** – The page number of the purchase order list.

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: text/html; charset=utf-8
```

Request Headers

- **Accept** – text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9

Status Codes

- **200 OK** – List of purchase orders received successfully.

`InvManage.views.delete_purchase_order_view(request, pk)`

Deletes the purchase order with primary key `object_id` on POST request.

POST /purchase_order/<str:object_id>/delete

Deletes the consumer represented by the primary key `object_id`.

Example request:

```
POST /purchase_order/187/delete HTTP/1.1
Host: localhost:8000
Content-Type: application/x-www-form-urlencoded
```

Parameters

- **object_id** – Purchase order primary key.

Response Headers

- **Content-Type** – application/x-www-form-urlencoded

Status Codes

- **302 Found** – Redirects to /purchase_order.
- **500 Internal Server Error** – Purchase order matching query does not exist.

InvManage.views.**print_purchase_order_view**(request, pk)

Opens the purchase order with primary key po_id for printing on GET request.

POST /purchase_order/<str:po_id>/print

Opens the purchase order represented by the primary key po_id.

Example request:

```
POST /purchase_order/182/print HTTP/1.1
Host: localhost:8000
Content-Type: application/x-www-form-urlencoded
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json; charset=utf-8

[
  {
    "company": {
      "name": "Fringillami",
      "owner": "Ivor Barnett",
      "gstn": "89AAC056465468",
      "phone": "332 220-7026",
      "address": "Ap #849-6241 Euismod Av., 677598, Carinthia, Belgium",
      "email": "est.tempor@fringillami.org",
      "location": "Belgium",
      "image": "/media/images/hyperlink-green_x91WW5n.png"
    },
    "po": {
      "vendor": {
        "name": "Girish",
        "identifier": "GJ",
        "gstn": "GSTIN002",
        "address": {
          "name": "alsf",
          "address": "jas;k",
          "city": ";sdalkf",
          "phone": "alsf",
          "state": "kjdfk",
          "country": "ljflkj",
          "post": "54545"
        }
      },
      "date": "25-Sep-2021",
      "po": 293,
```

(continues on next page)

(continued from previous page)

```

    "subtotal": 279975.0,
    "taxtotal": 22398.0,
    "ordertotal": 302373.0,
    "ppes": [
      {
        "ppe_id": 324,
        "product": {
          "pk": 637,
          "name": "piano",
          "category": "Ultricies PC",
          "quantity": 23921,
          "identifier": "PROD9",
          "location": "Musselburgh",
          "description": "88-key, Tri-sensor Scaled Hammer Action_
↳Keyboard II, Simulated ebony and ivory keys ",
          "prod_id": 637
        },
        "quantity": 100,
        "price": 10.0,
        "discount": 0.0,
        "order": 182,
        "pendingQty": 50
      },
      {
        "ppe_id": 325,
        "product": {
          "pk": 645,
          "name": "Tabl",
          "category": "Sociis Natoque Company",
          "quantity": 38276,
          "identifier": "PROD17",
          "location": "Schagen",
          "description": "aldgjllakjlkasdj",
          "prod_id": 645
        },
        "quantity": 250,
        "price": 90.0,
        "discount": 8.0,
        "order": 182,
        "pendingQty": 70
      },
      {
        "ppe_id": 326,
        "product": {
          "pk": 638,
          "name": "Goblet drum",
          "category": "Est Congue Consulting",
          "quantity": 46076,
          "identifier": "PROD10",
          "location": "Kaluga",
          "description": "aldgjllakjlkasdj",
          "prod_id": 638
        }
      }
    ]
  }
}

```

(continues on next page)

(continued from previous page)

```

    },
    "quantity": 200,
    "price": 150.0,
    "discount": 12.0,
    "order": 182,
    "pendingQty": 200
  },
  {
    "ppe_id": 327,
    "product": {
      "pk": 643,
      "name": "quinto",
      "category": "Enim Suspendisse Associates",
      "quantity": 51099,
      "identifier": "PROD15",
      "location": "Nagpur",
      "description": "aldgjklakjlkasdj",
      "prod_id": 643
    },
    "quantity": 350,
    "price": 150.0,
    "discount": 10.0,
    "order": 182,
    "pendingQty": 350
  },
  {
    "ppe_id": 328,
    "product": {
      "pk": 651,
      "name": "Igihumurizo",
      "category": "Curabitur Massa Vestibulum Consulting",
      "quantity": 48553,
      "identifier": "PROD23",
      "location": "Yeovil",
      "description": "aldgjklakjlkasdj",
      "prod_id": 651
    },
    "quantity": 500,
    "price": 50.0,
    "discount": 6.0,
    "order": 182,
    "pendingQty": 500
  },
  {
    "ppe_id": 329,
    "product": {
      "pk": 653,
      "name": "Balsié",
      "category": "Eu Foundation",
      "quantity": 29988,
      "identifier": "PROD25",
      "location": "Sudbury",

```

(continues on next page)

(continued from previous page)

```

        "description": "aldgjlakjlkasdj",
        "prod_id": 653
    },
    "quantity": 200,
    "price": 500.0,
    "discount": 15.0,
    "order": 182,
    "pendingQty": 200
},
{
    "ppe_id": 330,
    "product": {
        "pk": 656,
        "name": "Padayani thappu",
        "category": "Posuere LLP",
        "quantity": 6358,
        "identifier": "PROD28",
        "location": "Ruvo del Monte",
        "description": "aldgjlakjlkasdj",
        "prod_id": 656
    },
    "quantity": 350,
    "price": 250.0,
    "discount": 13.0,
    "order": 182,
    "pendingQty": 350
}
]
},
"shippingaddress": {
    "name": "Harding Gross",
    "address": "8798 At, St., 7639",
    "city": "Rome",
    "phone": "936 651-4847",
    "state": "Lazio",
    "country": "Italy",
    "post": "300326"
},
"communication": {
    "email": "asfs@aflk.com",
    "phone": "6546432"
}
}
]

```

Parameters

- **po_id** – Purchase order primary key.

Response Headers

- **Content-Type** – application/json

Status Codes

- 200 OK – Purchase order print request successful.
- 500 Internal Server Error – Purchase order matching query does not exist.

`InvManage.views.get_product_purchase_entries_view(request)`

Returns the JSON serialized data of product purchase entries on GET request.

GET /product_purchase_entries/

Gets the JSON serialized data of the requested product purchase entries.

Example request:

```
GET /product_purchase_entries/ HTTP/1.1
Host: localhost:8000
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json; charset=utf-8

[
  {
    "ppe_id": 324,
    "product": {
      "pk": 637,
      "name": "piano",
      "category": "Ultricies PC",
      "quantity": 23921,
      "identifier": "PROD9",
      "location": "Musselburgh",
      "description": "88-key, Tri-sensor Scaled Hammer Action Keyboard II,
      ↳ Simulated ebony and ivory keys ",
      "prod_id": 637
    },
    "quantity": 100,
    "price": 10.0,
    "discount": 0.0,
    "order": 182,
    "pendingQty": 50,
    "po": 293
  },
  {
    "ppe_id": 325,
    "product": {
      "pk": 645,
      "name": "Tabl",
      "category": "Sociis Natoque Company",
      "quantity": 38276,
      "identifier": "PROD17",
      "location": "Schagen",
      "description": "aldgjlakjlkasdj",
      "prod_id": 645
    }
  }
]
```

(continues on next page)

(continued from previous page)

```

    },
    "quantity": 250,
    "price": 90.0,
    "discount": 8.0,
    "order": 182,
    "pendingQty": 70,
    "po": 293
  },
  {
    "ppe_id": 326,
    "product": {
      "pk": 638,
      "name": "Goblet drum",
      "category": "Est Congue Consulting",
      "quantity": 46076,
      "identifier": "PROD10",
      "location": "Kaluga",
      "description": "aldgjlakjlkasdj",
      "prod_id": 638
    },
    "quantity": 200,
    "price": 150.0,
    "discount": 12.0,
    "order": 182,
    "pendingQty": 200,
    "po": 293
  },
  {
    "ppe_id": 327,
    "product": {
      "pk": 643,
      "name": "quinto",
      "category": "Enim Suspendisse Associates",
      "quantity": 51099,
      "identifier": "PROD15",
      "location": "Nagpur",
      "description": "aldgjlakjlkasdj",
      "prod_id": 643
    },
    "quantity": 350,
    "price": 150.0,
    "discount": 10.0,
    "order": 182,
    "pendingQty": 350,
    "po": 293
  },
  {
    "ppe_id": 328,
    "product": {
      "pk": 651,
      "name": "Igihumurizo",
      "category": "Curabitur Massa Vestibulum Consulting",

```

(continues on next page)

(continued from previous page)

```

        "quantity": 48553,
        "identifier": "PROD23",
        "location": "Yeovil",
        "description": "aldgjlakjlkasdj",
        "prod_id": 651
    },
    "quantity": 500,
    "price": 50.0,
    "discount": 6.0,
    "order": 182,
    "pendingQty": 500,
    "po": 293
},
{
    "ppe_id": 329,
    "product": {
        "pk": 653,
        "name": "Balsié",
        "category": "Eu Foundation",
        "quantity": 29988,
        "identifier": "PROD25",
        "location": "Sudbury",
        "description": "aldgjlakjlkasdj",
        "prod_id": 653
    },
    "quantity": 200,
    "price": 500.0,
    "discount": 15.0,
    "order": 182,
    "pendingQty": 200,
    "po": 293
},
{
    "ppe_id": 330,
    "product": {
        "pk": 656,
        "name": "Padayani thappu",
        "category": "Posuere LLP",
        "quantity": 6358,
        "identifier": "PROD28",
        "location": "Ruvo del Monte",
        "description": "aldgjlakjlkasdj",
        "prod_id": 656
    },
    "quantity": 350,
    "price": 250.0,
    "discount": 13.0,
    "order": 182,
    "pendingQty": 350,
    "po": 293
}
}
]

```

Request Headers

- **Accept** – text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9

Status Codes

- **200 OK** – List of product purchase entries received successfully.
- **400 Bad Request** – Bad request version
- **500 Internal Server Error** – Product purchase entries matching query does not exist.

```
class InvManage.views.ProductBasicInfoForm
```

```
    Bases: django.forms.Form
```

```
    Form for basic product information.
```

```
    name
```

```
        Name of the product.
```

```
        Type str
```

```
    item_type
```

```
        Type of the product.
```

```
        Type str
```

```
    category
```

```
        Product category.
```

```
        Type str
```

```
    description
```

```
        Short description of the product.
```

```
        Type str
```

```
    prefix = basic
```

```
    context
```

```
    name
```

```
    item_type
```

```
    category
```

```
    description
```

```
class InvManage.views.ProductDetailedInfoForm
```

```
    Bases: django.forms.Form
```

```
    Form for product physical details
```

```
    length
```

```
        Length of the product.
```

```
        Type str
```

```
    width
```

```
        Width of the product.
```

```
        Type str
```

```
    height
```

```
        Height of the product.
```

```

        Type str
    weight
        Weight of the product.
        Type str
    prefix = detailed
    context
    length
    width
    height
    weight
class InvManage.views.ProductPricingForm
    Bases: django.forms.Form
    Form for pricing information of the product.
    price
        Price of the product.
        Type float
    discount
        Default discount percentage on the product.
        Type float
    prefix = pricing
    context
    price
    discount
class InvManage.views.ProductPurchaseEntryForm
    Bases: django.forms.Form
    Form for product purchase entry.
    ppe_id
        Unique identifier of the product purchase entry.
        Type int
    product
        Product associated wit the product purchase entry.
        Type Product
    quantity
        Quantity of the product to be ordered.
        Type int
    price
        Price of the product.
        Type float

```

discount

Percentage discount on the product purchase.

Type float

prefix = form

context

ppe_id

product

quantity

price

discount

class InvManage.views.ProductSalesEntryForm

Bases: django.forms.Form

Form for product sales entry.

pse_id

Unique identifier of the product sales entry.

Type int

product

Product associated with the product sales entry.

Type *Product*

quantity

Quantity of the product.

Type int

price

Price of the product.

Type float

discount

Percentage discount on the product.

Type float

prefix = form

context

pse_id

product

quantity

price

discount

class InvManage.views.ProductStatusForm

Bases: django.forms.Form

Form for locating and identifying the product.

quantity

Stock quantity of the product.

Type int

identifier

Unique identifier of the product.

Type str

location

Physical location of the product.

Type str

context

prefix = status

quantity

identifier

location

class InvManage.views.ProductStorageInfoForm

Bases: django.forms.Form

Form for storage information.

barcode

Barcode of the product.

Type str

expiry

Expiry date of the product.

Type DateField

prefix = storage

barcode

context

expiry

class InvManage.views.PurchaseOrderBasicInfo

Bases: django.forms.Form

Form for basic information of the purchase order.

vendor

Vendor associated with the purchase order.

Type *Vendor*

date

Date of the purchase order creation.

Type DateField

po

Purchase order number.

Type int

discount

Percentage of overall discount.

Type float

tax

Percentage of tax.

Type float

paid

Amount paid to the vendor.

Type float

balance

Balance amount to be paid to the vendor.

Type float

subtotal

Total of all the product purchase entries associated with the purchase order.

Type float

taxtotal

Total tax applicable on the *subtotal*.

Type float

ordertotal

Total price of the purchase order including *taxtotal*.

Type float

prefix = po**context****product_choices****vendor_choices****vendor****date****po****discount****tax****paid****balance****subtotal****taxtotal****ordertotal****class** InvManage.views.GRNInfo

Bases: django.forms.Form

Form for goods receipt note (GRN).

vendor

Vendor associated with the goods receipt note.

Type *Vendor*

poRef

List of identifiers of the purchase orders from which the goods receipt note is derived.

Type *PurchaseOrder*

identifier

Unique identifier of the goods receipt note.

Type str

date

Date of GRN creation.

Type DateField

grnType

Type of GRN (auto or manual).

Type str

amendNumber

Amendment number of the GRN.

Type int

amendDate

Amendment date.

Type DateField

transporter

Name of the transport/shipping service.

Type str

vehicleNumber

Vehicle number using which the products are shipped.

Type str

gateInwardNumber

Gate inward number of the vehicle.

Type str

preparedBy

Name/identifier of the person who created the goods receipt note.

Type str

checkedBy

Name/identifier of the person who validated the goods receipt note.

Type str

checkedBy

Name/identifier of the person who inspected the physical products in the goods receipt note.

Type str

approvedBy

Name/identifier of the authority who approved the goods receipt note.

```
        Type str
    prefix = grn
    context
    product_choices
    vendor_choices
    vendor
    TYPE_CHOICES = [['manual', 'Blank'], ['auto', 'PO Reference']]
    poRef
    identifier
    date
    grnType
    amendNumber
    amendDate
    transporter
    vehicleNumber
    gateInwardNumber
    preparedBy
    checkedBy
    inspectedBy
    approvedBy
class InvManage.views.GRNEnterForm
    Bases: django.forms.Form
    Form for goods receipt note entry (GRNE).
    product
        Product associated with the goods receipt note entry.
        Type Product
    grne_id
        Unique identifier of the GRNE.
        Type int
    ppe_id
        Unique identifier of the product purchase entry associated with the GRNE.
        Type int
    quantity
        Ordered quantity of product with reference to product purchase entry.
        Type int
    remark
        Remarks of the quality engineer or the GRN creator about status of products received.
        Type str
```

receivedQty

Quantity of product received against the ordered quantity.

Type str

acceptedQty

Quantity of product accepted as OK.

Type str

rejectedQty

Quantity of product rejected (not OK, on HOLD, extra delivery, etc.)

Type str

prefix = form

context

product

grne_id

ppe_id

quantity

remark

receivedQty

acceptedQty

rejectedQty

class InvManage.views.**ThumbnailForm**

Bases: django.forms.Form

For for product thumbnail image.

image

Image of the product.

Type ImageField

prefix = thumbnail

image

class InvManage.views.**SalesOrderBasicInfo**

Bases: django.forms.Form

Form for basic information on sales order.

consumer

Consumer associated with the sales order.

Type *Consumer*

date

Date of the sales order creation.

Type DateField

so

Sales order number.

Type int

discount

Percentage discount associated with the sales order.

Type float

tax

Percentage of tax applicable.

Type float

paid

Amount received from the consumer.

Type float

balance

Amount balance with the consumer.

Type float

subtotal

Total of all the product sales entries associated with the sales order.

Type float

taxtotal

Total tax applicable on the *subtotal*.

Type float

ordertotal

Total price of the sales order including *taxtotal*.

Type float

prefix = so

context

product_choices

consumer_choices

consumer

date

so

discount

tax

paid

balance

subtotal

taxtotal

ordertotal

class InvManage.views.ShippingAddressForm

Bases: django.forms.Form

For for shipping address.

title
Title of the firm.
Type str

name
Name of the firm.
Type str

phone
Contact number of the firm.
Type str

address
Postal address.
Type str

city
City.
Type str

state
State.
Type str

country
Country.
Type str

website
Official website of the firm.
Type URLField

post
Postal code of the firm.
Type str

prefix = ship

context

title

name

phone

address

city

state

country

website

post

```
class InvManage.views.VendorForm
    Bases: django.forms.Form

    Form for basic informaion of the vendor.

    name
        Name of the vendor.

        Type str

    identifier
        Unique identifier of the vendor.

        Type str

    gstin
        GSTIN number of the vendor.

        Type str

    prefix = vend

    context

    name

    identifier

    gstin

class InvManage.views.ConsumerForm
    Bases: django.forms.Form

    Form for consumer.

    name
        Name of the consumer.

        Type str

    identifier
        Unique identifier of the consumer.

        Type str

    gstin
        GSTIN number of the consumer.

        Type str

    phone
        Contact number.

        Type str

    address
        Address of the consumer.

        Type str

    email
        E-mail address of the consumer.

        Type str

    location
        City of the consumer.
```

```

        Type str
    prefix = consumer
    context
    name
    identifier
    gstin
    phone
    address
    email
    location
class InvManage.views.CompanyForm
    Bases: django.forms.Form
    Form for company.
    name
        Name of the company.
        Type str
    owner
        Name of the owner of the company.
        Type str
    gstin
        GSTIN number of the company.
        Type str
    phone
        Contact number of the company.
        Type str
    address
        Postal address.
        Type str
    email
        E-mail address of the contact person.
        Type str
    location
        City of the company.
        Type str
    prefix = comp
    context
    name
    owner
    gstin

```

phone

address

email

location

class InvManage.views.**HistoryForm**

Bases: django.forms.Form

Form for history view.

qlen

Count of the events visible on the history view.

Type int

prefix = history

context

qlen

class InvManage.views.**CommunicationForm**

Bases: django.forms.Form

Form for communication information.

language

Language of communication.

Type str

phone

Contact number of the firm.

Type str

email

E-mail of the contact person.

Type EmailField

fax

Fax number.

Type str

prefix = com

context

language

phone

email

fax

class InvManage.views.**PurchaseDataForm**

Bases: django.forms.Form

Form for purchase related data.

currency

Currency of purchase.


```

        Type str
minorder
    Minimum order quantity.
        Type int
contactperson
    Name of the contact person.
        Type str
refcode
    Reference code.
        Type str
transportmode
    Mode of transport of the shipment.
        Type str
prefix = pdform
context
currency
minorder
contactperson
refcode
transportmode
class InvManage.views.BankAccountForm
    Bases: django.forms.Form
    Form for bank account details.
name
    Name of the bank.
        Type str
branch
    Branch number of the bank.
        Type str
region
    City in which the branch is located.
        Type str
route
    Transit number.
        Type str
number
    Bank account number.
        Type int
acctype
    Type of bank account.

```

```
        Type str
iban
    IBAN number.
        Type str
code
    Bank code.
        Type str
branchcode
    Branch code.
        Type str
prefix = bank
context
name
branch
region
route
number
acctype
iban
code
branchcode
class InvManage.views.Product
    Bases: django.db.models.Model
    Model of the product.
    name
        Name of the product.
        Type str
    category
        Product category.
        Type str
    item_type
        Type of the product.
        Type str
    description
        Short description of the product.
        Type str
    price
        Price of the product.
        Type float
```

quantity
Stock quantity of the product.
Type int

identifier
Unique identifier of the product.
Type str

location
Physical location of the product.
Type str

length
Length of the product.
Type str

width
Width of the product.
Type str

height
Height of the product.
Type str

weight
Weight of the product.
Type str

discount
Default discount percentage on the product.
Type float

barcode
Barcode of the product.
Type str

expiry
Expiry date of the product.
Type DateField

image
Image of the product.
Type ImageField

name

category

item_type

description

price

quantity

identifier

location
length
width
height
weight
discount
barcode
expiry
image
__str__(*self*)

class InvManage.views.**ProductPurchaseEntry**

Bases: `django.db.models.Model`

Model of the product purchase entry.

product

Product associated wit the product purchase entry.

Type `ModelChoiceField`

quantity

Quantity of the product to be ordered.

Type `int`

price

Price of the product.

Type `float`

discount

Percentage discount on the product purchase.

Type `float`

order

Referenced purchase order.

Type *PurchaseOrder*

receivedQty

Quantity of product received against the ordered quantity.

Type `str`

acceptedQty

Quantity of product accepted as OK.

Type `str`

rejectedQty

Quantity of product rejected (not OK, on HOLD, extra delivery, etc.)

Type `str`

is_complete()

Returns the completion status (boolean).

```

pending_quantity()
    Returns the pending quantity against the entry (int).

product

quantity

price

discount

order

receivedQty

acceptedQty

rejectedQty

is_complete(self)

property pending_quantity(self)

class InvManage.views.GRNEntry
    Bases: django.db.models.Model

    Model of the goods receipt note entry (GRNE).

    product
        Primary key of the Product instance associated with the goods receipt note entry.
        Type Product

    quantity
        Ordered quantity of product with reference to product purchase entry.
        Type int

    grne
        Primary key of the GoodsReceiptNote instance.
        Type int

    ppe
        Primary key of the ProductPurchaseEntry instance associated with the GRNE.
        Type int

    remark
        Remarks of the quality engineer or the GRN creator about status of products received.
        Type str

    receivedQty
        Quantity of product received against the ordered quantity.
        Type str

    acceptedQty
        Quantity of product accepted as OK.
        Type str

    rejectedQty
        Quantity of product rejected (not OK, on HOLD, extra delivery, etc.)
        Type str

```

product
quantity
grn
ppes
remark
receivedQty
acceptedQty
rejectedQty

class InvManage.views.**ProductSalesEntry**

Bases: django.db.models.Model

Model of the product sales entry.

product
Primary key of the Product associated with the product sales entry.

Type *Product*

quantity
Quantity of the product.

Type int

price
Price of the product.

Type float

discount
Percentage discount on the product.

Type float

order
Primary key of the SalesOrder referenced by the entry.

Type *SalesOrder*

product

quantity

price

discount

order

class InvManage.views.**SalesOrder**

Bases: django.db.models.Model

Model of the sales order.

consumer
Consumer associated with the sales order.

Type ModelChoiceField

date
Date of the sales order creation.

```

        Type DateField

so
    Sales order number.

        Type int

discount
    Percentage discount associated with the sales order.

        Type float

tax
    Percentage of tax applicable.

        Type float

paid
    Amount received from the consumer.

        Type float

balance
    Amount balance with the consumer.

        Type float

subtotal
    Total of all the product sales entries associated with the sales order.

        Type float

taxtotal
    Total tax applicable on the subtotal.

        Type float

ordertotal
    Total price of the sales order including taxtotal.

        Type float

consumer
date
so
discount
tax
paid
balance
subtotal
taxtotal
ordertotal

class InvManage.views.PurchaseOrder
    Bases: django.db.models.Model
    Model of the purchase order.

```

vendor

Vendor associated with the purchase order.

Type *Vendor*

date

Date and time of the purchase order creation.

Type *DateField*

po

Purchase order number.

Type *int*

discount

Percentage discount.

Type *float*

tax

Percentage of tax applicable for the purchase.

Type *float*

paid

Amount paid against the PO.

Type *float*

balance

Amount balance which remains to be paid.

Type *float*

subtotal

Total of all the product purchase entries associated with the purchase order.

Type *float*

taxtotal

Total tax applicable on the *subtotal*.

Type *float*

ordertotal

Total price of the purchase order including *taxtotal*.

Type *float*

is_complete()

Returns the completion status (boolean).

pending_ppes()

Returns list of product purchase entries which are not completed.

vendor**date****po****discount****tax****paid**


```

    balance
    subtotal
    taxtotal
    ordertotal
    is_complete(self)
    pending_ppes(self)
class InvManage.views.GoodsReceiptNote
    Bases: django.db.models.Model

    Model of the goods receipt note (GRN).

    vendor
        Vendor associated with the goods receipt note.
        Type ModelChoiceField

    poRef
        List of identifiers of the purchase orders from which the goods receipt note is derived.
        Type MultipleChoiceField

    identifier
        Unique identifier of the goods receipt note.
        Type str

    date
        Date of GRN creation.
        Type DateField

    grnType
        Type of GRN (auto or manual).
        Type ChoiceField

    amendNumber
        Amendment number of the GRN.
        Type int

    amendDate
        Amendment date.
        Type DateField

    transporter
        Name of the transport/shipping service.
        Type str

    vehicleNumber
        Vehicle number using which the products are shipped.
        Type str

    gateInwardNumber
        Gate inward number of the vehicle.
        Type str

```

preparedBy

Name/identifier of the person who created the goods receipt note.

Type str

checkedBy

Name/identifier of the person who validated the goods receipt note.

Type str

checkedBy

Name/identifier of the person who inspected the physical products in the goods receipt note.

Type str

approvedBy

Name/identifier of the authority who approved the goods receipt note.

Type str

TYPE_CHOICES = [['manual', 'Blank'], ['auto', 'PO Reference']]

vendor

poRef

identifier

date

grnType

amendNumber

amendDate

transporter

vehicleNumber

gateInwardNumber

preparedBy

checkedBy

inspectedBy

approvedBy

class InvManage.views.Consumer

Bases: django.db.models.Model

Model of the consumer.

name

Name of the consumer.

Type str

identifier

Unique identifier of the consumer.

Type str

gstIn

GSTIN number of the consumer.

Type str

phone

Contact number.

Type str**address**

Address of the consumer.

Type str**email**

E-mail address of the consumer.

Type str**location**

City of the consumer.

Type str**Returns** [description]**Return type** [type]**name****identifier****gstin****phone****address****email****location****__str__**(*self*)**class** InvManage.views.**Vendor**

Bases: django.db.models.Model

Model of the vendor.

name

Name of the vendor.

Type str**identifier**

Unique identifier of the vendor.

Type str**gstin**

GSTIN number of the vendor.

Type str**address**

Primary key of the ShippingAddress instance associated with the vendor.

Type *ShippingAddress***communication**

Primary key of the Communication instance associated with the vendor.

Type *Communication*

backaccount

Primary key of the BankAccount instance associated with the vendor.

Type *BankAccount*

purchasedata

Primary key of the PurchaseData instance associated with the vendor.

Type *PurchaseData*

Returns String representation of the vendor name.

Return type str

name

identifier

gstn

address

communication

bankaccount

purchasedata

__str__(self)

class InvManage.views.Company

Bases: django.db.models.Model

Model of the company.

name

Name of the company.

Type str

owner

Owner of the company.

Type str

gstn

GSTIN number of the company.

Type str

phone

Contact number.

Type str

address

Postal address.

Type str

email

E-mail address of the contact person.

Type str

location

City of the company.

Type str**image**

Photo/logo of the company.

Type ImageField**shippingaddress**

Primary key of the ShippingAddress instance associated with the company.

Type *ShippingAddress***Returns** String representation of the company name.**Return type** str**name****owner****gsttin****phone****address****email****location****image****shippingaddress****__str__**(*self*)**class** InvManage.views.**PurchaseInvoice**

Bases: django.db.models.Model

Model for the purchase invoice.

company

Primary key of the Company.

Type *Company***po**

Primary key of the PurchaseOrder.

Type *PurchaseOrder***shippingaddress**

Primary key of the ShippingAddress.

Type *ShippingAddress***communication**

Primary key of the Communication.

Type *Communication***company****po**

shippingaddress

communication

class InvManage.views.SalesInvoice

Bases: django.db.models.Model

Model for the sales order invoice.

company

Primary key of the Company.

Type *Company*

so

Primary key of the SalesOrder.

Type *SalesOrder*

shippingaddress

Primary key of the ShippingAddress.

Type *ShippingAddress*

company

so

shippingaddress

class InvManage.views.FilterState

Bases: django.db.models.Model

Model of the filter state.

name

Name of the filter state.

Type str

name

class InvManage.views.FilterColumn

Bases: django.db.models.Model

Model of the column state.

name

Name of the column.

Type str

label

Label of the column.

Type str

visible

Visibility flag.

Type bool

position

Index of the column.

Type int

state

Primary key of the `FilterState` instance being referenced by the column instance.

Type *FilterState*

name**label****visible****position****state****class** `InvManage.views.Object`

Bases: `django.db.models.Model`

Model for a generic object.

identifier

Unique identifier of the object.

Type `str`

name

Name of the object.

Type `str`

model

Model associated with the object.

Type `str`

identifier**name****model****class** `InvManage.views.EventCard`

Bases: `django.db.models.Model`

Model for event card to be displayed in the history view.

obj

Object associated with the event.

Type `OneToOneField`

objId

Unique identifier of the object.

Type `str`

objname

Name of the object.

Type `str`

objmodel

Model of the object.

Type `str`

date

Timestamp of the event.

Type DateField

operation
Type of event.

Type str

obj

objId

objname

objmodel

date

operation

class InvManage.views.EventType
Bases: django.db.models.Model
Model for event type.

name
Name of the event being triggered.

Type str

label
Label associated with the event.

Type str

Returns Returns the string representation of the label when the event type is queried.

Return type Label

name

label

__str__(self)

class InvManage.views.ObjectModel
Bases: django.db.models.Model
Model associated with an object.

name
Name of the object.

Type str

label
Label string of the object.

Type str

modName
Name of the model associated with the object.

Type str

Returns Returns string representation of the label when object model is queried.

Return type Label


```

    name
    label
    modName
    __str__(self)
class InvManage.views.HistoryFilterState
    Bases: django.db.models.Model
    Model of filter state of the history.
    params
        Paramameters of the configured filters.
        Type str
    params
class InvManage.views.ShippingAddress
    Bases: django.db.models.Model
    Model of the shipping address.
    title
        Title of the firm.
        Type str
    name
        Name of the firm.
        Type str
    phone
        Contact number of the firm.
        Type str
    address
        Postal address.
        Type str
    city
        City.
        Type str
    state
        State.
        Type str
    country
        Country.
        Type str
    website
        Official website of the firm.
        Type URLField
    post
        Postal code of the firm.

```

```

        Type str

    title
    name
    phone
    address
    city
    state
    country
    website
    post
class InvManage.views.PurchaseData
    Bases: django.db.models.Model

    Model of the purchase data.

    currency
        Currency of purchase.

        Type str

    minorder
        Minimum order quantity.

        Type int

    contactperson
        Name of the contact person.

        Type str

    refcode
        Reference code.

        Type str

    transportmode
        Mode of transport of the shipment.

        Type str

    currency
    minorder
    contactperson
    refcode
    transportmode
class InvManage.views.Communication
    Bases: django.db.models.Model

    Model of the communication information.

    language
        Language of communication.

        Type str
```

phone

Contact number of the firm.

Type str**email**

E-mail of the contact person.

Type EmailField**fax**

Fax number.

Type str**language****phone****email****fax****class** InvManage.views.**BankAccount**

Bases: django.db.models.Model

Model of the bank account details.

name

Name of the bank.

Type str**branch**

Branch number of the bank.

Type str**region**

City in which the branch is located.

Type str**route**

Transit number.

Type str**number**

Bank account number.

Type int**accttype**

Type of bank account.

Type str**iban**

IBAN number.

Type str**code**

Bank code.

Type str

branchcode

Branch code.

Type str**name****branch****region****route****number****acctype****iban****code****branchcode****InvManage.views.get_columns(*state*)**

Gets the columns of the display table.

This function takes a filter state and returns an array of column names that are ordered depending on the position attribute of the columns. This order is important as the queryset will be rendered in the display table according to the column order in this returned array.

Parameters **state** ([FilterState](#)) – Filter state of the columns.**Returns** A list that dictates the order of the columns.**Return type** List**InvManage.views.sort_ascending_descending(*request, model*)**

Sorts and saves the state of the column.

This function takes a request, finds which column needs to be sorted in ascending/descending order, and returns the sorted queryset.

Parameters

- **request** (*HttpRequest*) – GET request received for retrieving the list of model instances.
- **model** (*Model*) – Model of the objects being retrieved.

Returns List of sorted object instances.**Return type** QuerySet**InvManage.views.change_column_position(*request, state*)**

Modifies the column states depending on the request.

This function:

- Takes a filter state
- Extracts the column names into an order array
- Modifies the column order in the array depending on user input direction (left/right)
- Saves this modified state of columns into the database
- Returns the modified column names array

Parameters

- **request** (*HttpRequest*) – GET request for retrieving the state of the columns.
- **state** (*FilterState*) – Filter state of the columns.

Returns List of sorted columns.

Return type List

`InvManage.views.paginate(queryset, filter, page_number)`

Paginates the list of objects.

This function takes the entire queryset and filters out only objects belonging to the request page

Parameters

- **queryset** (*QuerySet*) – QuerySet of the objects retrieved from the database.
- **filter** (*FilterState*) – Filter state of the columns.
- **page_number** (*int*) – Page number requested.

class `InvManage.views.CompanySerializer`

Bases: `rest_framework.serializers.ModelSerializer`

Serializer for Company class instance.

The `CompanySerializer.data` attribute gives the JSON serialized data of the Company instance:

```
{
  "name": "Fringillami",
  "owner": "Ivor Barnett",
  "gstln": "89AAC056465468",
  "phone": "332 220-7026",
  "address": "Ap #849-6241 Euismod Av., 677598, Carinthia, Belgium",
  "email": "est.tempor@fringillami.org",
  "location": "Belgium",
  "image": "/media/images/logo.png"
}
```

class `Meta`

model

fields = ['name', 'owner', 'gstln', 'phone', 'address', 'email', 'location', 'image']

class `InvManage.views.ShippingAddressSerializer`

Bases: `rest_framework.serializers.ModelSerializer`

Serializer for ShippingAddress class instance.

The `ShippingAddressSerializer.data` attribute gives the JSON serialized data of the ShippingAddress instance:

```
{
  "name": "Harding Gross",
  "address": "8798 At, St., 7639",
  "city": "Rome",
  "phone": "936 651-4847",
  "state": "Lazio",
}
```

(continues on next page)

(continued from previous page)

```

"country": "Italy",
"post": "300326"
}

```

```
class Meta
```

```
    model
```

```
    fields = ['name', 'address', 'city', 'phone', 'state', 'country', 'post']
```

```
class InvManage.views.CompanyFilter
```

```
    Bases: django_filters.FilterSet
```

```
class Meta
```

```
    model
```

```
    fields
```

```
context
```

```
name
```

```
owner
```

```
phone
```

```
email
```

```
location
```

```
order_by_field = ordering
```

```
ordering
```

```
InvManage.views.create_company_view(request)
```

Creates a company on POST request and returns a company creation form GET request.

GET /company

Gets the company creation form.

Example request:

```

GET /company/ HTTP/1.1
Host: localhost:8000
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/
webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9

```

Example response:

```

HTTP/1.1 200 OK
Vary: Accept
Content-Type: text/html; charset=utf-8

```

Request Headers

- **Accept** – text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9

Status Codes

- **200 OK** – Company creation form received successfully.

POST /company

Creates a company.

Example request:

```
POST /company/ HTTP/1.1
Host: localhost:8000
Content-Type: multipart/form-data;
```

Form Parameters

- **comp-name** – Fringillami
- **comp-owner** – Ivor Barnett
- **comp-gstin** – 89AAC056465468
- **comp-phone** – 332 220-7026
- **comp-address** – Ap #849-6241 Euismod Av., 677598, Carinthia, Belgium
- **comp-email** – est.tempor@fringillami.org
- **comp-location** – Belgium
- **ship-title** – FingDocks
- **ship-name** – Harding Gross
- **ship-phone** – 936 651-4847
- **ship-address** – 8798 At, St., 7639
- **ship-city** – Rome
- **ship-state** – Lazio
- **ship-country** – Italy
- **ship-website** – fringdocs.com
- **ship-post** – 300326

Response Headers

- **Content-Type** – multipart/form-data; boundary=——WebKitFormBoundaryLTR88aZAnBUSE7mv

Status Codes

- **302 Found** – Redirects to /company.

`InvManage.views.update_company_view(request)`

Updates a company on POST request and returns the company update form GET request.

GET /company/update

Gets the company update form whose primary key matches the query parameter pk.

Example request:

```
GET /company/update HTTP/1.1
Host: localhost:8000
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
```

Query Parameters

- **pk** – The primary key of the company.

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: text/html; charset=utf-8
```

Request Headers

- **Accept** – text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9

Status Codes

- **200 OK** – Company update form received successfully.

POST /company/update

Updates a company.

Example request:

```
POST /company/ HTTP/1.1
Host: localhost:8000
Content-Type: multipart/form-data;
```

Form Parameters

- **pk** – 4
- **comp-name** – Fringillami
- **comp-owner** – Ivor Barnett
- **comp-gstin** – 89AAC056465468
- **comp-phone** – 332 220-7026
- **comp-address** – Ap #849-6241 Euismod Av., 677598, Carinthia, Belgium
- **comp-email** – est.tempor@fringillami.org
- **comp-location** – Belgium
- **ship-title** – FingDocks
- **ship-name** – Harding Gross
- **ship-phone** – 936 651-4847
- **ship-address** – 8798 At, St., 7639
- **ship-city** – Rome
- **ship-state** – Lazio
- **ship-country** – Italy
- **ship-website** – fringdocs.com
- **ship-post** – 300326

Response Headers

- **Content-Type** – multipart/form-data; boundary=—WebKitFormBoundaryLTR88aZAnBUSE7mv

Status Codes

- **302 Found** – Redirects to /company.

`InvManage.views.delete_company_view(request, pk)`

Deletes the company with primary key pk on POST request.

POST /company/<str:object_id>/delete

Deletes the company represented by the primary key object_id.

Example request:

```
POST /company/32/delete HTTP/1.1
Host: localhost:8000
Content-Type: application/x-www-form-urlencoded
```

Parameters

- **object_id** – Company primary key.

Response Headers

- **Content-Type** – application/x-www-form-urlencoded

Status Codes

- **302 Found** – Redirects to /company.
- **500 Internal Server Error** – Company matching query does not exist.

`InvManage.views.display_companies_view(request)`

Retrieves the list of companies on GET request.

GET /companies/

Gets the list of all companies.

Example request:

```
GET /companies/ HTTP/1.1
Host: localhost:8000
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
```

Form Parameters

- **page** – The page number of the companies list.

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: text/html; charset=utf-8
```

Request Headers

- **Accept** – text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9

Status Codes

- 200 OK – List of companies received successfully.

```
class InvManage.views.ProductBasicInfoForm
    Bases: django.forms.Form

    Form for basic product information.

    name
        Name of the product.

        Type str

    item_type
        Type of the product.

        Type str

    category
        Product category.

        Type str

    description
        Short description of the product.

        Type str

    prefix = basic

    context
        name
        item_type
        category
        description

class InvManage.views.ProductDetailedInfoForm
    Bases: django.forms.Form

    Form for product physical details

    length
        Length of the product.

        Type str

    width
        Width of the product.

        Type str

    height
        Height of the product.

        Type str

    weight
        Weight of the product.

        Type str

    prefix = detailed
```

```

    context
    length
    width
    height
    weight
class InvManage.views.ProductPricingForm
    Bases: django.forms.Form
    Form for pricing information of the product.
    price
        Price of the product.
        Type float
    discount
        Default discount percentage on the product.
        Type float
    prefix = pricing
    context
    price
    discount
class InvManage.views.ProductPurchaseEntryForm
    Bases: django.forms.Form
    Form for product purchase entry.
    ppe_id
        Unique identifier of the product purchase entry.
        Type int
    product
        Product associated wit the product purchase entry.
        Type Product
    quantity
        Quantity of the product to be ordered.
        Type int
    price
        Price of the product.
        Type float
    discount
        Percentage discount on the product purchase.
        Type float
    prefix = form
    context
    ppe_id

```

product

quantity

price

discount

class InvManage.views.ProductSalesEntryForm

Bases: django.forms.Form

Form for product sales entry.

pse_id

Unique identifier of the product sales entry.

Type int

product

Product associated with the product sales entry.

Type *Product*

quantity

Quantity of the product.

Type int

price

Price of the product.

Type float

discount

Percentage discount on the product.

Type float

prefix = form

context

pse_id

product

quantity

price

discount

class InvManage.views.ProductStatusForm

Bases: django.forms.Form

Form for locating and identifying the product.

quantity

Stock quantity of the product.

Type int

identifier

Unique identifier of the product.

Type str

location

Physical location of the product.

Type str**context****prefix = status****quantity****identifier****location****class** InvManage.views.ProductStorageInfoForm

Bases: django.forms.Form

Form for storage information.

barcode

Barcode of the product.

Type str**expiry**

Expiry date of the product.

Type DateField**prefix = storage****barcode****context****expiry****class** InvManage.views.PurchaseOrderBasicInfo

Bases: django.forms.Form

Form for basic information of the purchase order.

vendor

Vendor associated with the purchase order.

Type *Vendor***date**

Date of the purchase order creation.

Type DateField**po**

Purchase order number.

Type int**discount**

Percentage of overall discount.

Type float**tax**

Percentage of tax.

Type float

paid

Amount paid to the vendor.

Type float

balance

Balance amount to be paid to the vendor.

Type float

subtotal

Total of all the product purchase entries associated with the purchase order.

Type float

taxtotal

Total tax applicable on the *subtotal*.

Type float

ordertotal

Total price of the purchase order including *taxtotal*.

Type float

prefix = po**context****product_choices****vendor_choices****vendor****date****po****discount****tax****paid****balance****subtotal****taxtotal****ordertotal****class** InvManage.views.GRNInfo

Bases: django.forms.Form

Form for goods receipt note (GRN).

vendor

Vendor associated with the goods receipt note.

Type *Vendor*

poRef

List of identifiers of the purchase orders from which the goods receipt note is derived.

Type *PurchaseOrder*

identifier

Unique identifier of the goods receipt note.

Type str

date

Date of GRN creation.

Type DateField

grnType

Type of GRN (auto or manual).

Type str

amendNumber

Amendment number of the GRN.

Type int

amendDate

Amendment date.

Type DateField

transporter

Name of the transport/shipping service.

Type str

vehicleNumber

Vehicle number using which the products are shipped.

Type str

gateInwardNumber

Gate inward number of the vehicle.

Type str

preparedBy

Name/identifier of the person who created the goods receipt note.

Type str

checkedBy

Name/identifier of the person who validated the goods receipt note.

Type str

checkedBy

Name/identifier of the person who inspected the physical products in the goods receipt note.

Type str

approvedBy

Name/identifier of the authority who approved the goods receipt note.

Type str

prefix = grn**context****product_choices****vendor_choices**

```
vendor
TYPE_CHOICES = [['manual', 'Blank'], ['auto', 'PO Reference']]
poRef
identifier
date
grnType
amendNumber
amendDate
transporter
vehicleNumber
gateInwardNumber
preparedBy
checkedBy
inspectedBy
approvedBy
class InvManage.views.GRNEntryForm
    Bases: django.forms.Form
    Form for goods receipt note entry (GRNE).
    product
        Product associated with the goods receipt note entry.
        Type Product
    grne_id
        Unique identifier of the GRNE.
        Type int
    ppe_id
        Unique identifier of the product purchase entry associated with the GRNE.
        Type int
    quantity
        Ordered quantity of product with reference to product purchase entry.
        Type int
    remark
        Remarks of the quality engineer or the GRN creator about status of products received.
        Type str
    receivedQty
        Quantity of product received against the ordered quantity.
        Type str
    acceptedQty
        Quantity of product accepted as OK.
        Type str
```


rejectedQty
 Quantity of product rejected (not OK, on HOLD, extra delivery, etc.)
Type str

prefix = form

context

product

grne_id

ppe_id

quantity

remark

receivedQty

acceptedQty

rejectedQty

class InvManage.views.**ThumbnailForm**

Bases: django.forms.Form

For for product thumbnail image.

image
 Image of the product.
Type ImageField

prefix = thumbnail

image

class InvManage.views.**SalesOrderBasicInfo**

Bases: django.forms.Form

Form for basic information on sales order.

consumer
 Consumer associated with the sales order.
Type *Consumer*

date
 Date of the sales order creation.
Type DateField

so
 Sales order number.
Type int

discount
 Percentage discount associated with the sales order.
Type float

tax
 Percentage of tax applicable.
Type float

paid

Amount received from the consumer.

Type float

balance

Amount balance with the consumer.

Type float

subtotal

Total of all the product sales entries associated with the sales order.

Type float

taxtotal

Total tax applicable on the *subtotal*.

Type float

ordertotal

Total price of the sales order including *taxtotal*.

Type float

prefix = so**context****product_choices****consumer_choices****consumer****date****so****discount****tax****paid****balance****subtotal****taxtotal****ordertotal****class** InvManage.views.ShippingAddressForm

Bases: django.forms.Form

For for shipping address.

title

Title of the firm.

Type str

name

Name of the firm.

Type str

phone
Contact number of the firm.

Type str

address
Postal address.

Type str

city
City.

Type str

state
State.

Type str

country
Country.

Type str

website
Official website of the firm.

Type URLField

post
Postal code of the firm.

Type str

prefix = ship

context

title

name

phone

address

city

state

country

website

post

```
class InvManage.views.VendorForm
    Bases: django.forms.Form
    Form for basic informaion of the vendor.
```

name
Name of the vendor.

Type str

identifier

Unique identifier of the vendor.

Type str

gstn

GSTIN number of the vendor.

Type str

prefix = vend

context

name

identifier

gstn

class InvManage.views.ConsumerForm

Bases: django.forms.Form

Form for consumer.

name

Name of the consumer.

Type str

identifier

Unique identifier of the consumer.

Type str

gstn

GSTIN number of the consumer.

Type str

phone

Contact number.

Type str

address

Address of the consumer.

Type str

email

E-mail address of the consumer.

Type str

location

City of the consumer.

Type str

prefix = consumer

context

name

identifier

```

    gstin
    phone
    address
    email
    location
class InvManage.views.CompanyForm
    Bases: django.forms.Form
    Form for company.
    name
        Name of the company.
        Type str
    owner
        Name of the owner of the company.
        Type str
    gstin
        GSTIN number of the company.
        Type str
    phone
        Contact number of the company.
        Type str
    address
        Postal address.
        Type str
    email
        E-mail address of the contact person.
        Type str
    location
        City of the company.
        Type str
    prefix = comp
    context
    name
    owner
    gstin
    phone
    address
    email
    location

```

```
class InvManage.views.HistoryForm
```

```
    Bases: django.forms.Form
```

```
    Form for history view.
```

```
    qlen
```

```
        Count of the events visible on the history view.
```

```
        Type int
```

```
    prefix = history
```

```
    context
```

```
    qlen
```

```
class InvManage.views.CommunicationForm
```

```
    Bases: django.forms.Form
```

```
    Form for communication information.
```

```
    language
```

```
        Language of communication.
```

```
        Type str
```

```
    phone
```

```
        Contact number of the firm.
```

```
        Type str
```

```
    email
```

```
        E-mail of the contact person.
```

```
        Type EmailField
```

```
    fax
```

```
        Fax number.
```

```
        Type str
```

```
    prefix = com
```

```
    context
```

```
    language
```

```
    phone
```

```
    email
```

```
    fax
```

```
class InvManage.views.PurchaseDataForm
```

```
    Bases: django.forms.Form
```

```
    Form for purchase related data.
```

```
    currency
```

```
        Currency of purchase.
```

```
        Type str
```

```
    minorder
```

```
        Minimum order quantity.
```

```
        Type int
```

contactperson

Name of the contact person.

Type str**refcode**

Reference code.

Type str**transportmode**

Mode of transport of the shipment.

Type str**prefix = pdform****context****currency****minorder****contactperson****refcode****transportmode****class** InvManage.views.**BankAccountForm**

Bases: django.forms.Form

Form for bank account details.

name

Name of the bank.

Type str**branch**

Branch number of the bank.

Type str**region**

City in which the branch is located.

Type str**route**

Transit number.

Type str**number**

Bank account number.

Type int**accttype**

Type of bank account.

Type str**iban**

IBAN number.

Type str

code

Bank code.

Type str**branchcode**

Branch code.

Type str**prefix = bank****context****name****branch****region****route****number****acctype****iban****code****branchcode****class** InvManage.views.Product

Bases: django.db.models.Model

Model of the product.

name

Name of the product.

Type str**category**

Product category.

Type str**item_type**

Type of the product.

Type str**description**

Short description of the product.

Type str**price**

Price of the product.

Type float**quantity**

Stock quantity of the product.

Type int

identifier

Unique identifier of the product.

Type str

location

Physical location of the product.

Type str

length

Length of the product.

Type str

width

Width of the product.

Type str

height

Height of the product.

Type str

weight

Weight of the product.

Type str

discount

Default discount percentage on the product.

Type float

barcode

Barcode of the product.

Type str

expiry

Expiry date of the product.

Type DateField

image

Image of the product.

Type ImageField

name**category****item_type****description****price****quantity****identifier****location****length**

width
height
weight
discount
barcode
expiry
image
__str__(*self*)

class `InvManage.views.ProductPurchaseEntry`

Bases: `django.db.models.Model`

Model of the product purchase entry.

product

Product associated with the product purchase entry.

Type `ModelChoiceField`

quantity

Quantity of the product to be ordered.

Type `int`

price

Price of the product.

Type `float`

discount

Percentage discount on the product purchase.

Type `float`

order

Referenced purchase order.

Type `PurchaseOrder`

receivedQty

Quantity of product received against the ordered quantity.

Type `str`

acceptedQty

Quantity of product accepted as OK.

Type `str`

rejectedQty

Quantity of product rejected (not OK, on HOLD, extra delivery, etc.)

Type `str`

is_complete()

Returns the completion status (boolean).

pending_quantity()

Returns the pending quantity against the entry (int).

product

```

    quantity
    price
    discount
    order
    receivedQty
    acceptedQty
    rejectedQty
    is_complete(self)
    property pending_quantity(self)
class InvManage.views.GRNEntry
    Bases: django.db.models.Model

    Model of the goods receipt note entry (GRNE).

    product
        Primary key of the Product instance associated with the goods receipt note entry.

        Type Product

    quantity
        Ordered quantity of product with reference to product purchase entry.

        Type int

    grne
        Primary key of the GoodsReceiptNote instance.

        Type int

    ppe
        Primary key of the ProductPurchaseEntry instance associated with the GRNE.

        Type int

    remark
        Remarks of the quality engineer or the GRN creator about status of products received.

        Type str

    receivedQty
        Quantity of product received against the ordered quantity.

        Type str

    acceptedQty
        Quantity of product accepted as OK.

        Type str

    rejectedQty
        Quantity of product rejected (not OK, on HOLD, extra delivery, etc.)

        Type str

    product
    quantity
    grn

```

ppes

remark

receivedQty

acceptedQty

rejectedQty

class InvManage.views.ProductSalesEntry

Bases: django.db.models.Model

Model of the product sales entry.

product

Primary key of the Product associated with the product sales entry.

Type *Product*

quantity

Quantity of the product.

Type int

price

Price of the product.

Type float

discount

Percentage discount on the product.

Type float

order

Primary key of the SalesOrder referenced by the entry.

Type *SalesOrder*

product

quantity

price

discount

order

class InvManage.views.SalesOrder

Bases: django.db.models.Model

Model of the sales order.

consumer

Consumer associated with the sales order.

Type ModelChoiceField

date

Date of the sales order creation.

Type DateField

so

Sales order number.

Type int

discount

Percentage discount associated with the sales order.

Type float

tax

Percentage of tax applicable.

Type float

paid

Amount received from the consumer.

Type float

balance

Amount balance with the consumer.

Type float

subtotal

Total of all the product sales entries associated with the sales order.

Type float

taxtotal

Total tax applicable on the *subtotal*.

Type float

ordertotal

Total price of the sales order including *taxtotal*.

Type float

consumer

date

so

discount

tax

paid

balance

subtotal

taxtotal

ordertotal

class InvManage.views.**PurchaseOrder**

Bases: django.db.models.Model

Model of the purchase order.

vendor

Vendor associated with the purchase order.

Type *Vendor*

date
Date and time of the purchase order creation.
Type DateField

po
Purchase order number.
Type int

discount
Percentage discount.
Type float

tax
Percentage of tax applicable for the purchase.
Type float

paid
Amount paid against the PO.
Type float

balance
Amount balance which remains to be paid.
Type float

subtotal
Total of all the product purchase entries associated with the purchase order.
Type float

taxtotal
Total tax applicable on the *subtotal*.
Type float

ordertotal
Total price of the purchase order including *taxtotal*.
Type float

is_complete()
Returns the completion status (boolean).

pending_ppes()
Returns list of product purchase entries which are not completed.

vendor

date

po

discount

tax

paid

balance

subtotal

taxtotal

```

ordertotal
is_complete(self)
pending_ppes(self)
class InvManage.views.GoodsReceiptNote
    Bases: django.db.models.Model
    Model of the goods receipt note (GRN).
    vendor
        Vendor associated with the goods receipt note.
        Type ModelChoiceField
    poRef
        List of identifiers of the purchase orders from which the goods receipt note is derived.
        Type MultipleChoiceField
    identifier
        Unique identifier of the goods receipt note.
        Type str
    date
        Date of GRN creation.
        Type DateField
    grnType
        Type of GRN (auto or manual).
        Type ChoiceField
    amendNumber
        Amendment number of the GRN.
        Type int
    amendDate
        Amendment date.
        Type DateField
    transporter
        Name of the transport/shipping service.
        Type str
    vehicleNumber
        Vehicle number using which the products are shipped.
        Type str
    gateInwardNumber
        Gate inward number of the vehicle.
        Type str
    preparedBy
        Name/identifier of the person who created the goods receipt note.
        Type str

```

checkedBy

Name/identifier of the person who validated the goods receipt note.

Type str

checkedBy

Name/identifier of the person who inspected the physical products in the goods receipt note.

Type str

approvedBy

Name/identifier of the authority who approved the goods receipt note.

Type str

TYPE_CHOICES = [['manual', 'Blank'], ['auto', 'PO Reference']]

vendor**poRef****identifier****date****grnType****amendNumber****amendDate****transporter****vehicleNumber****gateInwardNumber****preparedBy****checkedBy****inspectedBy****approvedBy**

class InvManage.views.Consumer

Bases: django.db.models.Model

Model of the consumer.

name

Name of the consumer.

Type str

identifier

Unique identifier of the consumer.

Type str

gstIn

GSTIN number of the consumer.

Type str

phone

Contact number.

Type str

address

Address of the consumer.

Type str**email**

E-mail address of the consumer.

Type str**location**

City of the consumer.

Type str**Returns** [description]**Return type** [type]**name****identifier****gstin****phone****address****email****location****__str__**(*self*)**class** InvManage.views.**Vendor**

Bases: django.db.models.Model

Model of the vendor.

name

Name of the vendor.

Type str**identifier**

Unique identifier of the vendor.

Type str**gstin**

GSTIN number of the vendor.

Type str**address**

Primary key of the ShippingAddress instance associated with the vendor.

Type *ShippingAddress***communication**

Primary key of the Communication instance associated with the vendor.

Type *Communication***backaccount**

Primary key of the BankAccount instance associated with the vendor.

Type *BankAccount*

purchasedata

Primary key of the PurchaseData instance associated with the vendor.

Type *PurchaseData*

Returns String representation of the vendor name.

Return type str

name

identifier

gstn

address

communication

bankaccount

purchasedata

__str__(self)

class InvManage.views.**Company**

Bases: django.db.models.Model

Model of the company.

name

Name of the company.

Type str

owner

Owner of the company.

Type str

gstn

GSTIN number of the company.

Type str

phone

Contact number.

Type str

address

Postal address.

Type str

email

E-mail address of the contact person.

Type str

location

City of the company.

Type str

image

Photo/logo of the company.

Type `ImageField`

shippingaddress

Primary key of the `ShippingAddress` instance associated with the company.

Type `ShippingAddress`

Returns String representation of the company name.

Return type `str`

name**owner****gsttin****phone****address****email****location****image****shippingaddress****__str__** (*self*)

class `InvManage.views.PurchaseInvoice`

Bases: `django.db.models.Model`

Model for the purchase invoice.

company

Primary key of the `Company`.

Type `Company`

po

Primary key of the `PurchaseOrder`.

Type `PurchaseOrder`

shippingaddress

Primary key of the `ShippingAddress`.

Type `ShippingAddress`

communication

Primary key of the `Communication`.

Type `Communication`

company**po****shippingaddress****communication**

class `InvManage.views.SalesInvoice`

Bases: `django.db.models.Model`

Model for the sales order invoice.

company

Primary key of the Company.

Type *Company*

so

Primary key of the SalesOrder.

Type *SalesOrder*

shippingaddress

Primary key of the ShippingAddress.

Type *ShippingAddress*

company

so

shippingaddress

class `InvManage.views.FilterState`

Bases: `django.db.models.Model`

Model of the filter state.

name

Name of the filter state.

Type `str`

name

class `InvManage.views.FilterColumn`

Bases: `django.db.models.Model`

Model of the column state.

name

Name of the column.

Type `str`

label

Label of the column.

Type `str`

visible

Visibility flag.

Type `bool`

position

Index of the column.

Type `int`

state

Primary key of the `FilterState` instance being referenced by the column instance.

Type *FilterState*

```

    name
    label
    visible
    position
    state
class InvManage.views.Object
    Bases: django.db.models.Model
    Model for a generic object.
    identifier
        Unique identifier of the object.
        Type str
    name
        Name of the object.
        Type str
    model
        Model associated with the object.
        Type str
    identifier
    name
    model
class InvManage.views.EventCard
    Bases: django.db.models.Model
    Model for event card to be displayed in the history view.
    obj
        Object associated with the event.
        Type OneToOneField
    objId
        Unique identifier of the object.
        Type str
    objname
        Name of the object.
        Type str
    objmodel
        Model of the object.
        Type str
    date
        Timestamp of the event.
        Type DateField
    operation
        Type of event.

```

```

        Type str
    obj
    objId
    objname
    objmodel
    date
    operation
class InvManage.views.EventType
    Bases: django.db.models.Model
    Model for event type.
    name
        Name of the event being triggered.
        Type str
    label
        Label associated with the event.
        Type str
        Returns Returns the string representation of the label when the event type is queried.
        Return type Label
    name
    label
    __str__(self)
class InvManage.views.ObjectModel
    Bases: django.db.models.Model
    Model associated with an object.
    name
        Name of the object.
        Type str
    label
        Label string of the object.
        Type str
    modName
        Name of the model associated with the object.
        Type str
        Returns Returns string representation of the label when object model is queried.
        Return type Label
    name
    label
```

```

modName
    __str__(self)
class InvManage.views.HistoryFilterState
    Bases: django.db.models.Model
    Model of filter state of the history.
    params
        Paramameters of the configured filters.
        Type str
    params
class InvManage.views.ShippingAddress
    Bases: django.db.models.Model
    Model of the shipping address.
    title
        Title of the firm.
        Type str
    name
        Name of the firm.
        Type str
    phone
        Contact number of the firm.
        Type str
    address
        Postal address.
        Type str
    city
        City.
        Type str
    state
        State.
        Type str
    country
        Country.
        Type str
    website
        Official website of the firm.
        Type URLField
    post
        Postal code of the firm.
        Type str
    title

```

name
phone
address
city
state
country
website
post

class InvManage.views.**PurchaseData**

Bases: django.db.models.Model

Model of the purchase data.

currency
Currency of purchase.

Type str

minorder
Minimum order quantity.

Type int

contactperson
Name of the contact person.

Type str

refcode
Reference code.

Type str

transportmode
Mode of transport of the shipment.

Type str

currency

minorder

contactperson

refcode

transportmode

class InvManage.views.**Communication**

Bases: django.db.models.Model

Model of the communication information.

language
Language of communication.

Type str

phone
Contact number of the firm.


```

        Type str
    email
        E-mail of the contact person.
        Type EmailField
    fax
        Fax number.
        Type str
    language
    phone
    email
    fax
class InvManage.views.BankAccount
    Bases: django.db.models.Model
    Model of the bank account details.
    name
        Name of the bank.
        Type str
    branch
        Branch number of the bank.
        Type str
    region
        City in which the branch is located.
        Type str
    route
        Transit number.
        Type str
    number
        Bank account number.
        Type int
    accttype
        Type of bank account.
        Type str
    iban
        IBAN number.
        Type str
    code
        Bank code.
        Type str
    branchcode
        Branch code.

```

Type str

name
branch
region
route
number
accttype
iban
code
branchcode

`InvManage.views.get_columns(state)`

Gets the columns of the display table.

This function takes a filter state and returns an array of column names that are ordered depending on the position attribute of the columns. This order is important as the queryset will be rendered in the display table according to the column order in this returned array.

Parameters `state` ([FilterState](#)) – Filter state of the columns.

Returns A list that dictates the order of the columns.

Return type List

`InvManage.views.sort_ascending_descending(request, model)`

Sorts and saves the state of the column.

This function takes a request, finds which column needs to be sorted in ascending/descending order, and returns the sorted queryset.

Parameters

- **request** (*HttpRequest*) – GET request received for retrieving the list of model instances.
- **model** (*Model*) – Model of the objects being retrieved.

Returns List of sorted object instances.

Return type QuerySet

`InvManage.views.change_column_position(request, state)`

Modifies the column states depending on the request.

This function:

- Takes a filter state
- Extracts the column names into an order array
- Modifies the column order in the array depending on user input direction (left/right)
- Saves this modified state of columns into the database
- Returns the modified column names array

Parameters

- **request** (*HttpRequest*) – GET request for retrieving the state of the columns.
- **state** ([FilterState](#)) – Filter state of the columns.

Returns List of sorted columns.

Return type List

`InvManage.views.paginate(queryset, filter, page_number)`

Paginates the list of objects.

This function takes the entire queryset and filters out only objects belonging to the request page

Parameters

- **queryset** (*QuerySet*) – QuerySet of the objects retrieved from the database.
- **filter** (*FilterState*) – Filter state of the columns.
- **page_number** (*int*) – Page number requested.

class `InvManage.views.VendorFilter`

Bases: `django_filters.FilterSet`

class `Meta`

model

fields

context

name

identifier

phone

email

location

order_by_field = `ordering`

ordering

class `InvManage.views.VendorSerializer`

Bases: `rest_framework.serializers.ModelSerializer`

Serializer for Vendor class instance.

The `VendorSerializer.data` attribute gives the JSON serialized data of the Vendor instance:

```
{
  "name": "Harding Gross",
  "identifier": "HG",
  "gstn": "89ACC16843543",
  "address": {
    "name": "Harding Gross",
    "address": "8798 At, St., 7639",
    "city": "Rome",
    "phone": "936 651-4847",
    "state": "Lazio",
    "country": "Italy",
    "post": "300326"
  }
}
```

```
class Meta
```

```
    model
```

```
    fields = ['name', 'identifier', 'gstin', 'address']
```

```
    phone
```

```
    email
```

```
address
```

```
to_representation(self, instance)
```

```
InvManage.views.create_vendor_view(request)
```

Creates a vendor on POST request, and returns a vendor creation form on GET request.

GET /vendor

Gets the vendor creation form.

Example request:

```
GET /vendor/ HTTP/1.1
Host: localhost:8000
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/
webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: text/html; charset=utf-8
```

Request Headers

- **Accept** – text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9

Status Codes

- **200 OK** – Vendor creation form received successfully.

POST /vendor

Creates a vendor.

Example request:

```
POST /vendor/ HTTP/1.1
Host: localhost:8000
Content-Type: multipart/form-data;
```

Form Parameters

- **vend-name** – Lug Vendor
- **vend-identifier** – TBPB-02692
- **vend-gstin** – 89AAC254254F2
- **ship-title** – AKATSUKI
- **ship-name** – Kuame Burns

- **ship-phone** – 679 166-3127
- **ship-address** – Nonummy Avenue
- **ship-city** – Chung Cheong
- **ship-state** – Guanacaste
- **ship-country** – tellusidnunc.net
- **ship-website** – Germany.protitor@tellusid.net
- **ship-post** – 8949
- **pdform-currency** – DEM
- **pdform-minorder** – 2000
- **pdform-contactperson** – Harding Gross
- **pdform-refcode** – CUST000124
- **pdform-transportmode** – Express
- **com-language** – German
- **com-phone** – 936 651-4817
- **com-email** – non.sollicitudin@uttincidunt.org
- **com-fax** – 323 555 1234
- **bank-name** – FIRST FLORIDA INTEGRITY BANK
- **bank-branch** – Bavaria
- **bank-region** – Bayem
- **bank-route** – 67016325
- **bank-number** – 42543251393
- **bank-acctype** – Current
- **bank-iban** – DE6233542
- **bank-code** – BA54354354
- **bank-branchcode** – BA35435823

Response Headers

- **Content-Type** – multipart/form-data; boundary=——WebKitFormBoundaryLTR88aZAnBUSE7mv

Status Codes

- **302 Found** – Redirects to /vendor.

`InvManage.views.update_vendor_view(request)`

Updates a vendor on POST request and returns the vendor update form for GET request.

GET /vendor/update

Gets the vendor update form whose primary key matches the query parameter pk.

Example request:

```
GET /vendor/update HTTP/1.1
Host: localhost:8000
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/
webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
```

Query Parameters

- **pk** – The primary key of the vendor.

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: text/html; charset=utf-8
```

Request Headers

- **Accept** – text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9

Status Codes

- **200 OK** – Vendor update form received successfully.

POST /vendor/update

Updates the vendor.

Example request:

```
POST /vendor/update HTTP/1.1
Host: localhost:8000
Content-Type: multipart/form-data;
```

Form Parameters

- **vend-name** – Lug Vendor
- **vend-identifier** – TBPB-02692
- **vend-gstin** – 89AAC254254F2
- **ship-title** – AKATSUKI
- **ship-name** – Kuame Burns
- **ship-phone** – 679 166-3127
- **ship-address** – Nonummy Avenue
- **ship-city** – Chung Cheong
- **ship-state** – Guanacaste
- **ship-country** – tellusidnunc.net
- **ship-website** – Germany.protitor@tellusid.net
- **ship-post** – 8949
- **pdform-currency** – DEM
- **pdform-minorder** – 1000

- **pdform-contactperson** – Harding Gross
- **pdform-refcode** – CUST000124
- **pdform-transportmode** – Express
- **com-language** – German
- **com-phone** – 936 651-4817
- **com-email** – non.sollicitudin@uttincidunt.org
- **com-fax** – 323 555 1234
- **bank-name** – FIRST FLORIDA INTEGRITY BANK
- **bank-branch** – Bavaria
- **bank-region** – Bayem
- **bank-route** – 67016325
- **bank-number** – 42543251393
- **bank-acctype** – Current
- **bank-iban** – DE6233542
- **bank-code** – BA54354354
- **bank-branchcode** – BA35435823

Response Headers

- **Content-Type** – multipart/form-data; boundary=—WebKitFormBoundaryLTR88aZAnBUSE7mv

Status Codes

- **302 Found** – Redirects to /consumer.

`InvManage.views.delete_vendor_view(request, pk)`

Deletes the vendor with primary key pk on POST request.

POST /vendor/<str:object_id>/delete

Deletes the vendor represented by the primary key object_id.

Example request:

```
POST /vendor/5/delete HTTP/1.1
Host: localhost:8000
Content-Type: application/x-www-form-urlencoded
```

Parameters

- **object_id** – Vendor primary key.

Response Headers

- **Content-Type** – application/x-www-form-urlencoded

Status Codes

- **302 Found** – Redirects to /vendors.
- **500 Internal Server Error** – Vendor matching query does not exist.

`InvManage.views.display_vendors_view(request)`

Retrieves the list of vendors on GET request.

GET `/vendors/`

Gets the list of all vendors.

Example request:

```
GET /vendors/ HTTP/1.1
Host: localhost:8000
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/
webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
```

Form Parameters

- **page** – The page number of the vendors list.

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: text/html; charset=utf-8
```

Request Headers

- **Accept** – text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9

Status Codes

- **200 OK** – List of vendors received successfully.

`InvManage.views.get_vendor(request)`

Returns the JSON serialized data of the requested vendor on GET request.

GET `/get_vendor/`

Gets the JSON serialized data of the requested vendor.

Example request:

```
GET /get_vendor/ HTTP/1.1
Host: localhost:8000
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/
webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
```

Parameters

- **vendor_id** – Vendor primary key.

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json; charset=utf-8

[
  {
```

(continues on next page)

(continued from previous page)

```

    "name": "Lug Vendor",
    "identifier": "TBPN-02692",
    "gstn": "89AAC4683897343",
    "address": {
        "name": "Kuame Burns",
        "address": "Nonummy Avenue",
        "city": "Chung Cheong",
        "phone": "679 166-3127",
        "state": "Guanacaste",
        "country": "tellusidnunc.net",
        "post": "8949"
    }
}
]

```

Response Headers

- **Content-Type** – application/json

Status Codes

- **200 OK** – List of vendors received successfully.
- **400 Bad Request** – Bad request version
- **500 Internal Server Error** – Vendor matching query does not exist.

class InvManage.views.ProductBasicInfoForm

Bases: django.forms.Form

Form for basic product information.

name

Name of the product.

Type str**item_type**

Type of the product.

Type str**category**

Product category.

Type str**description**

Short description of the product.

Type str**prefix** = basic**context****name****item_type****category**

description**class** InvManage.views.ProductDetailedInfoForm

Bases: django.forms.Form

Form for product physical details

length

Length of the product.

Type str**width**

Width of the product.

Type str**height**

Height of the product.

Type str**weight**

Weight of the product.

Type str**prefix** = detailed**context****length****width****height****weight****class** InvManage.views.ProductPricingForm

Bases: django.forms.Form

Form for pricing information of the product.

price

Price of the product.

Type float**discount**

Default discount percentage on the product.

Type float**prefix** = pricing**context****price****discount****class** InvManage.views.ProductPurchaseEntryForm

Bases: django.forms.Form

Form for product purchase entry.

ppe_id

Unique identifier of the product purchase entry.

Type int

product

Product associated wit the product purchase entry.

Type *Product*

quantity

Quantity of the product to be ordered.

Type int

price

Price of the product.

Type float

discount

Percentage discount on the product purchase.

Type float

prefix = form**context****ppe_id****product****quantity****price****discount****class** InvManage.views.ProductSalesEntryForm

Bases: django.forms.Form

Form for product sales entry.

pse_id

Unique identifier of the product sales entry.

Type int

product

Product associated with the product sales entry.

Type *Product*

quantity

Quantity of the product.

Type int

price

Price of the product.

Type float

discount

Percentage discount on the product.

Type float

```
    prefix = form
    context
    pse_id
    product
    quantity
    price
    discount

class InvManage.views.ProductStatusForm
    Bases: django.forms.Form

    Form for locating and identifying the product.

    quantity
        Stock quantity of the product.

        Type int

    identifier
        Unique identifier of the product.

        Type str

    location
        Physical location of the product.

        Type str

    context
    prefix = status
    quantity
    identifier
    location

class InvManage.views.ProductStorageInfoForm
    Bases: django.forms.Form

    Form for storage information.

    barcode
        Barcode of the product.

        Type str

    expiry
        Expiry date of the product.

        Type DateField

    prefix = storage
    barcode
    context
    expiry
```

```

class InvManage.views.PurchaseOrderBasicInfo
    Bases: django.forms.Form

    Form for basic information of the purchase order.

    vendor
        Vendor associated with the purchahse order.

        Type Vendor

    date
        Date of the purchase order creation.

        Type DateField

    po
        Purchase order number.

        Type int

    discount
        Percentage of overall discount.

        Type float

    tax
        Percentage of tax.

        Type float

    paid
        Amount paid to the vendor.

        Type float

    balance
        Balance amount to be paid to the vendor.

        Type float

    subtotal
        Total of all the product purchase entries associated with the purchase order.

        Type float

    taxtotal
        Total tax applicable on the subtotal.

        Type float

    ordertotal
        Total price of the purchase order including taxtotal.

        Type float

    prefix = po
    context
    product_choices
    vendor_choices
    vendor
    date
    po

```

discount

tax

paid

balance

subtotal

taxtotal

ordertotal

class `InvManage.views.GRNInfo`

Bases: `django.forms.Form`

Form for goods receipt note (GRN).

vendor

Vendor associated with the goods receipt note.

Type *Vendor*

poRef

List of identifiers of the purchase orders from which the goods receipt note is derived.

Type *PurchaseOrder*

identifier

Unique identifier of the goods receipt note.

Type `str`

date

Date of GRN creation.

Type `DateField`

grnType

Type of GRN (auto or manual).

Type `str`

amendNumber

Amendment number of the GRN.

Type `int`

amendDate

Amendment date.

Type `DateField`

transporter

Name of the transport/shipping service.

Type `str`

vehicleNumber

Vehicle number using which the products are shipped.

Type `str`

gateInwardNumber

Gate inward number of the vehicle.

Type `str`

preparedBy

Name/identifier of the person who created the goods receipt note.

Type str

checkedBy

Name/identifier of the person who validated the goods receipt note.

Type str

checkedBy

Name/identifier of the person who inspected the physical products in the goods receipt note.

Type str

approvedBy

Name/identifier of the authority who approved the goods receipt note.

Type str

prefix = grn

context

product_choices

vendor_choices

vendor

TYPE_CHOICES = [['manual', 'Blank'], ['auto', 'PO Reference']]

poRef

identifier

date

grnType

amendNumber

amendDate

transporter

vehicleNumber

gateInwardNumber

preparedBy

checkedBy

inspectedBy

approvedBy

class InvManage.views.GRNEEntryForm

Bases: django.forms.Form

Form for goods receipt note entry (GRNE).

product

Product associated with the goods receipt note entry.

Type *Product*

grne_id

Unique identifier of the GRNE.

Type int

ppe_id

Unique identifier of the product purchase entry associated with the GRNE.

Type int

quantity

Ordered quantity of product with reference to product purchase entry.

Type int

remark

Remarks of the quality engineer or the GRN creator about status of products received.

Type str

receivedQty

Quantity of product received against the ordered quantity.

Type str

acceptedQty

Quantity of product accepted as OK.

Type str

rejectedQty

Quantity of product rejected (not OK, on HOLD, extra delivery, etc.)

Type str

prefix = form**context****product****grne_id****ppe_id****quantity****remark****receivedQty****acceptedQty****rejectedQty****class** InvManage.views.**ThumbnailForm**

Bases: django.forms.Form

For for product thumbnail image.

image

Image of the product.

Type ImageField

prefix = thumbnail**image**

```

class InvManage.views.SalesOrderBasicInfo
    Bases: django.forms.Form

    Form for basic information on sales order.

    consumer
        Consumer associated with the sales order.

        Type Consumer

    date
        Date of the sales order creation.

        Type DateField

    so
        Sales order number.

        Type int

    discount
        Percentage discount associated with the sales order.

        Type float

    tax
        Percentage of tax applicable.

        Type float

    paid
        Amount received from the consumer.

        Type float

    balance
        Amount balance with the consumer.

        Type float

    subtotal
        Total of all the product sales entries associated with the sales order.

        Type float

    taxtotal
        Total tax applicable on the subtotal.

        Type float

    ordertotal
        Total price of the sales order including taxtotal.

        Type float

    prefix = so

    context
    product_choices
    consumer_choices
    consumer
    date
    so

```

discount
tax
paid
balance
subtotal
taxtotal
ordertotal

class InvManage.views.ShippingAddressForm

Bases: django.forms.Form

For for shipping address.

title

Title of the firm.

Type str

name

Name of the firm.

Type str

phone

Contact number of the firm.

Type str

address

Postal address.

Type str

city

City.

Type str

state

State.

Type str

country

Country.

Type str

website

Official website of the firm.

Type URLField

post

Postal code of the firm.

Type str

prefix = ship

context

```

    title
    name
    phone
    address
    city
    state
    country
    website
    post

class InvManage.views.VendorForm
    Bases: django.forms.Form

    Form for basic informaion of the vendor.

    name
        Name of the vendor.

        Type str

    identifier
        Unique identifier of the vendor.

        Type str

    gstin
        GSTIN number of the vendor.

        Type str

    prefix = vend

    context

    name

    identifier

    gstin

class InvManage.views.ConsumerForm
    Bases: django.forms.Form

    Form for consumer.

    name
        Name of the consumer.

        Type str

    identifier
        Unique identifier of the consumer.

        Type str

    gstin
        GSTIN number of the consumer.

        Type str

```

phone

Contact number.

Type str**address**

Address of the consumer.

Type str**email**

E-mail address of the consumer.

Type str**location**

City of the consumer.

Type str**prefix = consumer****context****name****identifier****gstn****phone****address****email****location****class** InvManage.views.CompanyForm

Bases: django.forms.Form

Form for company.

name

Name of the company.

Type str**owner**

Name of the owner of the company.

Type str**gstn**

GSTIN number of the company.

Type str**phone**

Contact number of the company.

Type str**address**

Postal address.

Type str

```

email
    E-mail address of the contact person.
        Type str
location
    City of the company.
        Type str
prefix = comp
context
name
owner
gstn
phone
address
email
location
class InvManage.views.HistoryForm
    Bases: django.forms.Form
    Form for history view.
qlen
    Count of the events visible on the history view.
        Type int
prefix = history
context
qlen
class InvManage.views.CommunicationForm
    Bases: django.forms.Form
    Form for communication information.
language
    Language of communication.
        Type str
phone
    Contact number of the firm.
        Type str
email
    E-mail of the contact person.
        Type EmailField
fax
    Fax number.
        Type str

```

```
    prefix = com
    context
    language
    phone
    email
    fax

class InvManage.views.PurchaseDataForm
    Bases: django.forms.Form

    Form for purchase related data.

    currency
        Currency of purchase.

        Type str

    minorder
        Minimum order quantity.

        Type int

    contactperson
        Name of the contact person.

        Type str

    refcode
        Reference code.

        Type str

    transportmode
        Mode of transport of the shipment.

        Type str

    prefix = pdform
    context
    currency
    minorder
    contactperson
    refcode
    transportmode

class InvManage.views.BankAccountForm
    Bases: django.forms.Form

    Form for bank account details.

    name
        Name of the bank.

        Type str

    branch
        Branch number of the bank.
```

```

        Type str
region
    City in which the branch is located.
        Type str
route
    Transit number.
        Type str
number
    Bank account number.
        Type int
acctype
    Type of bank account.
        Type str
iban
    IBAN number.
        Type str
code
    Bank code.
        Type str
branchcode
    Branch code.
        Type str
prefix = bank
context
name
branch
region
route
number
acctype
iban
code
branchcode
class InvManage.views.Product
    Bases: django.db.models.Model
    Model of the product.
    name
        Name of the product.
        Type str

```

category

Product category.

Type str**item_type**

Type of the product.

Type str**description**

Short description of the product.

Type str**price**

Price of the product.

Type float**quantity**

Stock quantity of the product.

Type int**identifier**

Unique identifier of the product.

Type str**location**

Physical location of the product.

Type str**length**

Length of the product.

Type str**width**

Width of the product.

Type str**height**

Height of the product.

Type str**weight**

Weight of the product.

Type str**discount**

Default discount percentage on the product.

Type float**barcode**

Barcode of the product.

Type str**expiry**

Expiry date of the product.


```

        Type DateField

image
    Image of the product.
        Type ImageField

name
category
item_type
description
price
quantity
identifier
location
length
width
height
weight
discount
barcode
expiry
image
    __str__(self)

class InvManage.views.ProductPurchaseEntry
    Bases: django.db.models.Model

    Model of the product purchase entry.

    product
        Product associated wit the product purchase entry.
        Type ModelChoiceField

    quantity
        Quantity of the product to be ordered.
        Type int

    price
        Price of the product.
        Type float

    discount
        Percentage discount on the product purchase.
        Type float

    order
        Referenced purchase order.

```

Type *PurchaseOrder*

receivedQty

Quantity of product received against the ordered quantity.

Type str

acceptedQty

Quantity of product accepted as OK.

Type str

rejectedQty

Quantity of product rejected (not OK, on HOLD, extra delivery, etc.)

Type str

is_complete()

Returns the completion status (boolean).

pending_quantity()

Returns the pending quantity against the entry (int).

product

quantity

price

discount

order

receivedQty

acceptedQty

rejectedQty

is_complete(self)

property pending_quantity(self)

class InvManage.views.GRNEEntry

Bases: django.db.models.Model

Model of the goods receipt note entry (GRNE).

product

Primary key of the Product instance associated with the goods receipt note entry.

Type *Product*

quantity

Ordered quantity of product with reference to product purchase entry.

Type int

grne

Primary key of the GoodsReceiptNote instance.

Type int

ppe

Primary key of the ProductPurchaseEntry instance associated with the GRNE.

Type int

remark

Remarks of the quality engineer or the GRN creator about status of products received.

Type str

receivedQty

Quantity of product received against the ordered quantity.

Type str

acceptedQty

Quantity of product accepted as OK.

Type str

rejectedQty

Quantity of product rejected (not OK, on HOLD, extra delivery, etc.)

Type str

product**quantity****grn****ppes****remark****receivedQty****acceptedQty****rejectedQty****class** InvManage.views.ProductSalesEntry

Bases: django.db.models.Model

Model of the product sales entry.

product

Primary key of the Product associated with the product sales entry.

Type *Product*

quantity

Quantity of the product.

Type int

price

Price of the product.

Type float

discount

Percentage discount on the product.

Type float

order

Primary key of the SalesOrder referenced by the entry.

Type *SalesOrder*

product

quantity

price

discount

order

class `InvManage.views.SalesOrder`

Bases: `django.db.models.Model`

Model of the sales order.

consumer

Consumer associated with the sales order.

Type `ModelChoiceField`

date

Date of the sales order creation.

Type `DateField`

so

Sales order number.

Type `int`

discount

Percentage discount associated with the sales order.

Type `float`

tax

Percentage of tax applicable.

Type `float`

paid

Amount received from the consumer.

Type `float`

balance

Amount balance with the consumer.

Type `float`

subtotal

Total of all the product sales entries associated with the sales order.

Type `float`

taxtotal

Total tax applicable on the *subtotal*.

Type `float`

ordertotal

Total price of the sales order including *taxtotal*.

Type `float`

consumer

date

so

discount**tax****paid****balance****subtotal****taxtotal****ordertotal****class** InvManage.views.**PurchaseOrder**

Bases: django.db.models.Model

Model of the purchase order.

vendor

Vendor associated with the purchase order.

Type *Vendor***date**

Date and time of the purchase order creation.

Type DateField**po**

Purchase order number.

Type int**discount**

Percentage discount.

Type float**tax**

Percentage of tax applicable for the purchase.

Type float**paid**

Amount paid against the PO.

Type float**balance**

Amount balance which remains to be paid.

Type float**subtotal**

Total of all the product purchase entries associated with the purchase order.

Type float**taxtotal**Total tax applicable on the *subtotal*.**Type** float**ordertotal**Total price of the purchase order including *taxtotal*.**Type** float

is_complete()

Returns the completion status (boolean).

pending_ppes()

Returns list of product purchase entries which are not completed.

vendor

date

po

discount

tax

paid

balance

subtotal

taxtotal

ordertotal

is_complete(self)

pending_ppes(self)

class InvManage.views.GoodsReceiptNote

Bases: django.db.models.Model

Model of the goods receipt note (GRN).

vendor

Vendor associated with the goods receipt note.

Type ModelChoiceField

poRef

List of identifiers of the purchase orders from which the goods receipt note is derived.

Type MultipleChoiceField

identifier

Unique identifier of the goods receipt note.

Type str

date

Date of GRN creation.

Type DateField

grnType

Type of GRN (auto or manual).

Type ChoiceField

amendNumber

Amendment number of the GRN.

Type int

amendDate

Amendment date.

Type DateField

transporter

Name of the transport/shipping service.

Type str

vehicleNumber

Vehicle number using which the products are shipped.

Type str

gateInwardNumber

Gate inward number of the vehicle.

Type str

preparedBy

Name/identifier of the person who created the goods receipt note.

Type str

checkedBy

Name/identifier of the person who validated the goods receipt note.

Type str

checkedBy

Name/identifier of the person who inspected the physical products in the goods receipt note.

Type str

approvedBy

Name/identifier of the authority who approved the goods receipt note.

Type str

TYPE_CHOICES = [['manual', 'Blank'], ['auto', 'PO Reference']]

vendor

poRef

identifier

date

grnType

amendNumber

amendDate

transporter

vehicleNumber

gateInwardNumber

preparedBy

checkedBy

inspectedBy

approvedBy

```
class InvManage.views.Consumer
    Bases: django.db.models.Model

    Model of the consumer.

    name
        Name of the consumer.

        Type str

    identifier
        Unique identifier of the consumer.

        Type str

    gstn
        GSTIN number of the consumer.

        Type str

    phone
        Contact number.

        Type str

    address
        Address of the consumer.

        Type str

    email
        E-mail address of the consumer.

        Type str

    location
        City of the consumer.

        Type str

    Returns [description]
    Return type [type]

    name
    identifier
    gstn
    phone
    address
    email
    location
    __str__ (self)

class InvManage.views.Vendor
    Bases: django.db.models.Model

    Model of the vendor.

    name
        Name of the vendor.
```



```

        Type str
identifier
    Unique identifier of the vendor.
        Type str
gstn
    GSTIN number of the vendor.
        Type str
address
    Primary key of the ShippingAddress instance associated with the vendor.
        Type ShippingAddress
communication
    Primary key of the Communication instance associated with the vendor.
        Type Communication
backaccount
    Primary key of the BankAccount instance associated with the vendor.
        Type BankAccount
purchasedata
    Primary key of the PurchaseData instance associated with the vendor.
        Type PurchaseData

Returns String representation of the vendor name.
Return type str

name
identifier
gstn
address
communication
bankaccount
purchasedata
__str__(self)
class InvManage.views.Company
    Bases: django.db.models.Model
    Model of the company.
    name
        Name of the company.
        Type str
    owner
        Owner of the company.
        Type str

```

gstn

GSTIN number of the company.

Type str**phone**

Contact number.

Type str**address**

Postal address.

Type str**email**

E-mail address of the contact person.

Type str**location**

City of the company.

Type str**image**

Photo/logo of the company.

Type ImageField**shippingaddress**Primary key of the `ShippingAddress` instance associated with the company.**Type** *ShippingAddress***Returns** String representation of the company name.**Return type** str**name****owner****gstn****phone****address****email****location****image****shippingaddress****__str__** (*self*)**class** `InvManage.views.PurchaseInvoice`Bases: `django.db.models.Model`

Model for the purchase invoice.

company

Primary key of the Company.

Type *Company*

po

Primary key of the PurchaseOrder.

Type *PurchaseOrder*

shippingaddress

Primary key of the ShippingAddress.

Type *ShippingAddress*

communication

Primary key of the Communication.

Type *Communication*

company

po

shippingaddress

communication

class `InvManage.views.SalesInvoice`

Bases: `django.db.models.Model`

Model for the sales order invoice.

company

Primary key of the Company.

Type *Company*

so

Primary key of the SalesOrder.

Type *SalesOrder*

shippingaddress

Primary key of the ShippingAddress.

Type *ShippingAddress*

company

so

shippingaddress

class `InvManage.views.FilterState`

Bases: `django.db.models.Model`

Model of the filter state.

name

Name of the filter state.

Type `str`

name

class `InvManage.views.FilterColumn`

Bases: `django.db.models.Model`

Model of the column state.

name

Name of the column.

Type str**label**

Label of the column.

Type str**visible**

Visibility flag.

Type bool**position**

Index of the column.

Type int**state**Primary key of the `FilterState` instance being referenced by the column instance.**Type** *FilterState***name****label****visible****position****state****class** `InvManage.views.Object`Bases: `django.db.models.Model`

Model for a generic object.

identifier

Unique identifier of the object.

Type str**name**

Name of the object.

Type str**model**

Model associated with the object.

Type str**identifier****name****model****class** `InvManage.views.EventCard`Bases: `django.db.models.Model`

Model for event card to be displayed in the history view.

obj

Object associated with the event.

```

        Type OneToOneField
objId
    Unique identifier of the object.
        Type str
objname
    Name of the object.
        Type str
objmodel
    Model of the object.
        Type str
date
    Timestamp of the event.
        Type DateField
operation
    Type of event.
        Type str
obj
objId
objname
objmodel
date
operation
class InvManage.views.EventType
    Bases: django.db.models.Model
    Model for event type.
name
    Name of the event being triggered.
        Type str
label
    Label associated with the event.
        Type str

    Returns Returns the string representation of the label when the event type is queried.
    Return type Label

name
label
__str__(self)

```

class `InvManage.views.ObjectModel`

Bases: `django.db.models.Model`

Model associated with an object.

name

Name of the object.

Type `str`

label

Label string of the object.

Type `str`

modName

Name of the model associated with the object.

Type `str`

Returns Returns string representation of the label when object model is queried.

Return type `Label`

name

label

modName

__str__ (*self*)

class `InvManage.views.HistoryFilterState`

Bases: `django.db.models.Model`

Model of filter state of the history.

params

Paramameters of the configured filters.

Type `str`

params

class `InvManage.views.ShippingAddress`

Bases: `django.db.models.Model`

Model of the shipping address.

title

Title of the firm.

Type `str`

name

Name of the firm.

Type `str`

phone

Contact number of the firm.

Type `str`

address

Postal address.

```

        Type str
city
    City.
        Type str
state
    State.
        Type str
country
    Country.
        Type str
website
    Official website of the firm.
        Type URLField
post
    Postal code of the firm.
        Type str
title
name
phone
address
city
state
country
website
post
class InvManage.views.PurchaseData
    Bases: django.db.models.Model
    Model of the purchase data.
    currency
        Currency of purchase.
        Type str
    minorder
        Minimum order quantity.
        Type int
    contactperson
        Name of the contact person.
        Type str
    refcode
        Reference code.

```

Type str

transportmode

Mode of transport of the shipment.

Type str

currency

minorder

contactperson

refcode

transportmode

class InvManage.views.Communication

Bases: django.db.models.Model

Model of the communication information.

language

Language of communication.

Type str

phone

Contact number of the firm.

Type str

email

E-mail of the contact person.

Type EmailField

fax

Fax number.

Type str

language

phone

email

fax

class InvManage.views.BankAccount

Bases: django.db.models.Model

Model of the bank account details.

name

Name of the bank.

Type str

branch

Branch number of the bank.

Type str

region

City in which the branch is located.

Type str
route
 Transit number.
Type str
number
 Bank account number.
Type int
acctype
 Type of bank account.
Type str
iban
 IBAN number.
Type str
code
 Bank code.
Type str
branchcode
 Branch code.
Type str
name
branch
region
route
number
acctype
iban
code
branchcode

InvManage.views.get_columns(*state*)
 Gets the columns of the display table.

This function takes a filter state and returns an array of column names that are ordered depending on the position attribute of the columns. This order is important as the queryset will be rendered in the display table according to the column order in this returned array.

Parameters **state** ([FilterState](#)) – Filter state of the columns.

Returns A list that dictates the order of the columns.

Return type List

InvManage.views.sort_ascending_descending(*request, model*)
 Sorts and saves the state of the column.

This function takes a request, finds which column needs to be sorted in ascending/descending order, and returns the sorted queryset.

Parameters

- **request** (*HttpRequest*) – GET request received for retrieving the list of model instances.
- **model** (*Model*) – Model of the objects being retrieved.

Returns List of sorted object instances.

Return type *QuerySet*

`InvManage.views.change_column_position(request, state)`

Modifies the column states depending on the request.

This function:

- Takes a filter state
- Extracts the column names into an order array
- Modifies the column order in the array depending on user input direction (left/right)
- Saves this modified state of columns into the database
- Returns the modified column names array

Parameters

- **request** (*HttpRequest*) – GET request for retrieving the state of the columns.
- **state** (*FilterState*) – Filter state of the columns.

Returns List of sorted columns.

Return type *List*

`InvManage.views.paginate(queryset, filter, page_number)`

Paginates the list of objects.

This function takes the entire queryset and filters out only objects belonging to the request page

Parameters

- **queryset** (*QuerySet*) – *QuerySet* of the objects retrieved from the database.
- **filter** (*FilterState*) – Filter state of the columns.
- **page_number** (*int*) – Page number requested.

`class InvManage.views.ConsumerFilter`

Bases: *django_filters.FilterSet*

`class Meta`

`model`

`fields`

`context`

`name`

`identifier`

`phone`

`email`

`location`

```
order_by_field = ordering
```

```
ordering
```

```
class InvManage.views.ConsumerSerializer
```

```
Bases: rest_framework.serializers.ModelSerializer
```

```
Serializer for Consumer class instance.
```

The `ConsumerSerializer.data` attribute gives the JSON serialized data of the `Consumer` instance:

```
{
  "name": "The Music Store",
  "identifier": "CONS1256",
  "gsttin": "89AAC4633353643",
  "phone": "+91 8325642358",
  "address": "Plot no 958, N- 4, Neo Complex, Barh, Wokha, Nagaland, 797111",
  "email": "JohnDoe@themusic.store",
  "location": "Wokha"
}
```

```
class Meta
```

```
model
```

```
fields = ['name', 'identifier', 'gsttin', 'phone', 'address', 'email',
          'location']
```

```
to_representation(self, instance)
```

```
InvManage.views.create_consumer_view(request)
```

Creates a consumer on POST request, and returns a consumer creation form on GET request.

```
GET /consumer
```

Gets the consumer creation form.

Example request:

```
GET /consumer/ HTTP/1.1
Host: localhost:8000
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/
webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: text/html; charset=utf-8
```

Request Headers

- **Accept** – text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9

Status Codes

- **200 OK** – Consumer creation form received successfully.

POST /consumer

Creates a consumer.

Example request:

```
POST /consumer/ HTTP/1.1
Host: localhost:8000
Content-Type: multipart/form-data;
```

Form Parameters

- **consumer-name** – The Music Store
- **consumer-identifier** – CONS1256
- **consumer-gstin** – 89ACC654654335
- **consumer-phone** – +91 6543525422
- **consumer-address** – Plot no. 958, N-4, Neo Complex, Barh, Wokha, Nagaland, 797111
- **consumer-email** – johnDoe@themusic.store
- **consumer-location** – Wokha

Response Headers

- **Content-Type** – multipart/form-data; boundary=—WebKitFormBoundaryLTR88aZAnBUSE7mv

Status Codes

- **302 Found** – Redirects to /consumer.

`InvManage.views.update_consumer_view(request)`

Updates a consumer and returns the consumer update form for GET request.

GET /consumer/update

Gets the consumer update form whose primary key matches the query parameter pk.

Example request:

```
GET /consumer/update HTTP/1.1
Host: localhost:8000
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
```

Query Parameters

- **pk** – The primary key of the consumer.

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: text/html; charset=utf-8
```

Request Headers

- **Accept** – text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9

Status Codes

- **200 OK** – Consumer update form received successfully.

POST /consumer/update

Updates the consumer.

Example request:

```
POST /company/update HTTP/1.1
Host: localhost:8000
Content-Type: multipart/form-data;
```

Form Parameters

- **pk** – 7
- **consumer-name** – The Music Store
- **consumer-identifier** – CONS1256
- **consumer-gstin** – 89ACC654654335
- **consumer-phone** – +91 6543525422
- **consumer-address** – Plot no. 958, N-4, Neo Complex, Barh, Wokha, Nagaland, 797111
- **consumer-email** – johnDoe@themusic.store
- **consumer-location** – Wokha

Response Headers

- **Content-Type** – multipart/form-data; boundary=—WebKitFormBoundaryLTR88aZAnBUSE7mv

Status Codes

- **302 Found** – Redirects to /consumer.

`InvManage.views.delete_consumer_view(request, pk)`

Deletes the consumer with primary key `pk` on POST request.

POST /consumer/<str:object_id>/delete

Deletes the consumer represented by the primary key `object_id`.

Example request:

```
POST /consumer/5/delete HTTP/1.1
Host: localhost:8000
Content-Type: application/x-www-form-urlencoded
```

Parameters

- **object_id** – Consumer primary key.

Response Headers

- **Content-Type** – application/x-www-form-urlencoded

Status Codes

- **302 Found** – Redirects to /consumer.

- 500 Internal Server Error – Consumer matching query does not exist.

`InvManage.views.display_consumers_view(request)`

Retrieves the list of consumers on GET request.

GET `/consumers/`

Gets the list of all consumers.

Example request:

```
GET /companies/ HTTP/1.1
Host: localhost:8000
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
```

Form Parameters

- **page** – The page number of the consumers list.

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: text/html; charset=utf-8
```

Request Headers

- **Accept** – text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9

Status Codes

- **200 OK** – List of consumers received successfully.

`InvManage.views.get_consumer(request)`

Returns the JSON serialized data of the requested consumer on GET request.

GET `/get_consumer/`

Gets the JSON serialized data of the requested consumer.

Example request:

```
GET /get_consumer/ HTTP/1.1
Host: localhost:8000
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json; charset=utf-8

[
  {
    "name": "The Music Store",
    "identifier": "CONS1256",
```

(continues on next page)

(continued from previous page)

```

        "gsttin": "89AAC4633353643",
        "phone": "+91 8325642358",
        "address": "Plot no 958, N- 4, Neo Complex, Barh, Wokha, Nagaland, 797111",
        "email": "JohnDoe@themusic.store",
        "location": "Wokha"
    }
]

```

Response Headers

- **Content-Type** – application/json

Status Codes

- **200 OK** – List of consumers received successfully.
- **400 Bad Request** – Bad request version
- **500 Internal Server Error** – Consumer matching query does not exist.

class InvManage.views.ProductBasicInfoForm

Bases: django.forms.Form

Form for basic product information.

name

Name of the product.

Type str**item_type**

Type of the product.

Type str**category**

Product category.

Type str**description**

Short description of the product.

Type str**prefix** = basic**context****name****item_type****category****description****class InvManage.views.ProductDetailedInfoForm**

Bases: django.forms.Form

Form for product physical details

length

Length of the product.

Type str**width**

Width of the product.

Type str**height**

Height of the product.

Type str**weight**

Weight of the product.

Type str**prefix = detailed****context****length****width****height****weight****class** InvManage.views.ProductPricingForm

Bases: django.forms.Form

Form for pricing information of the product.

price

Price of the product.

Type float**discount**

Default discount percentage on the product.

Type float**prefix = pricing****context****price****discount****class** InvManage.views.ProductPurchaseEntryForm

Bases: django.forms.Form

Form for product purchase entry.

ppe_id

Unique identifier of the product purchase entry.

Type int**product**

Product associated with the product purchase entry.

Type *Product*

quantity

Quantity of the product to be ordered.

Type int

price

Price of the product.

Type float

discount

Percentage discount on the product purchase.

Type float

prefix = form

context

ppe_id

product

quantity

price

discount

class InvManage.views.ProductSalesEntryForm

Bases: django.forms.Form

Form for product sales entry.

pse_id

Unique identifier of the product sales entry.

Type int

product

Product associated with the product sales entry.

Type *Product*

quantity

Quantity of the product.

Type int

price

Price of the product.

Type float

discount

Percentage discount on the product.

Type float

prefix = form

context

pse_id

product

quantity

price

discount

class InvManage.views.ProductStatusForm

Bases: django.forms.Form

Form for locating and identifying the product.

quantity

Stock quantity of the product.

Type int

identifier

Unique identifier of the product.

Type str

location

Physical location of the product.

Type str

context

prefix = status

quantity

identifier

location

class InvManage.views.ProductStorageInfoForm

Bases: django.forms.Form

Form for storage information.

barcode

Barcode of the product.

Type str

expiry

Expiry date of the product.

Type DateField

prefix = storage

barcode

context

expiry

class InvManage.views.PurchaseOrderBasicInfo

Bases: django.forms.Form

Form for basic information of the purchase order.

vendor

Vendor associated with the purchahse order.

Type *Vendor*

date
Date of the purchase order creation.
Type DateField

po
Purchase order number.
Type int

discount
Percentage of overall discount.
Type float

tax
Percentage of tax.
Type float

paid
Amount paid to the vendor.
Type float

balance
Balance amount to be paid to the vendor.
Type float

subtotal
Total of all the product purchase entries associated with the purchase order.
Type float

taxtotal
Total tax applicable on the *subtotal*.
Type float

ordertotal
Total price of the purchase order including *taxtotal*.
Type float

prefix = po

context

product_choices

vendor_choices

vendor

date

po

discount

tax

paid

balance

subtotal

taxtotal

ordertotal

class `InvManage.views.GRNInfo`

Bases: `django.forms.Form`

Form for goods receipt note (GRN).

vendor

Vendor associated with the goods receipt note.

Type *Vendor*

poRef

List of identifiers of the purchase orders from which the goods receipt note is derived.

Type *PurchaseOrder*

identifier

Unique identifier of the goods receipt note.

Type `str`

date

Date of GRN creation.

Type `DateField`

grnType

Type of GRN (auto or manual).

Type `str`

amendNumber

Amendment number of the GRN.

Type `int`

amendDate

Amendment date.

Type `DateField`

transporter

Name of the transport/shipping service.

Type `str`

vehicleNumber

Vehicle number using which the products are shipped.

Type `str`

gateInwardNumber

Gate inward number of the vehicle.

Type `str`

preparedBy

Name/identifier of the person who created the goods receipt note.

Type `str`

checkedBy

Name/identifier of the person who validated the goods receipt note.

```

        Type str
    checkedBy
        Name/identifier of the person who inspected the physical products in the goods receipt note.
        Type str
    approvedBy
        Name/identifier of the authority who approved the goods receipt note.
        Type str
    prefix = grn
    context
    product_choices
    vendor_choices
    vendor
    TYPE_CHOICES = [['manual', 'Blank'], ['auto', 'PO Reference']]
    poRef
    identifier
    date
    grnType
    amendNumber
    amendDate
    transporter
    vehicleNumber
    gateInwardNumber
    preparedBy
    checkedBy
    inspectedBy
    approvedBy
class InvManage.views.GRNEEntryForm
    Bases: django.forms.Form
    Form for goods receipt note entry (GRNE).
    product
        Product associated with the goods receipt note entry.
        Type Product
    grne_id
        Unique identifier of the GRNE.
        Type int
    ppe_id
        Unique identifier of the product purchase entry associated with the GRNE.
        Type int

```

quantity

Ordered quantity of product with reference to product purchase entry.

Type int

remark

Remarks of the quality engineer or the GRN creator about status of products received.

Type str

receivedQty

Quantity of product received against the ordered quantity.

Type str

acceptedQty

Quantity of product accepted as OK.

Type str

rejectedQty

Quantity of product rejected (not OK, on HOLD, extra delivery, etc.)

Type str

prefix = form

context

product

grne_id

ppe_id

quantity

remark

receivedQty

acceptedQty

rejectedQty

class InvManage.views.ThumbnailForm

Bases: django.forms.Form

For for product thumbnail image.

image

Image of the product.

Type ImageField

prefix = thumbnail

image

class InvManage.views.SalesOrderBasicInfo

Bases: django.forms.Form

Form for basic information on sales order.

consumer

Consumer associated with the sales order.

Type *Consumer*

date
Date of the sales order creation.
Type DateField

so
Sales order number.
Type int

discount
Percentage discount associated with the sales order.
Type float

tax
Percentage of tax applicable.
Type float

paid
Amount received from the consumer.
Type float

balance
Amount balance with the consumer.
Type float

subtotal
Total of all the product sales entries associated with the sales order.
Type float

taxtotal
Total tax applicable on the *subtotal*.
Type float

ordertotal
Total price of the sales order including *taxtotal*.
Type float

prefix = so

context

product_choices

consumer_choices

consumer

date

so

discount

tax

paid

balance

subtotal

taxtotal

ordertotal

class InvManage.views.ShippingAddressForm

Bases: django.forms.Form

For for shipping address.

title

Title of the firm.

Type str

name

Name of the firm.

Type str

phone

Contact number of the firm.

Type str

address

Postal address.

Type str

city

City.

Type str

state

State.

Type str

country

Country.

Type str

website

Official website of the firm.

Type URLField

post

Postal code of the firm.

Type str

prefix = ship

context

title

name

phone

address

city


```

    state
    country
    website
    post
class InvManage.views.VendorForm
    Bases: django.forms.Form
    Form for basic informaion of the vendor.
    name
        Name of the vendor.
        Type str
    identifier
        Unique identifier of the vendor.
        Type str
    gstin
        GSTIN number of the vendor.
        Type str
    prefix = vend
    context
    name
    identifier
    gstin
class InvManage.views.ConsumerForm
    Bases: django.forms.Form
    Form for consumer.
    name
        Name of the consumer.
        Type str
    identifier
        Unique identifier of the consumer.
        Type str
    gstin
        GSTIN number of the consumer.
        Type str
    phone
        Contact number.
        Type str
    address
        Address of the consumer.
        Type str

```

email

E-mail address of the consumer.

Type str**location**

City of the consumer.

Type str**prefix = consumer****context****name****identifier****gstn****phone****address****email****location****class** InvManage.views.CompanyForm

Bases: django.forms.Form

Form for company.

name

Name of the company.

Type str**owner**

Name of the owner of the company.

Type str**gstn**

GSTIN number of the company.

Type str**phone**

Contact number of the company.

Type str**address**

Postal address.

Type str**email**

E-mail address of the contact person.

Type str**location**

City of the company.

Type str

```

    prefix = comp
    context
    name
    owner
    gstin
    phone
    address
    email
    location

class InvManage.views.HistoryForm
    Bases: django.forms.Form

    Form for history view.

    qlen
        Count of the events visible on the history view.

        Type int

    prefix = history
    context
    qlen

class InvManage.views.CommunicationForm
    Bases: django.forms.Form

    Form for communication information.

    language
        Language of communication.

        Type str

    phone
        Contact number of the firm.

        Type str

    email
        E-mail of the contact person.

        Type EmailField

    fax
        Fax number.

        Type str

    prefix = com
    context
    language
    phone
    email

```

fax

class InvManage.views.PurchaseDataForm

Bases: django.forms.Form

Form for purchase related data.

currency

Currency of purchase.

Type str

minorder

Minimum order quantity.

Type int

contactperson

Name of the contact person.

Type str

refcode

Reference code.

Type str

transportmode

Mode of transport of the shipment.

Type str

prefix = pdform

context

currency

minorder

contactperson

refcode

transportmode

class InvManage.views.BankAccountForm

Bases: django.forms.Form

Form for bank account details.

name

Name of the bank.

Type str

branch

Branch number of the bank.

Type str

region

City in which the branch is located.

Type str

route

Transit number.

```

        Type str
number
    Bank account number.
        Type int
acctype
    Type of bank account.
        Type str
iban
    IBAN number.
        Type str
code
    Bank code.
        Type str
branchcode
    Branch code.
        Type str
prefix = bank
context
name
branch
region
route
number
acctype
iban
code
branchcode
class InvManage.views.Product
    Bases: django.db.models.Model
    Model of the product.
    name
        Name of the product.
        Type str
    category
        Product category.
        Type str
    item_type
        Type of the product.
        Type str

```

description

Short description of the product.

Type str

price

Price of the product.

Type float

quantity

Stock quantity of the product.

Type int

identifier

Unique identifier of the product.

Type str

location

Physical location of the product.

Type str

length

Length of the product.

Type str

width

Width of the product.

Type str

height

Height of the product.

Type str

weight

Weight of the product.

Type str

discount

Default discount percentage on the product.

Type float

barcode

Barcode of the product.

Type str

expiry

Expiry date of the product.

Type DateField

image

Image of the product.

Type ImageField

name

```

category
item_type
description
price
quantity
identifier
location
length
width
height
weight
discount
barcode
expiry
image
__str__(self)

```

```

class InvManage.views.ProductPurchaseEntry
    Bases: django.db.models.Model

    Model of the product purchase entry.

    product
        Product associated wit the product purchase entry.
        Type ModelChoiceField

    quantity
        Quantity of the product to be ordered.
        Type int

    price
        Price of the product.
        Type float

    discount
        Percentage discount on the product purchase.
        Type float

    order
        Referenced purchase order.
        Type PurchaseOrder

    receivedQty
        Quantity of product received against the ordered quantity.
        Type str

    acceptedQty
        Quantity of product accepted as OK.

```

Type str

rejectedQty
Quantity of product rejected (not OK, on HOLD, extra delivery, etc.)

Type str

is_complete()
Returns the completion status (boolean).

pending_quantity()
Returns the pending quantity against the entry (int).

product

quantity

price

discount

order

receivedQty

acceptedQty

rejectedQty

is_complete(self)

property pending_quantity(self)

class InvManage.views.GRNEntry
Bases: django.db.models.Model

Model of the goods receipt note entry (GRNE).

product
Primary key of the Product instance associated with the goods receipt note entry.

Type *Product*

quantity
Ordered quantity of product with reference to product purchase entry.

Type int

grne
Primary key of the GoodsReceiptNote instance.

Type int

ppe
Primary key of the ProductPurchaseEntry instance associated with the GRNE.

Type int

remark
Remarks of the quality engineer or the GRN creator about status of products received.

Type str

receivedQty
Quantity of product received against the ordered quantity.

Type str

acceptedQty

Quantity of product accepted as OK.

Type str**rejectedQty**

Quantity of product rejected (not OK, on HOLD, extra delivery, etc.)

Type str**product****quantity****grn****ppes****remark****receivedQty****acceptedQty****rejectedQty****class** InvManage.views.ProductSalesEntry

Bases: django.db.models.Model

Model of the product sales entry.

product

Primary key of the Product associated with the product sales entry.

Type *Product***quantity**

Quantity of the product.

Type int**price**

Price of the product.

Type float**discount**

Percentage discount on the product.

Type float**order**

Primary key of the SalesOrder referenced by the entry.

Type *SalesOrder***product****quantity****price****discount****order**

```
class InvManage.views.SalesOrder
    Bases: django.db.models.Model

    Model of the sales order.

    consumer
        Consumer associated with the sales order.

        Type ModelChoiceField

    date
        Date of the sales order creation.

        Type DateField

    so
        Sales order number.

        Type int

    discount
        Percentage discount associated with the sales order.

        Type float

    tax
        Percentage of tax applicable.

        Type float

    paid
        Amount received from the consumer.

        Type float

    balance
        Amount balance with the consumer.

        Type float

    subtotal
        Total of all the product sales entries associated with the sales order.

        Type float

    taxtotal
        Total tax applicable on the subtotal.

        Type float

    ordertotal
        Total price of the sales order including taxtotal.

        Type float

    consumer
    date
    so
    discount
    tax
    paid
    balance
```

```

subtotal
taxtotal
ordertotal
class InvManage.views.PurchaseOrder
    Bases: django.db.models.Model
    Model of the purchase order.
    vendor
        Vendor associated with the purchase order.
        Type Vendor
    date
        Date and time of the purchase order creation.
        Type DateTimeField
    po
        Purchase order number.
        Type int
    discount
        Percentage discount.
        Type float
    tax
        Percentage of tax applicable for the purchase.
        Type float
    paid
        Amount paid against the PO.
        Type float
    balance
        Amount balance which remains to be paid.
        Type float
    subtotal
        Total of all the product purchase entries associated with the purchase order.
        Type float
    taxtotal
        Total tax applicable on the subtotal.
        Type float
    ordertotal
        Total price of the purchase order including taxtotal.
        Type float
    is_complete()
        Returns the completion status (boolean).
    pending_ppes()
        Returns list of product purchase entries which are not completed.

```

vendor

date

po

discount

tax

paid

balance

subtotal

taxtotal

ordertotal

is_complete(*self*)

pending_ppes(*self*)

class InvManage.views.GoodsReceiptNote

Bases: django.db.models.Model

Model of the goods receipt note (GRN).

vendor

Vendor associated with the goods receipt note.

Type ModelChoiceField

poRef

List of identifiers of the purchase orders from which the goods receipt note is derived.

Type MultipleChoiceField

identifier

Unique identifier of the goods receipt note.

Type str

date

Date of GRN creation.

Type DateField

grnType

Type of GRN (auto or manual).

Type ChoiceField

amendNumber

Amendment number of the GRN.

Type int

amendDate

Amendment date.

Type DateField

transporter

Name of the transport/shipping service.

Type str

vehicleNumber

Vehicle number using which the products are shipped.

Type str

gateInwardNumber

Gate inward number of the vehicle.

Type str

preparedBy

Name/identifier of the person who created the goods receipt note.

Type str

checkedBy

Name/identifier of the person who validated the goods receipt note.

Type str

checkedBy

Name/identifier of the person who inspected the physical products in the goods receipt note.

Type str

approvedBy

Name/identifier of the authority who approved the goods receipt note.

Type str

TYPE_CHOICES = [['manual', 'Blank'], ['auto', 'PO Reference']]

vendor**poRef****identifier****date****grnType****amendNumber****amendDate****transporter****vehicleNumber****gateInwardNumber****preparedBy****checkedBy****inspectedBy****approvedBy****class** InvManage.views.Consumer

Bases: django.db.models.Model

Model of the consumer.

name

Name of the consumer.

Type str

identifier

Unique identifier of the consumer.

Type str

gstn

GSTIN number of the consumer.

Type str

phone

Contact number.

Type str

address

Address of the consumer.

Type str

email

E-mail address of the consumer.

Type str

location

City of the consumer.

Type str

Returns [description]

Return type [type]

name**identifier****gstn****phone****address****email****location**

__str__ (*self*)

class InvManage.views.**Vendor**

Bases: django.db.models.Model

Model of the vendor.

name

Name of the vendor.

Type str

identifier

Unique identifier of the vendor.

Type str

gstn

GSTIN number of the vendor.

```

        Type str

address
    Primary key of the ShippingAddress instance associated with the vendor.

        Type ShippingAddress

communication
    Primary key of the Communication instance associated with the vendor.

        Type Communication

backaccount
    Primary key of the BankAccount instance associated with the vendor.

        Type BankAccount

purchasedata
    Primary key of the PurchaseData instance associated with the vendor.

        Type PurchaseData

Returns String representation of the vendor name.

Return type str

name
identifier
gstn
address
communication
bankaccount
purchasedata
__str__(self)

class InvManage.views.Company
    Bases: django.db.models.Model

    Model of the company.

    name
        Name of the company.

            Type str

    owner
        Owner of the company.

            Type str

    gstn
        GSTIN number of the company.

            Type str

    phone
        Contact number.

            Type str

```

address

Postal address.

Type str

email

E-mail address of the contact person.

Type str

location

City of the company.

Type str

image

Photo/logo of the company.

Type ImageField

shippingaddress

Primary key of the ShippingAddress instance associated with the company.

Type *ShippingAddress*

Returns String representation of the company name.

Return type str

name**owner****gstn****phone****address****email****location****image****shippingaddress**

__str__ (*self*)

class InvManage.views.**PurchaseInvoice**

Bases: django.db.models.Model

Model for the purchase invoice.

company

Primary key of the Company.

Type *Company*

po

Primary key of the PurchaseOrder.

Type *PurchaseOrder*

shippingaddress

Primary key of the ShippingAddress.

Type *ShippingAddress*

communication

Primary key of the Communication.

Type *Communication*

company

po

shippingaddress

communication

class `InvManage.views.SalesInvoice`

Bases: `django.db.models.Model`

Model for the sales order invoice.

company

Primary key of the Company.

Type *Company*

so

Primary key of the SalesOrder.

Type *SalesOrder*

shippingaddress

Primary key of the ShippingAddress.

Type *ShippingAddress*

company

so

shippingaddress

class `InvManage.views.FilterState`

Bases: `django.db.models.Model`

Model of the filter state.

name

Name of the filter state.

Type `str`

name

class `InvManage.views.FilterColumn`

Bases: `django.db.models.Model`

Model of the column state.

name

Name of the column.

Type `str`

label

Label of the column.

Type `str`

visible

Visibility flag.

Type bool**position**

Index of the column.

Type int**state**Primary key of the `FilterState` instance being referenced by the column instance.**Type** *FilterState***name****label****visible****position****state****class** `InvManage.views.Object`Bases: `django.db.models.Model`

Model for a generic object.

identifier

Unique identifier of the object.

Type str**name**

Name of the object.

Type str**model**

Model associated with the object.

Type str**identifier****name****model****class** `InvManage.views.EventCard`Bases: `django.db.models.Model`

Model for event card to be displayed in the history view.

obj

Object associated with the event.

Type `OneToOneField`**objId**

Unique identifier of the object.

Type str**objname**

Name of the object.

```

        Type str
objmodel
    Model of the object.
        Type str
date
    Timestamp of the event.
        Type DateField
operation
    Type of event.
        Type str
obj
objId
objname
objmodel
date
operation
class InvManage.views.EventType
    Bases: django.db.models.Model
    Model for event type.
name
    Name of the event being triggered.
        Type str
label
    Label associated with the event.
        Type str

    Returns Returns the string representation of the label when the event type is queried.
    Return type Label

name
label
    __str__(self)
class InvManage.views.ObjectModel
    Bases: django.db.models.Model
    Model associated with an object.
name
    Name of the object.
        Type str
label
    Label string of the object.

```

Type str

modName

Name of the model associated with the object.

Type str

Returns Returns string representation of the label when object model is queried.

Return type Label

name

label

modName

__str__(*self*)

class InvManage.views.**HistoryFilterState**

Bases: django.db.models.Model

Model of filter state of the history.

params

Paramameters of the configured filters.

Type str

params

class InvManage.views.**ShippingAddress**

Bases: django.db.models.Model

Model of the shipping address.

title

Title of the firm.

Type str

name

Name of the firm.

Type str

phone

Contact number of the firm.

Type str

address

Postal address.

Type str

city

City.

Type str

state

State.

Type str

country

Country.

Type str**website**

Official website of the firm.

Type URLField**post**

Postal code of the firm.

Type str**title****name****phone****address****city****state****country****website****post****class** InvManage.views.PurchaseData

Bases: django.db.models.Model

Model of the purchase data.

currency

Currency of purchase.

Type str**minorder**

Minimum order quantity.

Type int**contactperson**

Name of the contact person.

Type str**refcode**

Reference code.

Type str**transportmode**

Mode of transport of the shipment.

Type str**currency****minorder****contactperson**

refcode

transportmode

class InvManage.views.Communication

Bases: django.db.models.Model

Model of the communication information.

language

Language of communication.

Type str

phone

Contact number of the firm.

Type str

email

E-mail of the contact person.

Type EmailField

fax

Fax number.

Type str

language

phone

email

fax

class InvManage.views.BankAccount

Bases: django.db.models.Model

Model of the bank account details.

name

Name of the bank.

Type str

branch

Branch number of the bank.

Type str

region

City in which the branch is located.

Type str

route

Transit number.

Type str

number

Bank account number.

Type int

acctype

Type of bank account.

Type str

iban

IBAN number.

Type str

code

Bank code.

Type str

branchcode

Branch code.

Type str

name**branch****region****route****number****acctype****iban****code****branchcode****InvManage.views.get_columns(*state*)**

Gets the columns of the display table.

This function takes a filter state and returns an array of column names that are ordered depending on the position attribute of the columns. This order is important as the queryset will be rendered in the display table according to the column order in this returned array.

Parameters **state** (*FilterState*) – Filter state of the columns.

Returns A list that dictates the order of the columns.

Return type List

InvManage.views.sort_ascending_descending(*request, model*)

Sorts and saves the state of the column.

This function takes a request, finds which column needs to be sorted in ascending/descending order, and returns the sorted queryset.

Parameters

- **request** (*HttpRequest*) – GET request received for retrieving the list of model instances.
- **model** (*Model*) – Model of the objects being retrieved.

Returns List of sorted object instances.

Return type QuerySet

`InvManage.views.change_column_position(request, state)`

Modifies the column states depending on the request.

This function:

- Takes a filter state
- Extracts the column names into an order array
- Modifies the column order in the array depending on user input direction (left/right)
- Saves this modified state of columns into the database
- Returns the modified column names array

Parameters

- **request** (*HttpRequest*) – GET request for retrieving the state of the columns.
- **state** (*FilterState*) – Filter state of the columns.

Returns List of sorted columns.

Return type List

`InvManage.views.paginate(queryset, filter, page_number)`

Paginates the list of objects.

This function takes the entire queryset and filters out only objects belonging to the request page

Parameters

- **queryset** (*QuerySet*) – QuerySet of the objects retrieved from the database.
- **filter** (*FilterState*) – Filter state of the columns.
- **page_number** (*int*) – Page number requested.

```
class InvManage.views.SalesOrderFilter
```

```
    Bases: django_filters.FilterSet
```

```
    class Meta
```

```
        model
```

```
        fields
```

```
    context
```

```
    consumer
```

```
    date
```

```
    so
```

```
    ordertotal
```

```
    order_by_field = ordering
```

```
    ordering
```

```
class InvManage.views.ProductSerializer
```

```
    Bases: rest_framework.serializers.ModelSerializer
```

```
    Serializer for Product class instance.
```

```
    The ProductSerializer.data attribute gives the JSON serialized data of the Product instance:
```



```
{
  "pk": 637,
  "name": "Piano",
  "category": "Ultricies PC",
  "quantity": 23921,
  "identifier": "PROD9",
  "location": "Musselburgh",
  "description": "88-key, Tri-sensor Scaled Hammer Action Keyboard II, Simulated_
↳ebony and ivory keys ",
  "prod_id": 637
}
```

```
class Meta
```

```
    model
```

```
    fields = ['pk', 'name', 'category', 'quantity', 'identifier', 'location',
              'description', 'prod_id']
```

```
    prod_id
```

```
    to_representation(self, instance)
```

```
class InvManage.views.PSEntrySerializer
```

```
    Bases: rest_framework.serializers.ModelSerializer
```

```
    Serializer for ProductSalesEntry class instance.
```

```
    The PSEntrySerializer.data attribute gives the JSON serialized data of the ProductSalesEntry instance:
```

```
{
  "pse_id": 64,
  "product": {
    "pk": 637,
    "name": "Piano",
    "category": "Ultricies PC",
    "quantity": 23921,
    "identifier": "PROD9",
    "location": "Musselburgh",
    "description": "88-key, Tri-sensor Scaled Hammer Action Keyboard II,
↳Simulated ebony and ivory keys ",
    "prod_id": 637
  },
  "quantity": 50,
  "price": 9900.0,
  "discount": 10.0,
  "order": 29
}
```

```
class Meta
```

```
    model
```

```
    fields = ['pse_id', 'product', 'quantity', 'price', 'discount', 'order']
```

```
    pse_id
```

product

to_representation(*self*, *instance*)

create(*self*, *validated_data*)

update(*self*, *instance*, *validated_data*)

class InvManage.views.SalesInvoiceSerializer

Bases: `rest_framework.serializers.ModelSerializer`

Serializer for SalesInvoice class instance.

The `SalesInvoiceSerializer.data` attribute gives the JSON serialized data of the `SalesInvoice` instance:

```
{
  "company": {
    "name": "Fringillami",
    "owner": "Ivor Barnett",
    "gstn": "89AAC056465468",
    "phone": "332 220-7026",
    "address": "Ap #849-6241 Euismod Av., 677598, Carinthia, Belgium",
    "email": "est.tempor@fringillami.org",
    "location": "Belgium",
    "image": "/media/images/hyperlink-green_x91WW5n.png"
  },
  "so": {
    "consumer": {
      "name": "The Music Store",
      "identifier": "CONS1256",
      "gstn": "89AAC4633353643",
      "phone": "+91 8325642358",
      "address": "Plot no 958, N- 4, Neo Complex, Barh, Wokha, Nagaland, ↵
↵797111",
      "email": "JohnDoe@themusic.store",
      "location": "Wokha"
    },
    "date": "2021-09-25T00:00:00Z",
    "so": 89,
    "subtotal": 744900.0,
    "taxtotal": 59592.0,
    "ordertotal": 804492.0,
    "pses": [
      {
        "pse_id": 64,
        "product": {
          "pk": 637,
          "name": "piano",
          "category": "Ultricies PC",
          "quantity": 23921,
          "identifier": "PROD9",
          "location": "Musselburgh",
          "description": "88-key, Tri-sensor Scaled Hammer Action ↵
↵Keyboard II, Simulated ebony and ivory keys ",
          "prod_id": 637
        }
      }
    ]
  }
}
```

(continues on next page)

(continued from previous page)

```

        "quantity": 50,
        "price": 9900.0,
        "discount": 10.0,
        "order": 29
    },
    {
        "pse_id": 65,
        "product": {
            "pk": 649,
            "name": "Sabar",
            "category": "Amet Consulting",
            "quantity": 3903,
            "identifier": "PROD21",
            "location": "Serang",
            "description": "High Sierra Sabar, Travel bag, Blue, Grey,
↪Zipper, 36.5 L, 51.5 cm, 26 cm ",
            "prod_id": 649
        },
        "quantity": 25,
        "price": 4800.0,
        "discount": 8.0,
        "order": 29
    },
    {
        "pse_id": 66,
        "product": {
            "pk": 654,
            "name": "Parai",
            "category": "Aliquet Lobortis Ltd",
            "quantity": 8534,
            "identifier": "PROD26",
            "location": "Burhaniye",
            "description": "Kannan musical instruments Parai 15" inch
↪(Baffallow skin) Daf Instrument",
            "prod_id": 654
        },
        "quantity": 35,
        "price": 6000.0,
        "discount": 10.0,
        "order": 29
    }
}
    ],
    "shippingaddress": {
        "name": "Harding Gross",
        "address": "8798 At, St., 7639",
        "city": "Rome",
        "phone": "936 651-4847",
        "state": "Lazio",
        "country": "Italy",
        "post": "300326"
    }
}

```

(continues on next page)

(continued from previous page)

}

class Meta**model****fields** = ['company', 'so', 'shippingaddress']**company****so****shippingaddress**

InvManage.views.create_sales_order_view(request)

Creates a sales order (SO) on POST request, and returns a SO creation form on GET request.

GET /sales_order

Gets the sales order creation form.

Example request:

```
GET /sales_order/ HTTP/1.1
Host: localhost:8000
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: text/html; charset=utf-8
```

Request Headers

- **Accept** – text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9

Status Codes

- **200 OK** – Sales order creation form received successfully.

POST /sales_order

Creates a sales order.

Example request:

```
POST /sales_order/ HTTP/1.1
Host: localhost:8000
Content-Type: multipart/form-data;
```

Form Parameters

- **so-consumer** – 5
- **consumer-name** – The Music Store
- **consumer-identifier** – CONS1256

- **consumer-gstin** – 89ACC654654335
- **consumer-phone** – +91 8325642358
- **consumer-address** – Plot no 958, N- 4, Neo Complex, Barh, Wokha, Nagaland, 797111
- **consumer-email** – JohnDoe@themusic.store
- **consumer-location** – Wokha
- **so-so** – 465
- **so-date** – 2021-09-30
- **so-tax** – 12
- **so-discount** – 8
- **so-paid** – 4500
- **so-balance** – 1200
- **ship-title** – FingDocks
- **ship-name** – Harding Gross
- **ship-phone** – 936 651-4847
- **ship-address** – 8798 At, St., 7639
- **ship-city** – Rome
- **ship-state** – Lazio
- **ship-country** – Italy
- **ship-website** – <http://fringdocs.com>
- **ship-post** – 300326
- **so-subtotal** – 94600.00
- **so-taxtotal** – 7568.00
- **so-ordertotal** – 102168.00
- **form-TOTAL_FORMS** – 2
- **form-INITIAL_FORMS** – 0
- **form-MIN_NUM_FORMS** –
- **form-MAX_NUM_FORMS** –
- **form-0-product** – 645
- **form-0-quantity** – 450
- **form-0-price** – 120
- **form-0-discount** – 10
- **form-0-DELETE** –
- **form-0-pse_id** – -1
- **form-1-product** – 654
- **form-1-quantity** – 500

- **form-1-price** – 100
- **form-1-discount** – 8
- **form-1-DELETE** –
- **form-1-pse_id** – -1

Response Headers

- **Content-Type** – multipart/form-data; boundary=---WebKitFormBoundaryLTR88aZAnBUSE7mv

Status Codes

- **302 Found** – Redirects to /sales_order.

`InvManage.views.update_sales_order_view(request)`

Updates a sales order on POST request and returns the sales order update form for GET request.

GET /sales_order/update

Gets the sales order update form whose primary key matches the query parameter pk.

Example request:

```
GET /sales_order/update HTTP/1.1
Host: localhost:8000
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
```

Query Parameters

- **pk** – The primary key of the sales order.

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: text/html; charset=utf-8
```

Request Headers

- **Accept** – text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9

Status Codes

- **200 OK** – Sales order update form received successfully.

POST /sales_order/update

Updates the sales order.

Example request:

```
POST /sales_order/update HTTP/1.1
Host: localhost:8000
Content-Type: multipart/form-data;
```

Form Parameters

- **pk** – 30

- **so-consumer** – 7
- **consumer-name** – The Music Store
- **consumer-identifier** – CONS1256
- **consumer-gstin** – 89AAC4633353643
- **consumer-phone** – +91 8325642358
- **consumer-address** – Plot no 958, N- 4, Neo Complex, Barh, Wokha, Nagaland, 797111
- **consumer-email** – JohnDoe@themusic.store
- **consumer-location** – Wokha
- **so-so** – 465
- **so-date** – 2021-09-30
- **so-tax** – 10
- **so-discount** – 8.0
- **so-paid** – 4500.0
- **so-balance** – 1200.0
- **ship-title** – FingDocks
- **ship-name** – Harding Gross
- **ship-phone** – 936 651-4847
- **ship-address** – 8798 At, St., 7639
- **ship-city** – Rome
- **ship-state** – Lazio
- **ship-country** – Italy
- **ship-website** – <http://fringdocs.com>
- **ship-post** – 300326
- **so-subtotal** – 94600.0
- **so-taxtotal** – 7568.0
- **so-ordertotal** – 102168.0
- **form-TOTAL_FORMS** – 2
- **form-INITIAL_FORMS** – 2
- **form-MIN_NUM_FORMS** –
- **form-MAX_NUM_FORMS** –
- **form-0-product** – 645
- **form-0-quantity** – 450
- **form-0-price** – 120
- **form-0-discount** – 10
- **form-0-DELETE** –

- **form-0-pse_id** – 67
- **form-1-product** – 654
- **form-1-quantity** – 500
- **form-1-price** – 100
- **form-1-discount** – 8
- **form-1-DELETE** –
- **form-1-pse_id** – 68

Response Headers

- **Content-Type** – multipart/form-data; boundary=—WebKitFormBoundaryLTR88aZAnBUSE7mv

Status Codes

- **302 Found** – Redirects to /sales_order.

`InvManage.views.delete_sales_order_view(request, pk)`

Deletes the sales order with primary key `object_id` on POST request.

POST /sales_order/<str:object_id>/delete

Deletes the sales order represented by the primary key `object_id`.

Example request:

```
POST /sales_order/30/delete HTTP/1.1
Host: localhost:8000
Content-Type: application/x-www-form-urlencoded
```

Parameters

- **object_id** – Sales order primary key.

Response Headers

- **Content-Type** – application/x-www-form-urlencoded

Status Codes

- **302 Found** – Redirects to /sales_order.
- **500 Internal Server Error** – Sales order matching query does not exist.

`InvManage.views.display_sales_orders_view(request)`

Retrieves the list of sales orders on GET request.

GET /sales_orders/

Gets the list of all sales orders.

Example request:

```
GET /sales_orders/ HTTP/1.1
Host: localhost:8000
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/
webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
```

Form Parameters

- **page** – The page number of the sales order list.

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: text/html; charset=utf-8
```

Request Headers

- **Accept** – text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9

Status Codes

- **200 OK** – List of sales orders received successfully.

`InvManage.views.print_sales_order_view(request, pk)`

Opens the sales order with primary key `so_id` for printing on GET request.

POST /purchase_order/<str:so_id>/print

Opens the sales order represented by the primary key `so_id`.

Example request:

```
POST /sales_order/182/print HTTP/1.1
Host: localhost:8000
Content-Type: application/x-www-form-urlencoded
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: application/json; charset=utf-8

[
  {
    "company": {
      "name": "Fringillami",
      "owner": "Ivor Barnett",
      "gstn": "89AAC056465468",
      "phone": "332 220-7026",
      "address": "Ap #849-6241 Euismod Av., 677598, Carinthia, Belgium",
      "email": "est.tempor@fringillami.org",
      "location": "Belgium",
      "image": "/media/images/hyperlink-green_x91WW5n.png"
    },
    "so": {
      "consumer": {
        "name": "The Music Store",
        "identifier": "CONS1256",
        "gstn": "89AAC4633353643",
        "phone": "+91 8325642358",
        "address": "Plot no 958, N- 4, Neo Complex, Barh, Wokha, Nagaland, 797111",
        "email": "JohnDoe@themusic.store",
        "location": "Wokha"
      }
    }
  }
]
```

(continues on next page)

(continued from previous page)

```

    "date": "2021-09-25T00:00:00Z",
    "so": 89,
    "subtotal": 744900.0,
    "taxtotal": 59592.0,
    "ordertotal": 804492.0,
    "pses": [
      {
        "pse_id": 64,
        "product": {
          "pk": 637,
          "name": "piano",
          "category": "Ultricies PC",
          "quantity": 23921,
          "identifier": "PROD9",
          "location": "Musselburgh",
          "description": "88-key, Tri-sensor Scaled Hammer Action_
↳Keyboard II, Simulated ebony and ivory keys ",
          "prod_id": 637
        },
        "quantity": 50,
        "price": 9900.0,
        "discount": 10.0,
        "order": 29
      },
      {
        "pse_id": 65,
        "product": {
          "pk": 649,
          "name": "Sabar",
          "category": "Amet Consulting",
          "quantity": 3903,
          "identifier": "PROD21",
          "location": "Serang",
          "description": "High Sierra Sabar, Travel bag, Blue,
↳Grey, Zipper, 36.5 L, 51.5 cm, 26 cm ",
          "prod_id": 649
        },
        "quantity": 25,
        "price": 4800.0,
        "discount": 8.0,
        "order": 29
      },
      {
        "pse_id": 66,
        "product": {
          "pk": 654,
          "name": "Parai",
          "category": "Aliquet Lobortis Ltd",
          "quantity": 8534,
          "identifier": "PROD26",
          "location": "Burhaniye",
          "description": "Kannan musical instruments Parai 15"
↳inch (Baffallow skin) Daf Instrument",

```

(continues on next page)

(continued from previous page)

```

        "prod_id": 654
    },
    "quantity": 35,
    "price": 6000.0,
    "discount": 10.0,
    "order": 29
}
]
},
"shippingaddress": {
    "name": "Harding Gross",
    "address": "8798 At, St., 7639",
    "city": "Rome",
    "phone": "936 651-4847",
    "state": "Lazio",
    "country": "Italy",
    "post": "300326"
}
}
]

```

Parameters

- **so_id** – Sales order primary key.

Response Headers

- **Content-Type** – application/json

Status Codes

- **200 OK** – Sales order print request successful.
- **500 Internal Server Error** – Sales order matching query does not exist.

```
class InvManage.views.ProductBasicInfoForm
```

```
    Bases: django.forms.Form
```

```
    Form for basic product information.
```

```
    name
```

```
        Name of the product.
```

```
        Type str
```

```
    item_type
```

```
        Type of the product.
```

```
        Type str
```

```
    category
```

```
        Product category.
```

```
        Type str
```

```
    description
```

```
        Short description of the product.
```

```
        Type str
```

```
    prefix = basic
    context
    name
    item_type
    category
    description

class InvManage.views.ProductDetailedInfoForm
    Bases: django.forms.Form

    Form for product physical details

    length
        Length of the product.
        Type str

    width
        Width of the product.
        Type str

    height
        Height of the product.
        Type str

    weight
        Weight of the product.
        Type str

    prefix = detailed
    context
    length
    width
    height
    weight

class InvManage.views.ProductPricingForm
    Bases: django.forms.Form

    Form for pricing information of the product.

    price
        Price of the product.
        Type float

    discount
        Default discount percentage on the product.
        Type float

    prefix = pricing
    context
    price
```

```

    discount
class InvManage.views.ProductPurchaseEntryForm
    Bases: django.forms.Form
    Form for product purchase entry.
    ppe_id
        Unique identifier of the product purchase entry.
        Type int
    product
        Product associated wit the product purchase entry.
        Type Product
    quantity
        Quantity of the product to be ordered.
        Type int
    price
        Price of the product.
        Type float
    discount
        Percentage discount on the product purchase.
        Type float
    prefix = form
    context
    ppe_id
    product
    quantity
    price
    discount
class InvManage.views.ProductSalesEntryForm
    Bases: django.forms.Form
    Form for product sales entry.
    pse_id
        Unique identifier of the product sales entry.
        Type int
    product
        Product associated with the product sales entry.
        Type Product
    quantity
        Quantity of the product.
        Type int
    price
        Price of the product.

```

Type float

discount

Percentage discount on the product.

Type float

prefix = form

context

pse_id

product

quantity

price

discount

class InvManage.views.**ProductStatusForm**

Bases: django.forms.Form

Form for locating and identifying the product.

quantity

Stock quantity of the product.

Type int

identifier

Unique identifier of the product.

Type str

location

Physical location of the product.

Type str

context

prefix = status

quantity

identifier

location

class InvManage.views.**ProductStorageInfoForm**

Bases: django.forms.Form

Form for storage information.

barcode

Barcode of the product.

Type str

expiry

Expiry date of the product.

Type DateField

prefix = storage

barcode

```

context
expiry
class InvManage.views.PurchaseOrderBasicInfo
    Bases: django.forms.Form
    Form for basic information of the purchase order.
vendor
    Vendor associated with the purchahse order.
        Type Vendor
date
    Date of the purchase order creation.
        Type DateField
po
    Purchase order number.
        Type int
discount
    Percentage of overall discount.
        Type float
tax
    Percentage of tax.
        Type float
paid
    Amount paid to the vendor.
        Type float
balance
    Balance amount to be paid to the vendor.
        Type float
subtotal
    Total of all the product purchase entries associated with the purchase order.
        Type float
taxtotal
    Total tax applicable on the subtotal.
        Type float
ordertotal
    Total price of the purchase order including taxtotal.
        Type float
prefix = po
context
product_choices
vendor_choices
vendor

```

date

po

discount

tax

paid

balance

subtotal

taxtotal

ordertotal

class `InvManage.views.GRNInfo`

Bases: `django.forms.Form`

Form for goods receipt note (GRN).

vendor

Vendor associated with the goods receipt note.

Type *Vendor*

poRef

List of identifiers of the purchase orders from which the goods receipt note is derived.

Type *PurchaseOrder*

identifier

Unique identifier of the goods receipt note.

Type `str`

date

Date of GRN creation.

Type `DateField`

grnType

Type of GRN (auto or manual).

Type `str`

amendNumber

Amendment number of the GRN.

Type `int`

amendDate

Amendment date.

Type `DateField`

transporter

Name of the transport/shipping service.

Type `str`

vehicleNumber

Vehicle number using which the products are shipped.

Type `str`


```

gateInwardNumber
    Gate inward number of the vehicle.

    Type str

preparedBy
    Name/identifier of the person who created the goods receipt note.

    Type str

checkedBy
    Name/identifier of the person who validated the goods receipt note.

    Type str

checkedBy
    Name/identifier of the person who inspected the physical products in the goods receipt note.

    Type str

approvedBy
    Name/identifier of the authority who approved the goods receipt note.

    Type str

prefix = grn
context
product_choices
vendor_choices
vendor
TYPE_CHOICES = [['manual', 'Blank'], ['auto', 'PO Reference']]
poRef
identifier
date
grnType
amendNumber
amendDate
transporter
vehicleNumber
gateInwardNumber
preparedBy
checkedBy
inspectedBy
approvedBy

class InvManage.views.GRNEntryForm
    Bases: django.forms.Form

    Form for goods receipt note entry (GRNE).

```

product

Product associated with the goods receipt note entry.

Type *Product*

grne_id

Unique identifier of the GRNE.

Type `int`

ppe_id

Unique identifier of the product purchase entry associated with the GRNE.

Type `int`

quantity

Ordered quantity of product with reference to product purchase entry.

Type `int`

remark

Remarks of the quality engineer or the GRN creator about status of products received.

Type `str`

receivedQty

Quantity of product received against the ordered quantity.

Type `str`

acceptedQty

Quantity of product accepted as OK.

Type `str`

rejectedQty

Quantity of product rejected (not OK, on HOLD, extra delivery, etc.)

Type `str`

prefix = form**context****product****grne_id****ppe_id****quantity****remark****receivedQty****acceptedQty****rejectedQty****class** `InvManage.views.ThumbnailForm`

Bases: `django.forms.Form`

For for product thumbnail image.

image

Image of the product.

```

        Type ImageField
    prefix = thumbnail
    image
class InvManage.views.SalesOrderBasicInfo
    Bases: django.forms.Form
    Form for basic information on sales order.
    consumer
        Consumer associated with the sales order.
        Type Consumer
    date
        Date of the sales order creation.
        Type DateField
    so
        Sales order number.
        Type int
    discount
        Percentage discount associated with the sales order.
        Type float
    tax
        Percentage of tax applicable.
        Type float
    paid
        Amount received from the consumer.
        Type float
    balance
        Amount balance with the consumer.
        Type float
    subtotal
        Total of all the product sales entries associated with the sales order.
        Type float
    taxtotal
        Total tax applicable on the subtotal.
        Type float
    ordertotal
        Total price of the sales order including taxtotal.
        Type float
    prefix = so
    context
    product_choices
    consumer_choices

```

consumer
date
so
discount
tax
paid
balance
subtotal
taxtotal
ordertotal

class InvManage.views.**ShippingAddressForm**

Bases: django.forms.Form

For for shipping address.

title
Title of the firm.

Type str

name
Name of the firm.

Type str

phone
Contact number of the firm.

Type str

address
Postal address.

Type str

city
City.

Type str

state
State.

Type str

country
Country.

Type str

website
Official website of the firm.

Type URLField

post
Postal code of the firm.

```

        Type str
    prefix = ship
    context
    title
    name
    phone
    address
    city
    state
    country
    website
    post
class InvManage.views.VendorForm
    Bases: django.forms.Form
    Form for basic informaion of the vendor.
    name
        Name of the vendor.
        Type str
    identifier
        Unique identifier of the vendor.
        Type str
    gstn
        GSTIN number of the vendor.
        Type str
    prefix = vend
    context
    name
    identifier
    gstn
class InvManage.views.ConsumerForm
    Bases: django.forms.Form
    Form for consumer.
    name
        Name of the consumer.
        Type str
    identifier
        Unique identifier of the consumer.
        Type str

```

gstn

GSTIN number of the consumer.

Type str**phone**

Contact number.

Type str**address**

Address of the consumer.

Type str**email**

E-mail address of the consumer.

Type str**location**

City of the consumer.

Type str**prefix = consumer****context****name****identifier****gstn****phone****address****email****location****class** InvManage.views.CompanyForm

Bases: django.forms.Form

Form for company.

name

Name of the company.

Type str**owner**

Name of the owner of the company.

Type str**gstn**

GSTIN number of the company.

Type str**phone**

Contact number of the company.

Type str

address

Postal address.

Type str**email**

E-mail address of the contact person.

Type str**location**

City of the company.

Type str**prefix = comp****context****name****owner****gstn****phone****address****email****location****class** InvManage.views.**HistoryForm**

Bases: django.forms.Form

Form for history view.

qlen

Count of the events visible on the history view.

Type int**prefix = history****context****qlen****class** InvManage.views.**CommunicationForm**

Bases: django.forms.Form

Form for communication information.

language

Language of communication.

Type str**phone**

Contact number of the firm.

Type str**email**

E-mail of the contact person.

Type EmailField

fax

Fax number.

Type str

prefix = com

context

language

phone

email

fax

class InvManage.views.**PurchaseDataForm**

Bases: django.forms.Form

Form for purchase related data.

currency

Currency of purchase.

Type str

minorder

Minimum order quantity.

Type int

contactperson

Name of the contact person.

Type str

refcode

Reference code.

Type str

transportmode

Mode of transport of the shipment.

Type str

prefix = pdform

context

currency

minorder

contactperson

refcode

transportmode

class InvManage.views.**BankAccountForm**

Bases: django.forms.Form

Form for bank account details.

name

Name of the bank.


```

        Type str
branch
    Branch number of the bank.
        Type str
region
    City in which the branch is located.
        Type str
route
    Transit number.
        Type str
number
    Bank account number.
        Type int
acctype
    Type of bank account.
        Type str
iban
    IBAN number.
        Type str
code
    Bank code.
        Type str
branchcode
    Branch code.
        Type str
prefix = bank
context
name
branch
region
route
number
acctype
iban
code
branchcode
class InvManage.views.Product
    Bases: django.db.models.Model
    Model of the product.

```

name
Name of the product.
Type str

category
Product category.
Type str

item_type
Type of the product.
Type str

description
Short description of the product.
Type str

price
Price of the product.
Type float

quantity
Stock quantity of the product.
Type int

identifier
Unique identifier of the product.
Type str

location
Physical location of the product.
Type str

length
Length of the product.
Type str

width
Width of the product.
Type str

height
Height of the product.
Type str

weight
Weight of the product.
Type str

discount
Default discount percentage on the product.
Type float

barcode
Barcode of the product.

```

        Type str

    expiry
        Expiry date of the product.

        Type DateField

    image
        Image of the product.

        Type ImageField

    name

    category

    item_type

    description

    price

    quantity

    identifier

    location

    length

    width

    height

    weight

    discount

    barcode

    expiry

    image

    __str__(self)

class InvManage.views.ProductPurchaseEntry
    Bases: django.db.models.Model

    Model of the product purchase entry.

    product
        Product associated wit the product purchase entry.

        Type ModelChoiceField

    quantity
        Quantity of the product to be ordered.

        Type int

    price
        Price of the product.

        Type float

    discount
        Percentage discount on the product purchase.

```

Type float

order

Referenced purchase order.

Type *PurchaseOrder*

receivedQty

Quantity of product received against the ordered quantity.

Type str

acceptedQty

Quantity of product accepted as OK.

Type str

rejectedQty

Quantity of product rejected (not OK, on HOLD, extra delivery, etc.)

Type str

is_complete()

Returns the completion status (boolean).

pending_quantity()

Returns the pending quantity against the entry (int).

product

quantity

price

discount

order

receivedQty

acceptedQty

rejectedQty

is_complete(self)

property pending_quantity(self)

class InvManage.views.GRNEEntry

Bases: django.db.models.Model

Model of the goods receipt note entry (GRNE).

product

Primary key of the Product instance associated with the goods receipt note entry.

Type *Product*

quantity

Ordered quantity of product with reference to product purchase entry.

Type int

grne

Primary key of the GoodsReceiptNote instance.

Type int

ppe

Primary key of the ProductPurchaseEntry instance associated with the GRNE.

Type int

remark

Remarks of the quality engineer or the GRN creator about status of products received.

Type str

receivedQty

Quantity of product received against the ordered quantity.

Type str

acceptedQty

Quantity of product accepted as OK.

Type str

rejectedQty

Quantity of product rejected (not OK, on HOLD, extra delivery, etc.)

Type str

product**quantity****grn****ppes****remark****receivedQty****acceptedQty****rejectedQty****class** InvManage.views.ProductSalesEntry

Bases: django.db.models.Model

Model of the product sales entry.

product

Primary key of the Product associated with the product sales entry.

Type *Product*

quantity

Quantity of the product.

Type int

price

Price of the product.

Type float

discount

Percentage discount on the product.

Type float

order

Primary key of the SalesOrder referenced by the entry.

Type *SalesOrder*

product

quantity

price

discount

order

class `InvManage.views.SalesOrder`

Bases: `django.db.models.Model`

Model of the sales order.

consumer

Consumer associated with the sales order.

Type `ModelChoiceField`

date

Date of the sales order creation.

Type `DateField`

so

Sales order number.

Type `int`

discount

Percentage discount associated with the sales order.

Type `float`

tax

Percentage of tax applicable.

Type `float`

paid

Amount received from the consumer.

Type `float`

balance

Amount balance with the consumer.

Type `float`

subtotal

Total of all the product sales entries associated with the sales order.

Type `float`

taxtotal

Total tax applicable on the *subtotal*.

Type `float`

ordertotal

Total price of the sales order including *taxtotal*.

Type `float`

consumer

```

    date
    so
    discount
    tax
    paid
    balance
    subtotal
    taxtotal
    ordertotal

class InvManage.views.PurchaseOrder
    Bases: django.db.models.Model

    Model of the purchase order.

    vendor
        Vendor associated with the purchase order.

        Type Vendor

    date
        Date and time of the purchase order creation.

        Type DateTimeField

    po
        Purchase order number.

        Type int

    discount
        Percentage discount.

        Type float

    tax
        Percentage of tax applicable for the purchase.

        Type float

    paid
        Amount paid against the PO.

        Type float

    balance
        Amount balance which remains to be paid.

        Type float

    subtotal
        Total of all the product purchase entries associated with the purchase order.

        Type float

    taxtotal
        Total tax applicable on the subtotal.

        Type float

```

ordertotal

Total price of the purchase order including *taxtotal*.

Type float

is_complete()

Returns the completion status (boolean).

pending_ppes()

Returns list of product purchase entries which are not completed.

vendor**date****po****discount****tax****paid****balance****subtotal****taxtotal****ordertotal****is_complete(self)****pending_ppes(self)****class** InvManage.views.GoodsReceiptNote

Bases: django.db.models.Model

Model of the goods receipt note (GRN).

vendor

Vendor associated with the goods receipt note.

Type ModelChoiceField

poRef

List of identifiers of the purchase orders from which the goods receipt note is derived.

Type MultipleChoiceField

identifier

Unique identifier of the goods receipt note.

Type str

date

Date of GRN creation.

Type DateField

grnType

Type of GRN (auto or manual).

Type ChoiceField

amendNumber

Amendment number of the GRN.

Type int

amendDate
Amendment date.

Type DateField

transporter
Name of the transport/shipping service.

Type str

vehicleNumber
Vehicle number using which the products are shipped.

Type str

gateInwardNumber
Gate inward number of the vehicle.

Type str

preparedBy
Name/identifier of the person who created the goods receipt note.

Type str

checkedBy
Name/identifier of the person who validated the goods receipt note.

Type str

checkedBy
Name/identifier of the person who inspected the physical products in the goods receipt note.

Type str

approvedBy
Name/identifier of the authority who approved the goods receipt note.

Type str

TYPE_CHOICES = [['manual', 'Blank'], ['auto', 'PO Reference']]

vendor

poRef

identifier

date

grnType

amendNumber

amendDate

transporter

vehicleNumber

gateInwardNumber

preparedBy

checkedBy

inspectedBy

approvedBy

class InvManage.views.Consumer

Bases: django.db.models.Model

Model of the consumer.

name

Name of the consumer.

Type str

identifier

Unique identifier of the consumer.

Type str

gstin

GSTIN number of the consumer.

Type str

phone

Contact number.

Type str

address

Address of the consumer.

Type str

email

E-mail address of the consumer.

Type str

location

City of the consumer.

Type str

Returns [description]

Return type [type]

name

identifier

gstin

phone

address

email

location

__str__ (*self*)

class InvManage.views.Vendor

Bases: django.db.models.Model

Model of the vendor.

name

Name of the vendor.

Type str

identifier

Unique identifier of the vendor.

Type str

gstln

GSTIN number of the vendor.

Type str

address

Primary key of the `ShippingAddress` instance associated with the vendor.

Type *ShippingAddress*

communication

Primary key of the `Communication` instance associated with the vendor.

Type *Communication*

backaccount

Primary key of the `BankAccount` instance associated with the vendor.

Type *BankAccount*

purchasedata

Primary key of the `PurchaseData` instance associated with the vendor.

Type *PurchaseData*

Returns String representation of the vendor name.

Return type str

name**identifier****gstln****address****communication****bankaccount****purchasedata****__str__** (*self*)

class `InvManage.views.Company`

Bases: `django.db.models.Model`

Model of the company.

name

Name of the company.

Type str

owner

Owner of the company.

Type str

gstn
GSTIN number of the company.

Type str

phone
Contact number.

Type str

address
Postal address.

Type str

email
E-mail address of the contact person.

Type str

location
City of the company.

Type str

image
Photo/logo of the company.

Type ImageField

shippingaddress
Primary key of the `ShippingAddress` instance associated with the company.

Type *ShippingAddress*

Returns String representation of the company name.

Return type str

name

owner

gstn

phone

address

email

location

image

shippingaddress

__str__ (*self*)

class `InvManage.views.PurchaseInvoice`
Bases: `django.db.models.Model`
Model for the purchase invoice.

company

Primary key of the Company.

Type *Company***po**

Primary key of the PurchaseOrder.

Type *PurchaseOrder***shippingaddress**

Primary key of the ShippingAddress.

Type *ShippingAddress***communication**

Primary key of the Communication.

Type *Communication***company****po****shippingaddress****communication****class** `InvManage.views.SalesInvoice`Bases: `django.db.models.Model`

Model for the sales order invoice.

company

Primary key of the Company.

Type *Company***so**

Primary key of the SalesOrder.

Type *SalesOrder***shippingaddress**

Primary key of the ShippingAddress.

Type *ShippingAddress***company****so****shippingaddress****class** `InvManage.views.FilterState`Bases: `django.db.models.Model`

Model of the filter state.

name

Name of the filter state.

Type `str`**name**

class `InvManage.views.FilterColumn`

Bases: `django.db.models.Model`

Model of the column state.

name

Name of the column.

Type `str`

label

Label of the column.

Type `str`

visible

Visibility flag.

Type `bool`

position

Index of the column.

Type `int`

state

Primary key of the `FilterState` instance being referenced by the column instance.

Type *`FilterState`*

name

label

visible

position

state

class `InvManage.views.Object`

Bases: `django.db.models.Model`

Model for a generic object.

identifier

Unique identifier of the object.

Type `str`

name

Name of the object.

Type `str`

model

Model associated with the object.

Type `str`

identifier

name

model

class InvManage.views.EventCard

Bases: django.db.models.Model

Model for event card to be displayed in the history view.

obj

Object associated with the event.

Type OneToOneField**objId**

Unique identifier of the object.

Type str**objname**

Name of the object.

Type str**objmodel**

Model of the object.

Type str**date**

Timestamp of the event.

Type DateField**operation**

Type of event.

Type str**obj****objId****objname****objmodel****date****operation****class** InvManage.views.EventType

Bases: django.db.models.Model

Model for event type.

name

Name of the event being triggered.

Type str**label**

Label associated with the event.

Type str**Returns** Returns the string representation of the label when the event type is queried.**Return type** Label**name**

label

__str__(*self*)

class InvManage.views.**ObjectModel**

Bases: django.db.models.Model

Model associated with an object.

name

Name of the object.

Type str

label

Label string of the object.

Type str

modName

Name of the model associated with the object.

Type str

Returns Returns string representation of the label when object model is queried.

Return type Label

name

label

modName

__str__(*self*)

class InvManage.views.**HistoryFilterState**

Bases: django.db.models.Model

Model of filter state of the history.

params

Paramameters of the configured filters.

Type str

params

class InvManage.views.**ShippingAddress**

Bases: django.db.models.Model

Model of the shipping address.

title

Title of the firm.

Type str

name

Name of the firm.

Type str

phone

Contact number of the firm.

Type str

address

Postal address.

Type str**city**

City.

Type str**state**

State.

Type str**country**

Country.

Type str**website**

Official website of the firm.

Type URLField**post**

Postal code of the firm.

Type str**title****name****phone****address****city****state****country****website****post****class** InvManage.views.**PurchaseData**

Bases: django.db.models.Model

Model of the purchase data.

currency

Currency of purchase.

Type str**minorder**

Minimum order quantity.

Type int**contactperson**

Name of the contact person.

Type str

refcode

Reference code.

Type str**transportmode**

Mode of transport of the shipment.

Type str**currency****minorder****contactperson****refcode****transportmode****class** InvManage.views.**Communication**

Bases: django.db.models.Model

Model of the communication information.

language

Language of communication.

Type str**phone**

Contact number of the firm.

Type str**email**

E-mail of the contact person.

Type EmailField**fax**

Fax number.

Type str**language****phone****email****fax****class** InvManage.views.**BankAccount**

Bases: django.db.models.Model

Model of the bank account details.

name

Name of the bank.

Type str**branch**

Branch number of the bank.

Type str

region

City in which the branch is located.

Type str

route

Transit number.

Type str

number

Bank account number.

Type int

accttype

Type of bank account.

Type str

iban

IBAN number.

Type str

code

Bank code.

Type str

branchcode

Branch code.

Type str

name**branch****region****route****number****accttype****iban****code****branchcode**

`InvManage.views.get_columns(state)`

Gets the columns of the display table.

This function takes a filter state and returns an array of column names that are ordered depending on the position attribute of the columns. This order is important as the queryset will be rendered in the display table according to the column order in this returned array.

Parameters `state` (`FilterState`) – Filter state of the columns.

Returns A list that dictates the order of the columns.

Return type List

`InvManage.views.sort_ascending_descending(request, model)`

Sorts and saves the state of the column.

This function takes a request, finds which column needs to be sorted in ascending/descending order, and returns the sorted queryset.

Parameters

- **request** (*HttpRequest*) – GET request received for retrieving the list of model instances.
- **model** (*Model*) – Model of the objects being retrieved.

Returns List of sorted object instances.

Return type *QuerySet*

`InvManage.views.change_column_position(request, state)`

Modifies the column states depending on the request.

This function:

- Takes a filter state
- Extracts the column names into an order array
- Modifies the column order in the array depending on user input direction (left/right)
- Saves this modified state of columns into the database
- Returns the modified column names array

Parameters

- **request** (*HttpRequest*) – GET request for retrieving the state of the columns.
- **state** (*FilterState*) – Filter state of the columns.

Returns List of sorted columns.

Return type *List*

`InvManage.views.paginate(queryset, filter, page_number)`

Paginates the list of objects.

This function takes the entire queryset and filters out only objects belonging to the request page

Parameters

- **queryset** (*QuerySet*) – *QuerySet* of the objects retrieved from the database.
- **filter** (*FilterState*) – Filter state of the columns.
- **page_number** (*int*) – Page number requested.

class `InvManage.views.EventCard`

Bases: `django.db.models.Model`

Model for event card to be displayed in the history view.

obj

Object associated with the event.

Type *OneToOneField*

objId

Unique identifier of the object.

Type *str*

objname
Name of the object.
Type str

objmodel
Model of the object.
Type str

date
Timestamp of the event.
Type DateField

operation
Type of event.
Type str

obj
objId
objname
objmodel
date
operation

class InvManage.views.**HistoryFilterState**

Bases: django.db.models.Model

Model of filter state of the history.

params

Paramameters of the configured filters.

Type str

params

class InvManage.views.**EventCardFilter**

Bases: django_filters.FilterSet

class Meta

model

fields

context

MODEL_CHOICES = [['Company', 'Company'], ['Vendor', 'Vendor'], ['PurchaseOrder', 'Purchase Order'], ['Product', ...

OPERATION_CHOICES = [['Created', 'Created'], ['Updated', 'Updated'], ['Deleted', 'Deleted']]

objname

objmodel

date__gt

date__lt

operation

class InvManage.views.HistoryFilterStateSerializer

Bases: rest_framework.serializers.ModelSerializer

Serializer for HistoryFilterState class instance.

The HistoryFilterStateSerializer.data attribute gives the JSON serialized data of the HistoryFilterState instance:

```
{
  "name": PreRelease20x,
  "numEntries" : 10,
  "eventTypes": {
    "Created",
    "Updated",
    "Deleted"
  },
  "objModels": {
    "Company",
    "Vendor",
    "PurchaseOrder",
    "Product",
    "Consumer",
    "SalesOrder",
    "GoodsReceiptNote"
  }
}
```

class Meta

model

fields = ['name', 'numEntries', 'eventTypes', 'objModels']

events

objModels

update(self, instance, validated_data)

class InvManage.views.HistoryForm

Bases: django.forms.Form

Form for history view.

qlen

Count of the events visible on the history view.

Type int

prefix = history

context

qlen

InvManage.views.display_history_view(request)

Retrieves the list of events on GET request. The create, update, and delete events are registered for each model.

GET /history/

Gets the list of all history items.

Example request:

```
GET /history/ HTTP/1.1
Host: localhost:8000
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9

[
  {
    "objname": {
      ""
    },
    "operation": {
      "Created",
      "Updated",
      "Deleted"
    },
    "objmodel": {
      "Company",
      "Vendor",
      "PurchaseOrder",
      "Product",
      "Consumer",
      "SalesOrder",
      "GoodsReceiptNote"
    },
    "history-qlen": {
      "10"
    },
    "date__gt": {
      "11/01/2020"
    },
    "date__lt": {
      "09/26/2021"
    }
  }
]
```

Example response:

```
HTTP/1.1 200 OK
Vary: Accept
Content-Type: text/html; charset=utf-8
```

Request Headers

- **Accept** – text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9

Status Codes

- **200 OK** – List of filtered history events received successfully.

3.1.2 Submodules

3.1.2.1 InvManage.filters

3.1.2.1.1 Module Contents

3.1.2.1.1.1 Classes

ProductFilter

VendorFilter

ConsumerFilter

PurchaseOrderFilter

GoodsReceiptNoteFilter

SalesOrderFilter

CompanyFilter

EventCardFilter

```
class InvManage.filters.ProductFilter
```

```
    Bases: django_filters.FilterSet
```

```
    class Meta
```

```
        model
```

```
        fields
```

```
    context
```

```
    name
```

```
    category
```

```
    quantity
```

```
    price
```

```
    identifier
```

```
    location
```

```
    order_by_field = ordering
```

```
    ordering
```

```
class InvManage.filters.VendorFilter
```

```
    Bases: django_filters.FilterSet
```

```
    class Meta
```



```
        model
        fields
    context
    name
    identifier
    phone
    email
    location
    order_by_field = ordering
    ordering

class InvManage.filters.ConsumerFilter
    Bases: django_filters.FilterSet
    class Meta

        model
        fields
    context
    name
    identifier
    phone
    email
    location
    order_by_field = ordering
    ordering

class InvManage.filters.PurchaseOrderFilter
    Bases: django_filters.FilterSet
    class Meta

        model
        fields
    context
    vendor
    date
    po
    ordertotal
    order_by_field = ordering
    ordering
```

```
class InvManage.filters.GoodsReceiptNoteFilter
    Bases: django_filters.FilterSet

    class Meta

        model
        fields

    context
    identifier
    vendor
    date
    poRef
    grnType
    amendNumber
    amendDate
    vehicleNumber
    gateInwardNumber
    preparedBy
    checkedBy
    inspectedBy
    approvedBy
    ordering

class InvManage.filters.SalesOrderFilter
    Bases: django_filters.FilterSet

    class Meta

        model
        fields

    context
    consumer
    date
    so
    ordertotal
    order_by_field = ordering
    ordering

class InvManage.filters.CompanyFilter
    Bases: django_filters.FilterSet

    class Meta
```

```

        model
        fields
    context
    name
    owner
    phone
    email
    location
    order_by_field = ordering
    ordering

class InvManage.filters.EventCardFilter
    Bases: django_filters.FilterSet
    class Meta

        model
        fields

    context

    MODEL_CHOICES = [['Company', 'Company'], ['Vendor', 'Vendor'], ['PurchaseOrder',
        'Purchase Order'], ['Product', ...

    OPERATION_CHOICES = [['Created', 'Created'], ['Updated', 'Updated'], ['Deleted',
        'Deleted']]

    objname
    objmodel
    date__gt
    date__lt
    operation

```

3.1.2.2 InvManage.serializers

3.1.2.2.1 Module Contents

3.1.2.2.1.1 Classes

<i>ShippingAddressSerializer</i>	Serializer for ShippingAddress class instance.
<i>CommunicationSerializer</i>	Serializer for Communication class instance.
<i>ProductSerializer</i>	Serializer for Product class instance.
<i>VendorSerializer</i>	Serializer for Vendor class instance.
<i>ConsumerSerializer</i>	Serializer for Consumer class instance.
<i>PPEntrySerializer</i>	Serializer for ProductPurchaseEntry class instance.
<i>PurchaseOrderSerializer</i>	Serializer for PurchaseOrder class instance.

continues on next page

Table 21 – continued from previous page

<i>GRNEntrySerializer</i>	Serializer for GRNEntry class instance.
<i>GRNEntryWithPORefSerializer</i>	Serializer for GRNEntry class instance if the GRN references a purchase order.
<i>GoodsReceiptNoteSerializer</i>	Serializer for GoodsReceiptNote class instance.
<i>PSEntrySerializer</i>	Serializer for ProductSalesEntry class instance.
<i>SalesOrderSerializer</i>	Serializer for SalesOrder class instance.
<i>CompanySerializer</i>	Serializer for Company class instance.
<i>PurchaseInvoiceSerializer</i>	Serializer for PurchaseInvoice class instance.
<i>GRNInvoiceSerializer</i>	Serializer for GRNInvoice class instance.
<i>SalesInvoiceSerializer</i>	Serializer for SalesInvoice class instance.
<i>EventTypeSerializer</i>	Serializer for EventType class instance.
<i>ObjectModelSerializer</i>	Serializer for ObjectModel class instance.
<i>HistoryFilterStateSerializer</i>	Serializer for HistoryFilterState class instance.

class InvManage.serializers.ShippingAddressSerializer

Bases: rest_framework.serializers.ModelSerializer

Serializer for ShippingAddress class instance.

The ShippingAddressSerializer.data attribute gives the JSON serialized data of the ShippingAddress instance:

```
{
  "name": "Harding Gross",
  "address": "8798 At, St., 7639",
  "city": "Rome",
  "phone": "936 651-4847",
  "state": "Lazio",
  "country": "Italy",
  "post": "300326"
}
```

class Meta**model**

fields = ['name', 'address', 'city', 'phone', 'state', 'country', 'post']

class InvManage.serializers.CommunicationSerializer

Bases: rest_framework.serializers.ModelSerializer

Serializer for Communication class instance.

The CommunicationSerializer.data attribute gives the JSON serialized data of the Communication instance:

```
{
  "email": "JohnDoe@domain.com",
  "phone": "(332) 654 6432"
}
```

class Meta**model**

```
fields = ['email', 'phone']
```

```
class InvManage.serializers.ProductSerializer
```

```
Bases: rest_framework.serializers.ModelSerializer
```

Serializer for Product class instance.

The ProductSerializer.data attribute gives the JSON serialized data of the Product instance:

```
{
  "pk": 637,
  "name": "Piano",
  "category": "Ultricies PC",
  "quantity": 23921,
  "identifier": "PROD9",
  "location": "Musselburgh",
  "description": "88-key, Tri-sensor Scaled Hammer Action Keyboard II, Simulated_
  ↪ebony and ivory keys ",
  "prod_id": 637
}
```

```
class Meta
```

```
model
```

```
fields = ['pk', 'name', 'category', 'quantity', 'identifier', 'location',
  'description', 'prod_id']
```

```
prod_id
```

```
to_representation(self, instance)
```

```
class InvManage.serializers.VendorSerializer
```

```
Bases: rest_framework.serializers.ModelSerializer
```

Serializer for Vendor class instance.

The VendorSerializer.data attribute gives the JSON serialized data of the Vendor instance:

```
{
  "name": "Harding Gross",
  "identifier": "HG",
  "gstn": "89ACC16843543",
  "address": {
    "name": "Harding Gross",
    "address": "8798 At, St., 7639",
    "city": "Rome",
    "phone": "936 651-4847",
    "state": "Lazio",
    "country": "Italy",
    "post": "300326"
  }
}
```

```
class Meta
```

```
model
```

```
fields = ['name', 'identifier', 'gstn', 'address']
```

```
    phone
    email
    address
    to_representation(self, instance)
```

```
class InvManage.serializers.ConsumerSerializer
```

```
    Bases: rest_framework.serializers.ModelSerializer
```

```
    Serializer for Consumer class instance.
```

```
    The ConsumerSerializer.data attribute gives the JSON serialized data of the Consumer instance:
```

```
{
    "name": "The Music Store",
    "identifier": "CONS1256",
    "gstIn": "89AAC4633353643",
    "phone": "+91 8325642358",
    "address": "Plot no 958, N- 4, Neo Complex, Barh, Wokha, Nagaland, 797111",
    "email": "JohnDoe@themusic.store",
    "location": "Wokha"
}
```

```
class Meta
```

```
    model
```

```
    fields = ['name', 'identifier', 'gstIn', 'phone', 'address', 'email',
              'location']
```

```
    to_representation(self, instance)
```

```
class InvManage.serializers.PPEntrySerializer
```

```
    Bases: rest_framework.serializers.ModelSerializer
```

```
    Serializer for ProductPurchaseEntry class instance.
```

```
    The PPEntrySerializer.data attribute gives the JSON serialized data of the ProductPurchaseEntry instance:
```

```
{
    "ppe_id": 324,
    "product": {
        "pk": 637,
        "name": "Piano",
        "category": "Ultricies PC",
        "quantity": 23921,
        "identifier": "PROD9",
        "location": "Musselburgh",
        "description": "88-key, Tri-sensor Scaled Hammer Action Keyboard II, ↵
↳ Simulated ebony and ivory keys ",
        "prod_id": 637
    },
    "quantity": 100,
    "price": 10.0,
    "discount": 0.0,
}
```

(continues on next page)

(continued from previous page)

```

    "order": 182,
    "pendingQty": 50
}

```

class Meta

model

fields = ['ppe_id', 'product', 'quantity', 'price', 'discount', 'order',
'pendingQty']

ppe_id

product

pendingQty

to_representation(self, instance)

create(self, validated_data)

update(self, instance, validated_data)

class InvManage.serializers.**PurchaseOrderSerializer**

Bases: rest_framework.serializers.ModelSerializer

Serializer for PurchaseOrder class instance.

The PurchaseOrderSerializer.data attribute gives the JSON serialized data of the PurchaseOrder instance:

```

{
  "vendor": {
    "name": "Girish",
    "identifier": "GJ",
    "gstin": "GSTIN002",
    "address": {
      "name": "alsf",
      "address": "jas;k",
      "city": ";sdalkf",
      "phone": "alsf",
      "state": "kjdfk",
      "country": "ljflkj",
      "post": "54545"
    }
  },
  "date": "25-Sep-2021",
  "po": 293,
  "subtotal": 279975.0,
  "taxtotal": 22398.0,
  "ordertotal": 302373.0,
  "ppes": [
    {
      "ppe_id": 324,
      "product": {
        "pk": 637,

```

(continues on next page)

(continued from previous page)

```

        "name": "piano",
        "category": "Ultricies PC",
        "quantity": 23921,
        "identifier": "PROD9",
        "location": "Musselburgh",
        "description": "88-key, Tri-sensor Scaled Hammer Action Keyboard II,
→ Simulated ebony and ivory keys ",
        "prod_id": 637
    },
    "quantity": 100,
    "price": 10.0,
    "discount": 0.0,
    "order": 182,
    "pendingQty": 50
},
{
    "ppe_id": 325,
    "product": {
        "pk": 645,
        "name": "Tabl",
        "category": "Sociis Natoque Company",
        "quantity": 38276,
        "identifier": "PROD17",
        "location": "Schagen",
        "description": "aldgjlakjlkasdj",
        "prod_id": 645
    },
    "quantity": 250,
    "price": 90.0,
    "discount": 8.0,
    "order": 182,
    "pendingQty": 70
},
{
    "ppe_id": 326,
    "product": {
        "pk": 638,
        "name": "Goblet drum",
        "category": "Est Congue Consulting",
        "quantity": 46076,
        "identifier": "PROD10",
        "location": "Kaluga",
        "description": "aldgjlakjlkasdj",
        "prod_id": 638
    },
    "quantity": 200,
    "price": 150.0,
    "discount": 12.0,
    "order": 182,
    "pendingQty": 200
}
}
]

```

(continues on next page)

(continued from previous page)

}

class Meta**model****fields** = ['vendor', 'date', 'po', 'subtotal', 'taxtotal', 'ordertotal', 'ppes']**vendor****ppes****date****to_representation**(self, instance)**class** InvManage.serializers.GRNEntrySerializer

Bases: rest_framework.serializers.ModelSerializer

Serializer for GRNEntry class instance.

The GRNEntrySerializer.data attribute gives the JSON serialized data of the GRNEntry instance:

```
{
  "grn": 103,
  "grne_id": 117,
  "product": {
    "pk": 637,
    "name": "piano",
    "category": "Ultricies PC",
    "quantity": 23921,
    "identifier": "PROD9",
    "location": "Musselburgh",
    "description": "88-key, Tri-sensor Scaled Hammer Action Keyboard II, ↵
↳ Simulated ebony and ivory keys ",
    "prod_id": 637
  },
  "quantity": 100,
  "receivedQty": 50,
  "acceptedQty": 50,
  "rejectedQty": 0,
  "remark": "OK"
}
```

class Meta**model****fields** = ['grn', 'grne_id', 'product', 'quantity', 'receivedQty', 'acceptedQty', 'rejectedQty', 'remark']**grne_id****product****grn****to_representation**(self, instance)

```
create(self, validated_data)
```

```
update(self, instance, validated_data)
```

```
class InvManage.serializers.GRNEntryWithPORefSerializer
```

```
Bases: rest_framework.serializers.ModelSerializer
```

Serializer for GRNEntry class instance if the GRN references a purchase order.

The GRNEntryWithPORefSerializer.data attribute gives the JSON serialized data of the GRNEntry instance:

```
{
  "grn": 103,
  "grne_id": 117,
  "ppe_id": 324,
  "po_id": 293,
  "product": {
    "pk": 637,
    "name": "piano",
    "category": "Ultricies PC",
    "quantity": 23921,
    "identifier": "PROD9",
    "location": "Musselburgh",
    "description": "88-key, Tri-sensor Scaled Hammer Action Keyboard II, ↵
↳ Simulated ebony and ivory keys ",
    "prod_id": 637
  },
  "quantity": 100,
  "receivedQty": 50,
  "acceptedQty": 50,
  "rejectedQty": 0,
  "remark": "OK"
}
```

```
class Meta
```

```
model
```

```
fields = ['grn', 'grne_id', 'ppe_id', 'po_id', 'product', 'quantity',
'receivedQty', 'acceptedQty', ...]
```

```
grne_id
```

```
ppe_id
```

```
po_id
```

```
product
```

```
grn
```

```
to_representation(self, instance)
```

```
create(self, validated_data)
```

```
update(self, instance, validated_data)
```

```
class InvManage.serializers.GoodsReceiptNoteSerializer
```

```
Bases: rest_framework.serializers.ModelSerializer
```

Serializer for GoodsReceiptNote class instance.

The `GoodsReceiptNoteSerializer.data` attribute gives the JSON serialized data of the `GoodsReceiptNote` instance:

```
{
  "grnes": [
    {
      "grn": 103,
      "grne_id": 117,
      "ppe_id": 324,
      "po_id": 293,
      "product": {
        "pk": 637,
        "name": "piano",
        "category": "Ultricies PC",
        "quantity": 23921,
        "identifier": "PROD9",
        "location": "Musselburgh",
        "description": "88-key, Tri-sensor Scaled Hammer Action Keyboard II,
→ Simulated ebony and ivory keys ",
        "prod_id": 637
      },
      "quantity": 100,
      "receivedQty": 50,
      "acceptedQty": 50,
      "rejectedQty": 0,
      "remark": "OK"
    },
    {
      "grn": 103,
      "grne_id": 118,
      "ppe_id": 325,
      "po_id": 293,
      "product": {
        "pk": 645,
        "name": "Tabl",
        "category": "Sociis Natoque Company",
        "quantity": 38276,
        "identifier": "PROD17",
        "location": "Schagen",
        "description": "aldgjlakjlkasdj",
        "prod_id": 645
      },
      "quantity": 250,
      "receivedQty": 200,
      "acceptedQty": 180,
      "rejectedQty": 20,
      "remark": "20 pieces faulty"
    }
  ],
  "date": "29-Sep-2021",
  "vendor": {
    "name": "Girish",
    "identifier": "GJ",

```

(continues on next page)

(continued from previous page)

```

        "gstIn": "GSTIN002",
        "address": {
            "name": "alsf",
            "address": "jas;k",
            "city": ";sdalkf",
            "phone": "alsf",
            "state": "kjdfk",
            "country": "ljflkj",
            "post": "54545"
        },
        "poRef": [
            182
        ],
        "identifier": 846,
        "grnType": "auto",
        "amendDate": "2021-09-29T00:00:00Z",
        "transporter": "TeraTransport",
        "vehicleNumber": "GH-646358",
        "gateInwardNumber": "864353",
        "preparedBy": "KJL",
        "checkedBy": "KJH",
        "inspectedBy": "GIO",
        "approvedBy": "BHI"
    }
}

```

class Meta

model

fields = ['grnes', 'date', 'vendor', 'poRef', 'identifier', 'grnType',
'amendDate', 'transporter',...]

grnes

date

vendor

to_representation(self, instance)

get_grnes(self, instance)

class InvManage.serializers.PSEntrySerializer

Bases: rest_framework.serializers.ModelSerializer

Serializer for ProductSalesEntry class instance.

The PSEntrySerializer.data attribute gives the JSON serialized data of the ProductSalesEntry instance:

```

{
    "pse_id": 64,
    "product": {
        "pk": 637,
        "name": "Piano",
        "category": "Ultricies PC",
    }
}

```

(continues on next page)

(continued from previous page)

```

        "quantity": 23921,
        "identifier": "PROD9",
        "location": "Musselburgh",
        "description": "88-key, Tri-sensor Scaled Hammer Action Keyboard II, ↵
↵Simulated ebony and ivory keys ",
        "prod_id": 637
    },
    "quantity": 50,
    "price": 9900.0,
    "discount": 10.0,
    "order": 29
}

```

class Meta

model

fields = ['pse_id', 'product', 'quantity', 'price', 'discount', 'order']

pse_id

product

to_representation(self, instance)

create(self, validated_data)

update(self, instance, validated_data)

class InvManage.serializers.SalesOrderSerializer

Bases: rest_framework.serializers.ModelSerializer

Serializer for SalesOrder class instance.

The SalesOrderSerializer.data attribute gives the JSON serialized data of the SalesOrder instance:

```

{
  "consumer": {
    "name": "The Music Store",
    "identifier": "CONS1256",
    "gstn": "89AAC4633353643",
    "phone": "+91 8325642358",
    "address": "Plot no 958, N- 4, Neo Complex, Barh, Wokha, Nagaland, 797111",
    "email": "JohnDoe@themusic.store",
    "location": "Wokha"
  },
  "date": "2021-09-25T00:00:00Z",
  "so": 89,
  "subtotal": 744900.0,
  "taxtotal": 59592.0,
  "ordertotal": 804492.0,
  "pses": [
    {
      "pse_id": 64,
      "product": {
        "pk": 637,

```

(continues on next page)

(continued from previous page)

```

        "name": "piano",
        "category": "Ultricies PC",
        "quantity": 23921,
        "identifier": "PROD9",
        "location": "Musselburgh",
        "description": "88-key, Tri-sensor Scaled Hammer Action Keyboard II,
→ Simulated ebony and ivory keys ",
        "prod_id": 637
    },
    "quantity": 50,
    "price": 9900.0,
    "discount": 10.0,
    "order": 29
},
{
    "pse_id": 65,
    "product": {
        "pk": 649,
        "name": "Sabar",
        "category": "Amet Consulting",
        "quantity": 3903,
        "identifier": "PROD21",
        "location": "Serang",
        "description": "High Sierra Sabar, Travel bag, Blue, Grey, Zipper,
→ 36.5 L, 51.5 cm, 26 cm ",
        "prod_id": 649
    },
    "quantity": 25,
    "price": 4800.0,
    "discount": 8.0,
    "order": 29
},
{
    "pse_id": 66,
    "product": {
        "pk": 654,
        "name": "Parai",
        "category": "Aliquet Lobortis Ltd",
        "quantity": 8534,
        "identifier": "PROD26",
        "location": "Burhaniye",
        "description": "Kannan musical instruments Parai 15" inch,
→ (Baffallow skin) Daf Instrument",
        "prod_id": 654
    },
    "quantity": 35,
    "price": 6000.0,
    "discount": 10.0,
    "order": 29
}
}
]
}

```

```
class Meta
```

```
    model
```

```
    fields = ['consumer', 'date', 'so', 'subtotal', 'taxtotal', 'ordertotal',
              'pses']
```

```
    consumer
```

```
    pses
```

```
    to_representation(self, instance)
```

```
class InvManage.serializers.CompanySerializer
```

```
    Bases: rest_framework.serializers.ModelSerializer
```

```
    Serializer for Company class instance.
```

The `CompanySerializer.data` attribute gives the JSON serialized data of the `Company` instance:

```
{
    "name": "Fringillami",
    "owner": "Ivor Barnett",
    "gstln": "89AAC056465468",
    "phone": "332 220-7026",
    "address": "Ap #849-6241 Euismod Av., 677598, Carinthia, Belgium",
    "email": "est.tempor@fringillami.org",
    "location": "Belgium",
    "image": "/media/images/logo.png"
}
```

```
class Meta
```

```
    model
```

```
    fields = ['name', 'owner', 'gstln', 'phone', 'address', 'email', 'location',
              'image']
```

```
class InvManage.serializers.PurchaseInvoiceSerializer
```

```
    Bases: rest_framework.serializers.ModelSerializer
```

```
    Serializer for PurchaseInvoice class instance.
```

The `PurchaseInvoiceSerializer.data` attribute gives the JSON serialized data of the `PurchaseInvoice` instance:

```
{
    "company": {
        "name": "Fringillami",
        "owner": "Ivor Barnett",
        "gstln": "89AAC056465468",
        "phone": "332 220-7026",
        "address": "Ap #849-6241 Euismod Av., 677598, Carinthia, Belgium",
        "email": "est.tempor@fringillami.org",
        "location": "Belgium",
        "image": "/media/images/logo.png"
    },
    "po": {
```

(continues on next page)

(continued from previous page)

```

    "vendor": {
      "name": "Girish",
      "identifier": "GJ",
      "gstn": "GSTIN002",
      "address": {
        "name": "alsf",
        "address": "jas;k",
        "city": ";sdalkf",
        "phone": "alsf",
        "state": "kjdfk",
        "country": "ljflkj",
        "post": "54545"
      }
    },
    "date": "25-Sep-2021",
    "po": 293,
    "subtotal": 279975.0,
    "taxtotal": 22398.0,
    "ordertotal": 302373.0,
    "ppes": [
      {
        "ppe_id": 324,
        "product": {
          "pk": 637,
          "name": "piano",
          "category": "Ultricies PC",
          "quantity": 23921,
          "identifier": "PROD9",
          "location": "Musselburgh",
          "description": "88-key, Tri-sensor Scaled Hammer Action,
↳ Keyboard II, Simulated ebony and ivory keys ",
          "prod_id": 637
        },
        "quantity": 100,
        "price": 10.0,
        "discount": 0.0,
        "order": 182,
        "pendingQty": 50
      },
      {
        "ppe_id": 325,
        "product": {
          "pk": 645,
          "name": "Tabl",
          "category": "Sociis Natoque Company",
          "quantity": 38276,
          "identifier": "PROD17",
          "location": "Schagen",
          "description": "aldgjlakjlkasdj",
          "prod_id": 645
        },
        "quantity": 250,

```

(continues on next page)

(continued from previous page)

```

    "price": 90.0,
    "discount": 8.0,
    "order": 182,
    "pendingQty": 70
  },
  {
    "ppe_id": 326,
    "product": {
      "pk": 638,
      "name": "Goblet drum",
      "category": "Est Congue Consulting",
      "quantity": 46076,
      "identifier": "PROD10",
      "location": "Kaluga",
      "description": "aldgjlakjlkasdj",
      "prod_id": 638
    },
    "quantity": 200,
    "price": 150.0,
    "discount": 12.0,
    "order": 182,
    "pendingQty": 200
  },
  {
    "ppe_id": 327,
    "product": {
      "pk": 643,
      "name": "quinto",
      "category": "Enim Suspendisse Associates",
      "quantity": 51099,
      "identifier": "PROD15",
      "location": "Nagpur",
      "description": "aldgjlakjlkasdj",
      "prod_id": 643
    },
    "quantity": 350,
    "price": 150.0,
    "discount": 10.0,
    "order": 182,
    "pendingQty": 350
  },
  {
    "ppe_id": 328,
    "product": {
      "pk": 651,
      "name": "Igihumurizo",
      "category": "Curabitur Massa Vestibulum Consulting",
      "quantity": 48553,
      "identifier": "PROD23",
      "location": "Yeovil",
      "description": "aldgjlakjlkasdj",
      "prod_id": 651
    }
  }

```

(continues on next page)

(continued from previous page)

```

        },
        "quantity": 500,
        "price": 50.0,
        "discount": 6.0,
        "order": 182,
        "pendingQty": 500
    },
    {
        "ppe_id": 329,
        "product": {
            "pk": 653,
            "name": "Balsié",
            "category": "Eu Foundation",
            "quantity": 29988,
            "identifier": "PROD25",
            "location": "Sudbury",
            "description": "aldgjlakjlkasdj",
            "prod_id": 653
        },
        "quantity": 200,
        "price": 500.0,
        "discount": 15.0,
        "order": 182,
        "pendingQty": 200
    },
    {
        "ppe_id": 330,
        "product": {
            "pk": 656,
            "name": "Padayani thappu",
            "category": "Posuere LLP",
            "quantity": 6358,
            "identifier": "PROD28",
            "location": "Ruvo del Monte",
            "description": "aldgjlakjlkasdj",
            "prod_id": 656
        },
        "quantity": 350,
        "price": 250.0,
        "discount": 13.0,
        "order": 182,
        "pendingQty": 350
    }
]
},
"shippingaddress": {
    "name": "Harding Gross",
    "address": "8798 At, St., 7639",
    "city": "Rome",
    "phone": "936 651-4847",
    "state": "Lazio",
    "country": "Italy",

```

(continues on next page)

(continued from previous page)

```

        "post": "300326"
    },
    "communication": {
        "email": "asfs@aflk.com",
        "phone": "6546432"
    }
}

```

class Meta

model

fields = ['company', 'po', 'shippingaddress', 'communication']

company

po

shippingaddress

communication

class InvManage.serializers.GRNInvoiceSerializer

Bases: rest_framework.serializers.ModelSerializer

Serializer for GRNInvoice class instance.

The GRNInvoiceSerializer.data attribute gives the JSON serialized data of the GRNInvoice instance:

```

{
  "company": {
    "name": "Fringillami",
    "owner": "Ivor Barnett",
    "gstn": "89AAC056465468",
    "phone": "332 220-7026",
    "address": "Ap #849-6241 Euismod Av., 677598, Carinthia, Belgium",
    "email": "est.tempor@fringillami.org",
    "location": "Belgium",
    "image": "/media/images/logo.png"
  },
  "grn": {
    "grnes": [
      {
        "grn": 103,
        "grne_id": 117,
        "ppe_id": 324,
        "po_id": 293,
        "product": {
          "pk": 637,
          "name": "piano",
          "category": "Ultricies PC",
          "quantity": 23921,
          "identifier": "PROD9",
          "location": "Musselburgh",
          "description": "88-key, Tri-sensor Scaled Hammer Action,
↳Keyboard II, Simulated ebony and ivory keys ",

```

(continues on next page)

(continued from previous page)

```

        "prod_id": 637
    },
    "quantity": 100,
    "receivedQty": 50,
    "acceptedQty": 50,
    "rejectedQty": 0,
    "remark": "OK"
},
{
    "grn": 103,
    "grne_id": 118,
    "ppe_id": 325,
    "po_id": 293,
    "product": {
        "pk": 645,
        "name": "Tabl",
        "category": "Sociis Natoque Company",
        "quantity": 38276,
        "identifier": "PROD17",
        "location": "Schagen",
        "description": "aldgjlakjlkasdj",
        "prod_id": 645
    },
    "quantity": 250,
    "receivedQty": 200,
    "acceptedQty": 180,
    "rejectedQty": 20,
    "remark": "20 pieces faulty"
}
],
"date": "29-Sep-2021",
"vendor": {
    "name": "Girish",
    "identifier": "GJ",
    "gstIn": "GSTIN002",
    "address": {
        "name": "alsf",
        "address": "jas;k",
        "city": ";sdalkf",
        "phone": "alsf",
        "state": "kjdfk",
        "country": "ljflkj",
        "post": "54545"
    }
},
"poRef": [
    182
],
"identifier": 846,
"grnType": "auto",
"amendDate": "2021-09-29T00:00:00Z",
"transporter": "TeraTransport",

```

(continues on next page)

(continued from previous page)

```

        "vehicleNumber": "GH-646358",
        "gateInwardNumber": "864353",
        "preparedBy": "KJL",
        "checkedBy": "KJH",
        "inspectedBy": "GIO",
        "approvedBy": "BHI"
    },
    "shippingaddress": {
        "name": "Harding Gross",
        "address": "8798 At, St., 7639",
        "city": "Rome",
        "phone": "936 651-4847",
        "state": "Lazio",
        "country": "Italy",
        "post": "300326"
    },
    "communication": {
        "email": "asfs@aflk.com",
        "phone": "6546432"
    }
}

```

class Meta

model

fields = ['company', 'grn', 'shippingaddress', 'communication']

company

grn

shippingaddress

communication

class InvManage.serializers.SalesInvoiceSerializer

Bases: rest_framework.serializers.ModelSerializer

Serializer for SalesInvoice class instance.

The SalesInvoiceSerializer.data attribute gives the JSON serialized data of the SalesInvoice instance:

```

{
    "company": {
        "name": "Fringillami",
        "owner": "Ivor Barnett",
        "gstIn": "89AAC056465468",
        "phone": "332 220-7026",
        "address": "Ap #849-6241 Euismod Av., 677598, Carinthia, Belgium",
        "email": "est.tempor@fringillami.org",
        "location": "Belgium",
        "image": "/media/images/hyperlink-green_x91WW5n.png"
    },
    "so": {
        "consumer": {

```

(continues on next page)

(continued from previous page)

```

        "name": "The Music Store",
        "identifier": "CONS1256",
        "gstn": "89AAC4633353643",
        "phone": "+91 8325642358",
        "address": "Plot no 958, N- 4, Neo Complex, Barh, Wokha, Nagaland, ↵
↵797111",
        "email": "JohnDoe@themusic.store",
        "location": "Wokha"
    },
    "date": "2021-09-25T00:00:00Z",
    "so": 89,
    "subtotal": 744900.0,
    "taxtotal": 59592.0,
    "ordertotal": 804492.0,
    "pses": [
        {
            "pse_id": 64,
            "product": {
                "pk": 637,
                "name": "piano",
                "category": "Ultricies PC",
                "quantity": 23921,
                "identifier": "PROD9",
                "location": "Musselburgh",
                "description": "88-key, Tri-sensor Scaled Hammer Action ↵
↵Keyboard II, Simulated ebony and ivory keys ",
                "prod_id": 637
            },
            "quantity": 50,
            "price": 9900.0,
            "discount": 10.0,
            "order": 29
        },
        {
            "pse_id": 65,
            "product": {
                "pk": 649,
                "name": "Sabar",
                "category": "Amet Consulting",
                "quantity": 3903,
                "identifier": "PROD21",
                "location": "Serang",
                "description": "High Sierra Sabar, Travel bag, Blue, Grey, ↵
↵Zipper, 36.5 L, 51.5 cm, 26 cm ",
                "prod_id": 649
            },
            "quantity": 25,
            "price": 4800.0,
            "discount": 8.0,
            "order": 29
        },
        {

```

(continues on next page)

(continued from previous page)

```

        "pse_id": 66,
        "product": {
            "pk": 654,
            "name": "Parai",
            "category": "Aliquet Lobortis Ltd",
            "quantity": 8534,
            "identifier": "PROD26",
            "location": "Burhaniye",
            "description": "Kannan musical instruments Parai 15" inch_
→ (Baffallow skin) Daf Instrument",
            "prod_id": 654
        },
        "quantity": 35,
        "price": 6000.0,
        "discount": 10.0,
        "order": 29
    }
]
},
"shippingaddress": {
    "name": "Harding Gross",
    "address": "8798 At, St., 7639",
    "city": "Rome",
    "phone": "936 651-4847",
    "state": "Lazio",
    "country": "Italy",
    "post": "300326"
}
}

```

```
class Meta
```

```
    model
```

```
    fields = ['company', 'so', 'shippingaddress']
```

```
company
```

```
so
```

```
shippingaddress
```

```
class InvManage.serializers.EventTypeSerializer
```

```
    Bases: rest_framework.serializers.ModelSerializer
```

```
    Serializer for EventType class instance.
```

```
    The EventTypeSerializer.data attribute gives the JSON serialized data of the EventType instance:
```

```

{
    "name": "# 85",
    "label": "Purchase Order"
}

```

```
class Meta
```

```
    model
    fields = ['created', 'updated', 'deleted']
```

```
class InvManage.serializers.ObjectModelSerializer
    Bases: rest_framework.serializers.ModelSerializer
```

Serializer for ObjectModel class instance.

The ObjectModelSerializer.data attribute gives the JSON serialized data of the ObjectModel instance:

```
{
    "name": "# 85",
    "label": "Purchase Order",
    "modName": "PurchaseOrder"
}
```

```
class Meta
```

```
    model
    fields = ['company', 'vendor', 'po', 'product', 'consumer', 'so']
```

```
class InvManage.serializers.HistoryFilterStateSerializer
    Bases: rest_framework.serializers.ModelSerializer
```

Serializer for HistoryFilterState class instance.

The HistoryFilterStateSerializer.data attribute gives the JSON serialized data of the HistoryFilterState instance:

```
{
    "name": PreRelease20x,
    "numEntries" : 10,
    "eventTypes": {
        "Created",
        "Updated",
        "Deleted"
    },
    "objModels": {
        "Company",
        "Vendor",
        "PurchaseOrder",
        "Product",
        "Consumer",
        "SalesOrder",
        "GoodsReceiptNote"
    }
}
```

```
class Meta
```

```
    model
    fields = ['name', 'numEntries', 'eventTypes', 'objModels']
```

```
    events
```

```
    objModels
```


update(*self*, *instance*, *validated_data*)

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`

PYTHON MODULE INDEX

i

- InvManage, 7
- InvManage.filters, 436
- InvManage.forms, 7
- InvManage.forms.forms, 7
- InvManage.forms.vendor_forms, 18
- InvManage.models, 37
- InvManage.models.misc, 37
- InvManage.models.objects, 39
- InvManage.models.reuse, 52
- InvManage.models.states, 56
- InvManage.scripts, 75
- InvManage.scripts.filters, 75
- InvManage.serializers, 439
- InvManage.views, 76
- InvManage.views.company_views, 76
- InvManage.views.consumer_views, 80
- InvManage.views.goods_receipt_note_views, 84
- InvManage.views.history_views, 92
- InvManage.views.product_views, 94
- InvManage.views.purchase_order_views, 98
- InvManage.views.sales_order_views, 110
- InvManage.views.vendor_views, 118

HTTP ROUTING TABLE

/companies

GET /companies/, 277

/company

GET /company, 274

GET /company/update, 275

POST /company, 275

POST /company/<str:object_id>/delete, 277

POST /company/update, 276

/consumer

GET /consumer, 351

GET /consumer/update, 352

POST /consumer, 351

POST /consumer/<str:object_id>/delete, 353

POST /consumer/update, 353

/consumers

GET /consumers/, 354

/get_consumer

GET /get_consumer/, 354

/get_vendor

GET /get_vendor/, 122

/grn

GET /grn, 182

GET /grn/update, 184

POST /grn, 182

POST /grn/<str:grn_id>/print, 187

POST /grn/<str:object_id>/delete, 186

POST /grn/update, 184

/grns

GET /grns/, 186

/history

GET /history/, 434

/product

GET /product, 94

GET /product/update, 95

POST /product, 94

POST /product/<str:product_id>/delete, 97

POST /product/update, 96

/product_purchase_entries

GET /product_purchase_entries/, 107

/products

GET /products/, 97

/purchaes_order

GET /purchaes_order/update, 100

/purchase_order

GET /purchase_order, 98

POST /purchase_order, 99

POST /purchase_order/<str:object_id>/delete, 102

POST /purchase_order/<str:po_id>/print, 103

POST /purchase_order/<str:so_id>/print, 115

POST /purchase_order/update, 101

/purchase_orders

GET /purchase_orders/, 102

/sales_order

GET /sales_order, 110

GET /sales_order/update, 112

POST /sales_order, 111

POST /sales_order/<str:object_id>/delete, 114

POST /sales_order/update, 113

/sales_orders

GET /sales_orders/, 115

/upload

POST /upload/<data>/, 98

/vendor

GET /vendor, 118

GET /vendor/update, 120

POST /vendor, 119

POST /vendor/<str:object_id>/delete, 122

POST /vendor/update, 120

/vendors

GET /vendors/, 122

Symbols

__str__() (InvManage.models.Company method), 68
 __str__() (InvManage.models.Consumer method), 66
 __str__() (InvManage.models.EventType method), 71
 __str__() (InvManage.models.ObjectModel method), 72
 __str__() (InvManage.models.Product method), 59
 __str__() (InvManage.models.Vendor method), 67
 __str__() (InvManage.models.misc.EventType method), 38
 __str__() (InvManage.models.misc.ObjectModel method), 38
 __str__() (InvManage.models.objects.Company method), 41
 __str__() (InvManage.models.objects.Consumer method), 43
 __str__() (InvManage.models.objects.Product method), 48
 __str__() (InvManage.models.objects.Vendor method), 42
 __str__() (InvManage.views.Company method), 162, 215, 265, 303, 342, 380, 424
 __str__() (InvManage.views.Consumer method), 160, 213, 263, 301, 340, 378, 422
 __str__() (InvManage.views.EventType method), 165, 218, 268, 306, 345, 383, 428
 __str__() (InvManage.views.ObjectModel method), 166, 218, 269, 307, 346, 384, 428
 __str__() (InvManage.views.Product method), 153, 205, 256, 294, 333, 371, 415
 __str__() (InvManage.views.Vendor method), 161, 214, 264, 302, 341, 379, 423

A

acceptedQty (InvManage.forms.forms.GRNEntryForm attribute), 15, 16
 acceptedQty (InvManage.forms.GRNEntryForm attribute), 29, 30
 acceptedQty (InvManage.models.GRNEntry attribute), 60, 61
 acceptedQty (InvManage.models.objects.GRNEntry attribute), 50
 acceptedQty (InvManage.models.objects.ProductPurchaseEntry attribute), 49
 acceptedQty (InvManage.models.ProductPurchaseEntry attribute), 59, 60
 acceptedQty (InvManage.views.GRNEntry attribute), 154, 155, 207, 257, 258, 295, 296, 335, 372, 373, 417
 acceptedQty (InvManage.views.GRNEntryForm attribute), 144, 196, 197, 247, 284, 285, 324, 362, 406
 acceptedQty (InvManage.views.ProductPurchaseEntry attribute), 153, 154, 206, 256, 257, 294, 295, 334, 371, 372, 416
 acctype (InvManage.forms.BankAccountForm attribute), 36, 37
 acctype (InvManage.forms.vendor_forms.BankAccountForm attribute), 21, 22
 acctype (InvManage.models.BankAccount attribute), 74, 75
 acctype (InvManage.models.reuse.BankAccount attribute), 55
 acctype (InvManage.views.BankAccount attribute), 168, 169, 221, 271, 272, 309, 310, 349, 386, 387, 431
 acctype (InvManage.views.BankAccountForm attribute), 150, 151, 203, 204, 253, 254, 291, 292, 331, 369, 413
 address (InvManage.forms.CompanyForm attribute), 32, 33
 address (InvManage.forms.ConsumerForm attribute), 31, 32
 address (InvManage.forms.forms.CompanyForm attribute), 17, 18
 address (InvManage.forms.forms.ConsumerForm attribute), 14, 15
 address (InvManage.forms.ShippingAddressForm attribute), 33, 34
 address (InvManage.forms.vendor_forms.ShippingAddressForm attribute), 19, 20
 address (InvManage.models.Company attribute), 67, 68
 address (InvManage.models.Consumer attribute), 66

address (*InvManage.models.objects.Company* attribute), 40, 41
 address (*InvManage.models.objects.Consumer* attribute), 42, 43
 address (*InvManage.models.objects.Vendor* attribute), 41, 42
 address (*InvManage.models.reuse.ShippingAddress* attribute), 53
 address (*InvManage.models.ShippingAddress* attribute), 72, 73
 address (*InvManage.models.Vendor* attribute), 66, 67
 address (*InvManage.serializers.VendorSerializer* attribute), 442
 address (*InvManage.views.Company* attribute), 161, 162, 214, 215, 264, 265, 302, 303, 342, 379, 380, 424
 address (*InvManage.views.CompanyForm* attribute), 148, 201, 251, 252, 289, 328, 329, 366, 367, 410, 411
 address (*InvManage.views.Consumer* attribute), 159, 160, 212, 213, 263, 300, 301, 340, 378, 422
 address (*InvManage.views.ConsumerForm* attribute), 147, 148, 200, 250, 251, 288, 289, 328, 365, 366, 410
 address (*InvManage.views.ShippingAddress* attribute), 166, 167, 219, 269, 270, 307, 308, 346, 347, 384, 385, 428, 429
 address (*InvManage.views.ShippingAddressForm* attribute), 146, 198, 199, 249, 287, 326, 327, 364, 408, 409
 address (*InvManage.views.Vendor* attribute), 160, 161, 213, 214, 263, 264, 301, 302, 341, 379, 423
 address (*InvManage.views.VendorSerializer* attribute), 312
 amendDate (*InvManage.filters.GoodsReceiptNoteFilter* attribute), 438
 amendDate (*InvManage.forms.forms.GRNInfo* attribute), 12
 amendDate (*InvManage.forms.GRNInfo* attribute), 28, 29
 amendDate (*InvManage.models.GoodsReceiptNote* attribute), 64, 65
 amendDate (*InvManage.models.objects.GoodsReceiptNote* attribute), 44, 45
 amendDate (*InvManage.views.GoodsReceiptNote* attribute), 158, 159, 211, 212, 261, 262, 299, 300, 338, 339, 376, 377, 421
 amendDate (*InvManage.views.GoodsReceiptNoteFilter* attribute), 170
 amendDate (*InvManage.views.GRNInfo* attribute), 142, 143, 195, 245, 246, 283, 284, 322, 323, 360, 404, 405
 amendNumber (*InvManage.models.objects.Company* attribute), 40, 41
 amendNumber (*InvManage.forms.forms.GRNInfo* attribute), 11, 12
 amendNumber (*InvManage.forms.GRNInfo* attribute), 28, 29
 amendNumber (*InvManage.models.GoodsReceiptNote* attribute), 64, 65
 amendNumber (*InvManage.models.objects.GoodsReceiptNote* attribute), 44, 45
 amendNumber (*InvManage.views.GoodsReceiptNote* attribute), 158, 159, 211, 212, 261, 262, 299, 300, 338, 339, 376, 377, 420, 421
 amendNumber (*InvManage.views.GoodsReceiptNoteFilter* attribute), 170
 amendNumber (*InvManage.views.GRNInfo* attribute), 142, 143, 194, 195, 245, 246, 283, 284, 322, 323, 360, 361, 404, 405
 approvedBy (*InvManage.filters.GoodsReceiptNoteFilter* attribute), 438
 approvedBy (*InvManage.forms.forms.GRNInfo* attribute), 12, 13
 approvedBy (*InvManage.forms.GRNInfo* attribute), 28, 29
 approvedBy (*InvManage.models.GoodsReceiptNote* attribute), 65
 approvedBy (*InvManage.models.objects.GoodsReceiptNote* attribute), 45
 approvedBy (*InvManage.views.GoodsReceiptNote* attribute), 159, 211, 212, 262, 300, 339, 377, 421
 approvedBy (*InvManage.views.GoodsReceiptNoteFilter* attribute), 170
 approvedBy (*InvManage.views.GRNInfo* attribute), 142, 143, 195, 196, 245, 246, 283, 284, 323, 361, 405

B

backaccount (*InvManage.models.objects.Vendor* attribute), 41
 backaccount (*InvManage.models.Vendor* attribute), 67
 backaccount (*InvManage.views.Vendor* attribute), 160, 213, 264, 301, 341, 379, 423
 balance (*InvManage.forms.forms.PurchaseOrderBasicInfo* attribute), 10, 11
 balance (*InvManage.forms.forms.SalesOrderBasicInfo* attribute), 13, 14
 balance (*InvManage.forms.PurchaseOrderBasicInfo* attribute), 26, 27
 balance (*InvManage.forms.SalesOrderBasicInfo* attribute), 30, 31
 balance (*InvManage.models.objects.PurchaseOrder* attribute), 43, 44
 balance (*InvManage.models.objects.SalesOrder* attribute), 46

- balance (*InvManage.models.PurchaseOrder* attribute), 63
 - balance (*InvManage.models.SalesOrder* attribute), 62
 - balance (*InvManage.views.PurchaseOrder* attribute), 157, 210, 260, 298, 337, 338, 375, 376, 419, 420
 - balance (*InvManage.views.PurchaseOrderBasicInfo* attribute), 141, 193, 194, 244, 282, 321, 322, 359, 403, 404
 - balance (*InvManage.views.SalesOrder* attribute), 156, 209, 259, 297, 336, 337, 374, 418, 419
 - balance (*InvManage.views.SalesOrderBasicInfo* attribute), 145, 197, 198, 248, 286, 325, 326, 363, 407, 408
 - BankAccount (class in *InvManage.models*), 74
 - BankAccount (class in *InvManage.models.reuse*), 54
 - BankAccount (class in *InvManage.views*), 168, 221, 271, 309, 348, 386, 430
 - bankaccount (*InvManage.models.objects.Vendor* attribute), 42
 - bankaccount (*InvManage.models.Vendor* attribute), 67
 - bankaccount (*InvManage.views.Vendor* attribute), 161, 214, 264, 302, 341, 379, 423
 - BankAccountForm (class in *InvManage.forms*), 36
 - BankAccountForm (class in *InvManage.forms.vendor_forms*), 21
 - BankAccountForm (class in *InvManage.views*), 150, 203, 253, 291, 330, 368, 412
 - barcode (*InvManage.forms.forms.ProductStorageInfoForm* attribute), 9
 - barcode (*InvManage.forms.ProductStorageInfoForm* attribute), 26
 - barcode (*InvManage.models.objects.Product* attribute), 48
 - barcode (*InvManage.models.Product* attribute), 58, 59
 - barcode (*InvManage.views.Product* attribute), 152, 153, 205, 255, 256, 293, 294, 332, 333, 370, 371, 414, 415
 - barcode (*InvManage.views.ProductStorageInfoForm* attribute), 140, 193, 243, 281, 320, 358, 402
 - branch (*InvManage.forms.BankAccountForm* attribute), 36, 37
 - branch (*InvManage.forms.vendor_forms.BankAccountForm* attribute), 21, 22
 - branch (*InvManage.models.BankAccount* attribute), 74, 75
 - branch (*InvManage.models.reuse.BankAccount* attribute), 55
 - branch (*InvManage.views.BankAccount* attribute), 168, 169, 221, 271, 272, 309, 310, 348, 349, 386, 387, 430, 431
 - branch (*InvManage.views.BankAccountForm* attribute), 150, 151, 203, 253, 254, 291, 292, 330, 331, 368, 369, 413
 - branchcode (*InvManage.forms.BankAccountForm* attribute), 36, 37
 - branchcode (*InvManage.forms.vendor_forms.BankAccountForm* attribute), 22
 - branchcode (*InvManage.models.BankAccount* attribute), 75
 - branchcode (*InvManage.models.reuse.BankAccount* attribute), 55
 - branchcode (*InvManage.views.BankAccount* attribute), 168, 169, 221, 222, 271, 272, 309, 310, 349, 387, 431
 - branchcode (*InvManage.views.BankAccountForm* attribute), 151, 203, 204, 254, 292, 331, 369, 413
- ## C
- category (*InvManage.filters.ProductFilter* attribute), 436
 - category (*InvManage.forms.forms.ProductBasicInfoForm* attribute), 8
 - category (*InvManage.forms.ProductBasicInfoForm* attribute), 23
 - category (*InvManage.models.objects.Product* attribute), 47, 48
 - category (*InvManage.models.Product* attribute), 57, 58
 - category (*InvManage.views.Product* attribute), 151, 152, 204, 205, 254, 255, 292, 293, 331, 333, 369, 370, 414, 415
 - category (*InvManage.views.ProductBasicInfoForm* attribute), 137, 190, 240, 278, 317, 355, 399, 400
 - category (*InvManage.views.ProductFilter* attribute), 133
 - change_column_position() (in module *InvManage.scripts.filters*), 76
 - change_column_position() (in module *InvManage.views*), 169, 222, 272, 310, 350, 387, 432
 - checkedBy (*InvManage.filters.GoodsReceiptNoteFilter* attribute), 438
 - checkedBy (*InvManage.forms.forms.GRNInfo* attribute), 12, 13
 - checkedBy (*InvManage.forms.GRNInfo* attribute), 28, 29
 - checkedBy (*InvManage.models.GoodsReceiptNote* attribute), 65
 - checkedBy (*InvManage.models.objects.GoodsReceiptNote* attribute), 45
 - checkedBy (*InvManage.views.GoodsReceiptNote* attribute), 158, 159, 211, 212, 262, 299, 300, 339, 377, 421
 - checkedBy (*InvManage.views.GoodsReceiptNoteFilter* attribute), 170
 - checkedBy (*InvManage.views.GRNInfo* attribute), 142, 143, 195, 196, 245, 246, 283, 284, 323, 360, 361, 405

[city](#) (*InvManage.forms.ShippingAddressForm* attribute), 33, 34
[city](#) (*InvManage.forms.vendor_forms.ShippingAddressForm* attribute), 19, 20
[city](#) (*InvManage.models.reuse.ShippingAddress* attribute), 53
[city](#) (*InvManage.models.ShippingAddress* attribute), 72, 73
[city](#) (*InvManage.views.ShippingAddress* attribute), 166, 167, 219, 269, 270, 307, 308, 347, 384, 385, 429
[city](#) (*InvManage.views.ShippingAddressForm* attribute), 146, 198, 199, 249, 287, 326, 327, 364, 408, 409
[code](#) (*InvManage.forms.BankAccountForm* attribute), 36, 37
[code](#) (*InvManage.forms.vendor_forms.BankAccountForm* attribute), 21, 22
[code](#) (*InvManage.models.BankAccount* attribute), 74, 75
[code](#) (*InvManage.models.reuse.BankAccount* attribute), 55
[code](#) (*InvManage.views.BankAccount* attribute), 168, 169, 221, 222, 271, 272, 309, 310, 349, 387, 431
[code](#) (*InvManage.views.BankAccountForm* attribute), 151, 203, 204, 254, 291, 292, 331, 369, 413
[Communication](#) (class in *InvManage.models*), 73
[Communication](#) (class in *InvManage.models.reuse*), 53
[Communication](#) (class in *InvManage.views*), 167, 220, 270, 308, 348, 386, 430
[communication](#) (*InvManage.models.objects.GRNInvoice* attribute), 52
[communication](#) (*InvManage.models.objects.PurchaseInvoice* attribute), 51
[communication](#) (*InvManage.models.objects.Vendor* attribute), 41, 42
[communication](#) (*InvManage.models.PurchaseInvoice* attribute), 68, 69
[communication](#) (*InvManage.models.Vendor* attribute), 66, 67
[communication](#) (*InvManage.serializers.GRNInvoiceSerializer* attribute), 457
[communication](#) (*InvManage.serializers.PurchaseInvoiceSerializer* attribute), 455
[communication](#) (*InvManage.views.GRNInvoice* attribute), 182
[communication](#) (*InvManage.views.GRNInvoiceSerializer* attribute), 174
[communication](#) (*InvManage.views.PurchaseInvoice* attribute), 162, 215, 265, 266, 303, 343, 381, 425
[communication](#) (*InvManage.views.PurchaseInvoiceSerializer* attribute), 181, 228
[communication](#) (*InvManage.views.Vendor* attribute), 160, 161, 213, 214, 263, 264, 301, 302, 341, 379, 423
[CommunicationForm](#) (class in *InvManage.forms*), 35
[CommunicationForm](#) (class in *InvManage.forms.vendor_forms*), 20
[CommunicationForm](#) (class in *InvManage.views*), 149, 201, 252, 290, 329, 367, 411
[CommunicationSerializer](#) (class in *InvManage.serializers*), 440
[CommunicationSerializer.Meta](#) (class in *InvManage.serializers*), 440
[Company](#) (class in *InvManage.models*), 67
[Company](#) (class in *InvManage.models.objects*), 40
[Company](#) (class in *InvManage.views*), 161, 214, 264, 302, 341, 379, 423
[company](#) (*InvManage.models.objects.GRNInvoice* attribute), 51, 52
[company](#) (*InvManage.models.objects.PurchaseInvoice* attribute), 51
[company](#) (*InvManage.models.objects.SalesInvoice* attribute), 52
[company](#) (*InvManage.models.PurchaseInvoice* attribute), 68
[company](#) (*InvManage.models.SalesInvoice* attribute), 69
[company](#) (*InvManage.serializers.GRNInvoiceSerializer* attribute), 457
[company](#) (*InvManage.serializers.PurchaseInvoiceSerializer* attribute), 455
[company](#) (*InvManage.serializers.SalesInvoiceSerializer* attribute), 459
[company](#) (*InvManage.views.GRNInvoice* attribute), 181, 182
[company](#) (*InvManage.views.GRNInvoiceSerializer* attribute), 174
[company](#) (*InvManage.views.PurchaseInvoice* attribute), 162, 215, 265, 303, 342, 343, 380, 381, 424, 425
[company](#) (*InvManage.views.PurchaseInvoiceSerializer* attribute), 181, 228
[company](#) (*InvManage.views.SalesInvoice* attribute), 163, 215, 216, 266, 304, 343, 381, 425
[company](#) (*InvManage.views.SalesInvoiceSerializer* attribute), 392
[CompanyFilter](#) (class in *InvManage.filters*), 438
[CompanyFilter](#) (class in *InvManage.views*), 274
[CompanyFilter.Meta](#) (class in *InvManage.filters*), 438
[CompanyFilter.Meta](#) (class in *InvManage.views*), 274
[CompanyForm](#) (class in *InvManage.forms*), 32
[CompanyForm](#) (class in *InvManage.forms.forms*), 17
[CompanyForm](#) (class in *InvManage.views*), 148, 200, 251, 289, 328, 366, 410

CompanySerializer (class in *InvManage.serializers*), 451
 CompanySerializer (class in *InvManage.views*), 273
 CompanySerializer.Meta (class in *InvManage.serializers*), 451
 CompanySerializer.Meta (class in *InvManage.views*), 273
 Consumer (class in *InvManage.models*), 65
 Consumer (class in *InvManage.models.objects*), 42
 Consumer (class in *InvManage.views*), 159, 212, 262, 300, 339, 377, 422
 consumer (*InvManage.filters.SalesOrderFilter* attribute), 438
 consumer (*InvManage.forms.forms.SalesOrderBasicInfo* attribute), 13, 14
 consumer (*InvManage.forms.SalesOrderBasicInfo* attribute), 30, 31
 consumer (*InvManage.models.objects.SalesOrder* attribute), 45, 46
 consumer (*InvManage.models.SalesOrder* attribute), 61, 62
 consumer (*InvManage.serializers.SalesOrderSerializer* attribute), 451
 consumer (*InvManage.views.SalesOrder* attribute), 155, 156, 208, 209, 258, 259, 296, 297, 336, 374, 418
 consumer (*InvManage.views.SalesOrderBasicInfo* attribute), 144, 145, 197, 198, 247, 248, 285, 286, 325, 362, 363, 407
 consumer (*InvManage.views.SalesOrderFilter* attribute), 388
 consumer_choices (*InvManage.forms.forms.SalesOrderBasicInfo* attribute), 14
 consumer_choices (*InvManage.forms.SalesOrderBasicInfo* attribute), 31
 consumer_choices (*InvManage.views.SalesOrderBasicInfo* attribute), 145, 198, 248, 286, 325, 363, 407
 ConsumerFilter (class in *InvManage.filters*), 437
 ConsumerFilter (class in *InvManage.views*), 350
 ConsumerFilter.Meta (class in *InvManage.filters*), 437
 ConsumerFilter.Meta (class in *InvManage.views*), 350
 ConsumerForm (class in *InvManage.forms*), 31
 ConsumerForm (class in *InvManage.forms.forms*), 14
 ConsumerForm (class in *InvManage.views*), 147, 200, 250, 288, 327, 365, 409
 ConsumerSerializer (class in *InvManage.serializers*), 442
 ConsumerSerializer (class in *InvManage.views*), 351
 ConsumerSerializer.Meta (class in *InvManage.serializers*), 442
 ConsumerSerializer.Meta (class in *InvManage.views*), 351
 contactperson (*InvManage.forms.PurchaseDataForm* attribute), 35, 36
 contactperson (*InvManage.forms.vendor_forms.PurchaseDataForm* attribute), 20, 21
 contactperson (*InvManage.models.PurchaseData* attribute), 73, 74
 contactperson (*InvManage.models.reuse.PurchaseData* attribute), 54
 contactperson (*InvManage.views.PurchaseData* attribute), 167, 220, 270, 308, 347, 348, 385, 429, 430
 contactperson (*InvManage.views.PurchaseDataForm* attribute), 150, 202, 203, 253, 290, 291, 330, 368, 412
 context (*InvManage.filters.CompanyFilter* attribute), 439
 context (*InvManage.filters.ConsumerFilter* attribute), 437
 context (*InvManage.filters.EventCardFilter* attribute), 439
 context (*InvManage.filters.GoodsReceiptNoteFilter* attribute), 438
 context (*InvManage.filters.ProductFilter* attribute), 436
 context (*InvManage.filters.PurchaseOrderFilter* attribute), 437
 context (*InvManage.filters.SalesOrderFilter* attribute), 438
 context (*InvManage.filters.VendorFilter* attribute), 437
 context (*InvManage.forms.BankAccountForm* attribute), 36
 context (*InvManage.forms.CommunicationForm* attribute), 35
 context (*InvManage.forms.CompanyForm* attribute), 33
 context (*InvManage.forms.ConsumerForm* attribute), 32
 context (*InvManage.forms.forms.CompanyForm* attribute), 18
 context (*InvManage.forms.forms.ConsumerForm* attribute), 14
 context (*InvManage.forms.forms.GRNEntryForm* attribute), 15
 context (*InvManage.forms.forms.GRNInfo* attribute), 12
 context (*InvManage.forms.forms.HistoryForm* attribute), 18
 context (*InvManage.forms.forms.ProductBasicInfoForm* attribute), 8
 context (*InvManage.forms.forms.ProductDetailedInfoForm* attribute), 8
 context (*InvManage.forms.forms.ProductPricingForm* attribute), 9
 context (*InvManage.forms.forms.ProductPurchaseEntryForm* attribute), 9

- attribute), 16
- context (InvManage.forms.forms.ProductSalesEntryForm attribute), 17
- context (InvManage.forms.forms.ProductStatusForm attribute), 10
- context (InvManage.forms.forms.ProductStorageInfoForm attribute), 9
- context (InvManage.forms.forms.PurchaseOrderBasicInfo attribute), 11
- context (InvManage.forms.forms.SalesOrderBasicInfo attribute), 14
- context (InvManage.forms.GRNEntryForm attribute), 29
- context (InvManage.forms.GRNInfo attribute), 28
- context (InvManage.forms.HistoryForm attribute), 33
- context (InvManage.forms.ProductBasicInfoForm attribute), 23
- context (InvManage.forms.ProductDetailedInfoForm attribute), 23
- context (InvManage.forms.ProductPricingForm attribute), 24
- context (InvManage.forms.ProductPurchaseEntryForm attribute), 24
- context (InvManage.forms.ProductSalesEntryForm attribute), 25
- context (InvManage.forms.ProductStatusForm attribute), 25
- context (InvManage.forms.ProductStorageInfoForm attribute), 26
- context (InvManage.forms.PurchaseDataForm attribute), 36
- context (InvManage.forms.PurchaseOrderBasicInfo attribute), 27
- context (InvManage.forms.SalesOrderBasicInfo attribute), 31
- context (InvManage.forms.ShippingAddressForm attribute), 34
- context (InvManage.forms.vendor_forms.BankAccountForm attribute), 22
- context (InvManage.forms.vendor_forms.CommunicationForm attribute), 20
- context (InvManage.forms.vendor_forms.PurchaseDataForm attribute), 21
- context (InvManage.forms.vendor_forms.ShippingAddressForm attribute), 19
- context (InvManage.forms.vendor_forms.VendorForm attribute), 19
- context (InvManage.forms.VendorForm attribute), 34
- context (InvManage.views.BankAccountForm attribute), 151, 203, 254, 292, 331, 369, 413
- context (InvManage.views.CommunicationForm attribute), 149, 202, 252, 290, 330, 367, 412
- context (InvManage.views.CompanyFilter attribute), 274
- context (InvManage.views.CompanyForm attribute), 148, 201, 251, 289, 329, 367, 411
- context (InvManage.views.ConsumerFilter attribute), 350
- context (InvManage.views.ConsumerForm attribute), 147, 200, 251, 288, 328, 366, 410
- context (InvManage.views.EventCardFilter attribute), 433
- context (InvManage.views.GoodsReceiptNoteFilter attribute), 170
- context (InvManage.views.GRNEntryForm attribute), 144, 196, 247, 285, 324, 362, 406
- context (InvManage.views.GRNInfo attribute), 142, 195, 246, 283, 323, 361, 405
- context (InvManage.views.HistoryForm attribute), 149, 201, 252, 290, 329, 367, 411, 434
- context (InvManage.views.ProductBasicInfoForm attribute), 137, 190, 240, 278, 317, 355, 400
- context (InvManage.views.ProductDetailedInfoForm attribute), 138, 190, 241, 278, 318, 356, 400
- context (InvManage.views.ProductFilter attribute), 133
- context (InvManage.views.ProductPricingForm attribute), 138, 191, 241, 279, 318, 356, 400
- context (InvManage.views.ProductPurchaseEntryForm attribute), 139, 191, 242, 279, 319, 357, 401
- context (InvManage.views.ProductSalesEntryForm attribute), 139, 192, 242, 280, 320, 357, 402
- context (InvManage.views.ProductStatusForm attribute), 140, 192, 243, 281, 320, 358, 402
- context (InvManage.views.ProductStorageInfoForm attribute), 140, 193, 243, 281, 320, 358, 402
- context (InvManage.views.PurchaseDataForm attribute), 150, 202, 253, 291, 330, 368, 412
- context (InvManage.views.PurchaseOrderBasicInfo attribute), 141, 194, 244, 282, 321, 359, 403
- context (InvManage.views.PurchaseOrderFilter attribute), 223
- context (InvManage.views.SalesOrderBasicInfo attribute), 145, 198, 248, 286, 325, 363, 407
- context (InvManage.views.SalesOrderFilter attribute), 388
- context (InvManage.views.ShippingAddressForm attribute), 146, 199, 249, 287, 326, 364, 409
- context (InvManage.views.VendorFilter attribute), 311
- context (InvManage.views.VendorForm attribute), 147, 199, 250, 288, 327, 365, 409
- country (InvManage.forms.ShippingAddressForm attribute), 34
- country (InvManage.forms.vendor_forms.ShippingAddressForm attribute), 19, 20
- country (InvManage.models.reuse.ShippingAddress attribute), 53
- country (InvManage.models.ShippingAddress attribute), 72, 73

country (*InvManage.views.ShippingAddress* attribute), 166, 167, 219, 269, 270, 307, 308, 347, 384, 385, 429

country (*InvManage.views.ShippingAddressForm* attribute), 146, 199, 249, 287, 326, 327, 364, 365, 408, 409

create() (*InvManage.serializers.GRNEntrySerializer* method), 445

create() (*InvManage.serializers.GRNEntryWithPORefSerializer* method), 446

create() (*InvManage.serializers.PPEntrySerializer* method), 443

create() (*InvManage.serializers.PSEntrySerializer* method), 449

create() (*InvManage.views.GRNEntrySerializer* method), 171

create() (*InvManage.views.GRNEntryWithPORefSerializer* method), 172

create() (*InvManage.views.PPEntrySerializer* method), 177, 224

create() (*InvManage.views.PSEntrySerializer* method), 390

create_company_view() (in module *InvManage.views*), 274

create_company_view() (in module *InvManage.views.company_views*), 77

create_consumer_view() (in module *InvManage.views*), 351

create_consumer_view() (in module *InvManage.views.consumer_views*), 80

create_grn_view() (in module *InvManage.views*), 182

create_grn_view() (in module *InvManage.views.goods_receipt_note_views*), 85

create_product_view() (in module *InvManage.views*), 133

create_product_view() (in module *InvManage.views.product_views*), 94

create_purchase_order_view() (in module *InvManage.views*), 228

create_purchase_order_view() (in module *InvManage.views.purchase_order_views*), 98

create_sales_order_view() (in module *InvManage.views*), 392

create_sales_order_view() (in module *InvManage.views.sales_order_views*), 110

create_vendor_view() (in module *InvManage.views*), 312

create_vendor_view() (in module *InvManage.views.vendor_views*), 118

currency (*InvManage.forms.PurchaseDataForm* attribute), 35, 36

currency (*InvManage.forms.vendor_forms.PurchaseDataForm* attribute), 20, 21

currency (*InvManage.models.PurchaseData* attribute), 73, 74

currency (*InvManage.models.reuse.PurchaseData* attribute), 54

currency (*InvManage.views.PurchaseData* attribute), 167, 220, 270, 308, 347, 348, 385, 429, 430

currency (*InvManage.views.PurchaseDataForm* attribute), 149, 150, 202, 203, 252, 253, 290, 291, 330, 368, 412

D

Dashboard (class in *InvManage.models.objects*), 40

date (*InvManage.filters.GoodsReceiptNoteFilter* attribute), 438

date (*InvManage.filters.PurchaseOrderFilter* attribute), 437

date (*InvManage.filters.SalesOrderFilter* attribute), 438

date (*InvManage.forms.forms.GRNInfo* attribute), 11, 12

date (*InvManage.forms.forms.PurchaseOrderBasicInfo* attribute), 10, 11

date (*InvManage.forms.forms.SalesOrderBasicInfo* attribute), 13, 14

date (*InvManage.forms.GRNInfo* attribute), 27, 28

date (*InvManage.forms.PurchaseOrderBasicInfo* attribute), 26, 27

date (*InvManage.forms.SalesOrderBasicInfo* attribute), 30, 31

date (*InvManage.models.EventCard* attribute), 70, 71

date (*InvManage.models.GoodsReceiptNote* attribute), 64, 65

date (*InvManage.models.misc.EventCard* attribute), 39

date (*InvManage.models.objects.GoodsReceiptNote* attribute), 44, 45

date (*InvManage.models.objects.PurchaseOrder* attribute), 43, 44

date (*InvManage.models.objects.SalesOrder* attribute), 46

date (*InvManage.models.PurchaseOrder* attribute), 63

date (*InvManage.models.SalesOrder* attribute), 61, 62

date (*InvManage.serializers.GoodsReceiptNoteSerializer* attribute), 448

date (*InvManage.serializers.PurchaseOrderSerializer* attribute), 445

date (*InvManage.views.EventCard* attribute), 164, 165, 217, 267, 268, 305, 306, 345, 383, 427, 433

date (*InvManage.views.GoodsReceiptNote* attribute), 158, 159, 211, 212, 261, 262, 299, 300, 338, 339, 376, 377, 420, 421

date (*InvManage.views.GoodsReceiptNoteFilter* attribute), 170

date (*InvManage.views.GoodsReceiptNoteSerializer* attribute), 176

date (*InvManage.views.GRNInfo* attribute), 142, 143, 194, 195, 245, 246, 283, 284, 322, 323, 360, 361, 404, 405

date (*InvManage.views.PurchaseOrder* attribute), 156, 157, 209, 210, 260, 297, 298, 337, 338, 375, 376, 419, 420
date (*InvManage.views.PurchaseOrderBasicInfo* attribute), 140, 141, 193, 194, 243, 244, 281, 282, 321, 358, 359, 403
date (*InvManage.views.PurchaseOrderFilter* attribute), 223
date (*InvManage.views.SalesOrder* attribute), 155, 156, 208, 209, 258, 259, 296, 297, 336, 374, 418
date (*InvManage.views.SalesOrderBasicInfo* attribute), 144, 145, 197, 198, 247, 248, 285, 286, 325, 362, 363, 407, 408
date (*InvManage.views.SalesOrderFilter* attribute), 388
date__gt (*InvManage.filters.EventCardFilter* attribute), 439
date__gt (*InvManage.views.EventCardFilter* attribute), 433
date__lt (*InvManage.filters.EventCardFilter* attribute), 439
date__lt (*InvManage.views.EventCardFilter* attribute), 433
delete_company_view() (in module *InvManage.views*), 277
delete_company_view() (in module *InvManage.views.company_views*), 79
delete_consumer_view() (in module *InvManage.views*), 353
delete_consumer_view() (in module *InvManage.views.consumer_views*), 83
delete_grn_view() (in module *InvManage.views*), 186
delete_grn_view() (in module *InvManage.views.goods_receipt_note_views*), 89
delete_product_view() (in module *InvManage.views*), 135
delete_product_view() (in module *InvManage.views.product_views*), 97
delete_purchase_order_view() (in module *InvManage.views*), 232
delete_purchase_order_view() (in module *InvManage.views.purchase_order_views*), 102
delete_sales_order_view() (in module *InvManage.views*), 396
delete_sales_order_view() (in module *InvManage.views.sales_order_views*), 114
delete_vendor_view() (in module *InvManage.views*), 315
delete_vendor_view() (in module *InvManage.views.vendor_views*), 121
description (*InvManage.forms.forms.ProductBasicInfoForm* attribute), 8
description (*InvManage.forms.ProductBasicInfoForm* attribute), 23
description (*InvManage.models.objects.Product* attribute), 47, 48
description (*InvManage.models.Product* attribute), 57, 58
description (*InvManage.views.Product* attribute), 151, 152, 204, 205, 254, 255, 292, 293, 332, 333, 369, 371, 414, 415
description (*InvManage.views.ProductBasicInfoForm* attribute), 137, 190, 240, 278, 317, 355, 399, 400
discount (*InvManage.forms.forms.ProductPricingForm* attribute), 9
discount (*InvManage.forms.forms.ProductPurchaseEntryForm* attribute), 16
discount (*InvManage.forms.forms.ProductSalesEntryForm* attribute), 17
discount (*InvManage.forms.forms.PurchaseOrderBasicInfo* attribute), 10, 11
discount (*InvManage.forms.forms.SalesOrderBasicInfo* attribute), 13, 14
discount (*InvManage.forms.ProductPricingForm* attribute), 24
discount (*InvManage.forms.ProductPurchaseEntryForm* attribute), 24
discount (*InvManage.forms.ProductSalesEntryForm* attribute), 25
discount (*InvManage.forms.PurchaseOrderBasicInfo* attribute), 26, 27
discount (*InvManage.forms.SalesOrderBasicInfo* attribute), 30, 31
discount (*InvManage.models.objects.Product* attribute), 47, 48
discount (*InvManage.models.objects.ProductPurchaseEntry* attribute), 49
discount (*InvManage.models.objects.ProductSalesEntry* attribute), 50, 51
discount (*InvManage.models.objects.PurchaseOrder* attribute), 43, 44
discount (*InvManage.models.objects.SalesOrder* attribute), 46
discount (*InvManage.models.Product* attribute), 58, 59
discount (*InvManage.models.ProductPurchaseEntry* attribute), 59, 60
discount (*InvManage.models.ProductSalesEntry* attribute), 61
discount (*InvManage.models.PurchaseOrder* attribute), 63
discount (*InvManage.models.SalesOrder* attribute), 62
discount (*InvManage.views.Product* attribute), 152, 153, 205, 255, 256, 293, 294, 332, 333, 370, 371, 414, 415
discount (*InvManage.views.ProductPricingForm* attribute), 138, 191, 241, 279, 318, 356, 400
discount (*InvManage.views.ProductPurchaseEntry* at-

tribute), 153, 154, 206, 256, 257, 294, 295, 333, 334, 371, 372, 415, 416

discount (*InvManage.views.ProductPurchaseEntryForm attribute*), 138, 139, 191, 241, 242, 279, 280, 319, 357, 401

discount (*InvManage.views.ProductSalesEntry attribute*), 155, 208, 258, 296, 335, 336, 373, 417, 418

discount (*InvManage.views.ProductSalesEntryForm attribute*), 139, 192, 242, 280, 319, 320, 357, 358, 402

discount (*InvManage.views.PurchaseOrder attribute*), 157, 209, 210, 260, 298, 337, 338, 375, 376, 419, 420

discount (*InvManage.views.PurchaseOrderBasicInfo attribute*), 140, 141, 193, 194, 243, 244, 281, 282, 321, 359, 403, 404

discount (*InvManage.views.SalesOrder attribute*), 156, 208, 209, 259, 297, 336, 374, 418, 419

discount (*InvManage.views.SalesOrderBasicInfo attribute*), 144, 145, 197, 198, 247, 248, 285, 286, 325, 363, 407, 408

display_companies_view() (in module *InvManage.views*), 277

display_companies_view() (in module *InvManage.views.company_views*), 80

display_consumers_view() (in module *InvManage.views*), 354

display_consumers_view() (in module *InvManage.views.consumer_views*), 83

display_grns_view() (in module *InvManage.views*), 186

display_grns_view() (in module *InvManage.views.goods_receipt_note_views*), 89

display_history_view() (in module *InvManage.views*), 434

display_history_view() (in module *InvManage.views.history_views*), 92

display_products_view() (in module *InvManage.views*), 136

display_products_view() (in module *InvManage.views.product_views*), 97

display_purchase_orders_view() (in module *InvManage.views*), 232

display_purchase_orders_view() (in module *InvManage.views.purchase_order_views*), 102

display_sales_orders_view() (in module *InvManage.views*), 396

display_sales_orders_view() (in module *InvManage.views.sales_order_views*), 115

display_vendors_view() (in module *InvManage.views*), 315

display_vendors_view() (in module *InvManage.views.vendor_views*), 122

E

email (*InvManage.filters.CompanyFilter attribute*), 439

email (*InvManage.filters.ConsumerFilter attribute*), 437

email (*InvManage.filters.VendorFilter attribute*), 437

email (*InvManage.forms.CommunicationForm attribute*), 35

email (*InvManage.forms.CompanyForm attribute*), 32, 33

email (*InvManage.forms.ConsumerForm attribute*), 32

email (*InvManage.forms.forms.CompanyForm attribute*), 17, 18

email (*InvManage.forms.forms.ConsumerForm attribute*), 14, 15

email (*InvManage.forms.vendor_forms.CommunicationForm attribute*), 20

email (*InvManage.models.Communication attribute*), 73

email (*InvManage.models.Company attribute*), 67, 68

email (*InvManage.models.Consumer attribute*), 66

email (*InvManage.models.objects.Company attribute*), 40, 41

email (*InvManage.models.objects.Consumer attribute*), 42, 43

email (*InvManage.models.reuse.Communication attribute*), 54

email (*InvManage.serializers.VendorSerializer.Meta attribute*), 442

email (*InvManage.views.Communication attribute*), 168, 220, 221, 271, 309, 348, 386, 430

email (*InvManage.views.CommunicationForm attribute*), 149, 202, 252, 290, 329, 330, 367, 411, 412

email (*InvManage.views.Company attribute*), 161, 162, 214, 215, 264, 265, 302, 303, 342, 380, 424

email (*InvManage.views.CompanyFilter attribute*), 274

email (*InvManage.views.CompanyForm attribute*), 148, 201, 251, 252, 289, 328, 329, 366, 367, 411

email (*InvManage.views.Consumer attribute*), 160, 212, 213, 263, 301, 340, 378, 422

email (*InvManage.views.ConsumerFilter attribute*), 350

email (*InvManage.views.ConsumerForm attribute*), 147, 148, 200, 250, 251, 288, 289, 328, 365, 366, 410

email (*InvManage.views.VendorFilter attribute*), 311

email (*InvManage.views.VendorSerializer.Meta attribute*), 312

EventCard (class in *InvManage.models*), 70

EventCard (class in *InvManage.models.misc*), 38

EventCard (class in *InvManage.views*), 164, 217, 267, 305, 344, 382, 426, 432

EventCardFilter (class in *InvManage.filters*), 439

EventCardFilter (class in *InvManage.views*), 433

EventCardFilter.Meta (class in *InvManage.filters*), 439

EventCardFilter.Meta (class in *InvManage.views*), 433

- events (*InvManage.serializers.HistoryFilterStateSerializer* attribute), 460
- events (*InvManage.views.HistoryFilterStateSerializer* attribute), 434
- EventType (class in *InvManage.models*), 71
- EventType (class in *InvManage.models.misc*), 37
- EventType (class in *InvManage.views*), 165, 217, 268, 306, 345, 383, 427
- EventTypeSerializer (class in *InvManage.serializers*), 459
- EventTypeSerializer.Meta (class in *InvManage.serializers*), 459
- expiry (*InvManage.forms.forms.ProductStorageInfoForm* attribute), 9
- expiry (*InvManage.forms.ProductStorageInfoForm* attribute), 26
- expiry (*InvManage.models.objects.Product* attribute), 48
- expiry (*InvManage.models.Product* attribute), 58, 59
- expiry (*InvManage.views.Product* attribute), 152, 153, 205, 255, 256, 293, 294, 332, 333, 370, 371, 415
- expiry (*InvManage.views.ProductStorageInfoForm* attribute), 140, 193, 243, 281, 320, 358, 402, 403
- F**
- fax (*InvManage.forms.CommunicationForm* attribute), 35
- fax (*InvManage.forms.vendor_forms.CommunicationForm* attribute), 20
- fax (*InvManage.models.Communication* attribute), 73
- fax (*InvManage.models.reuse.Communication* attribute), 54
- fax (*InvManage.views.Communication* attribute), 168, 220, 221, 271, 309, 348, 386, 430
- fax (*InvManage.views.CommunicationForm* attribute), 149, 202, 252, 290, 329, 330, 367, 411, 412
- fields (*InvManage.filters.CompanyFilter.Meta* attribute), 439
- fields (*InvManage.filters.ConsumerFilter.Meta* attribute), 437
- fields (*InvManage.filters.EventCardFilter.Meta* attribute), 439
- fields (*InvManage.filters.GoodsReceiptNoteFilter.Meta* attribute), 438
- fields (*InvManage.filters.ProductFilter.Meta* attribute), 436
- fields (*InvManage.filters.PurchaseOrderFilter.Meta* attribute), 437
- fields (*InvManage.filters.SalesOrderFilter.Meta* attribute), 438
- fields (*InvManage.filters.VendorFilter.Meta* attribute), 437
- fields (*InvManage.serializers.CommunicationSerializer.Meta* attribute), 440
- fields (*InvManage.serializers.CompanySerializer.Meta* attribute), 451
- fields (*InvManage.serializers.ConsumerSerializer.Meta* attribute), 442
- fields (*InvManage.serializers.EventTypeSerializer.Meta* attribute), 460
- fields (*InvManage.serializers.GoodsReceiptNoteSerializer.Meta* attribute), 448
- fields (*InvManage.serializers.GRNEntrySerializer.Meta* attribute), 445
- fields (*InvManage.serializers.GRNEntryWithPORefSerializer.Meta* attribute), 446
- fields (*InvManage.serializers.GRNInvoiceSerializer.Meta* attribute), 457
- fields (*InvManage.serializers.HistoryFilterStateSerializer.Meta* attribute), 460
- fields (*InvManage.serializers.ObjectModelSerializer.Meta* attribute), 460
- fields (*InvManage.serializers.PPEntrySerializer.Meta* attribute), 443
- fields (*InvManage.serializers.ProductSerializer.Meta* attribute), 441
- fields (*InvManage.serializers.PSEntrySerializer.Meta* attribute), 449
- fields (*InvManage.serializers.PurchaseInvoiceSerializer.Meta* attribute), 455
- fields (*InvManage.serializers.PurchaseOrderSerializer.Meta* attribute), 445
- fields (*InvManage.serializers.SalesInvoiceSerializer.Meta* attribute), 459
- fields (*InvManage.serializers.SalesOrderSerializer.Meta* attribute), 451
- fields (*InvManage.serializers.ShippingAddressSerializer.Meta* attribute), 440
- fields (*InvManage.serializers.VendorSerializer.Meta* attribute), 441
- fields (*InvManage.views.CompanyFilter.Meta* attribute), 274
- fields (*InvManage.views.CompanySerializer.Meta* attribute), 273
- fields (*InvManage.views.ConsumerFilter.Meta* attribute), 350
- fields (*InvManage.views.ConsumerSerializer.Meta* attribute), 351
- fields (*InvManage.views.EventCardFilter.Meta* attribute), 433
- fields (*InvManage.views.GoodsReceiptNoteFilter.Meta* attribute), 170
- fields (*InvManage.views.GoodsReceiptNoteSerializer.Meta* attribute), 176
- fields (*InvManage.views.GRNEntrySerializer.Meta* attribute), 171

fields (*InvManage.views.GRNEntryWithPORefSerializer.Meta attribute*), 172

fields (*InvManage.views.GRNInvoiceSerializer.Meta attribute*), 174

fields (*InvManage.views.HistoryFilterStateSerializer.Meta attribute*), 434

fields (*InvManage.views.PPEntrySerializer.Meta attribute*), 177, 224

fields (*InvManage.views.ProductFilter.Meta attribute*), 132

fields (*InvManage.views.ProductSerializer.Meta attribute*), 177, 223, 389

fields (*InvManage.views.PSEntrySerializer.Meta attribute*), 389

fields (*InvManage.views.PurchaseInvoiceSerializer.Meta attribute*), 181, 228

fields (*InvManage.views.PurchaseOrderFilter.Meta attribute*), 223

fields (*InvManage.views.SalesInvoiceSerializer.Meta attribute*), 392

fields (*InvManage.views.SalesOrderFilter.Meta attribute*), 388

fields (*InvManage.views.ShippingAddressSerializer.Meta attribute*), 274

fields (*InvManage.views.VendorFilter.Meta attribute*), 311

fields (*InvManage.views.VendorSerializer.Meta attribute*), 312

FilterColumn (class in *InvManage.models*), 69

FilterColumn (class in *InvManage.models.states*), 56

FilterColumn (class in *InvManage.views*), 163, 216, 266, 304, 343, 381, 425

FilterState (class in *InvManage.models*), 70

FilterState (class in *InvManage.models.states*), 56

FilterState (class in *InvManage.views*), 163, 216, 266, 304, 343, 381, 425

G

gateInwardNumber (*InvManage.filters.GoodsReceiptNoteFilter attribute*), 438

gateInwardNumber (*InvManage.forms.forms.GRNInfo attribute*), 12, 13

gateInwardNumber (*InvManage.forms.GRNInfo attribute*), 28, 29

gateInwardNumber (*InvManage.models.GoodsReceiptNote attribute*), 64, 65

gateInwardNumber (*InvManage.models.objects.GoodsReceiptNote attribute*), 45

gateInwardNumber (*InvManage.views.GoodsReceiptNote attribute*), 158,

159, 211, 212, 261, 262, 299, 300, 339, 377, 421

gateInwardNumber (*InvManage.views.GoodsReceiptNoteFilter attribute*), 170

gateInwardNumber (*InvManage.views.GRNInfo attribute*), 142, 143, 195, 196, 245, 246, 283, 284, 322, 323, 360, 361, 404, 405

get_columns() (in module *InvManage.scripts.filters*), 75

get_columns() (in module *InvManage.views*), 169, 222, 272, 310, 349, 387, 431

get_consumer() (in module *InvManage.views*), 354

get_consumer() (in module *InvManage.views.consumer_views*), 84

get_grnes() (*InvManage.serializers.GoodsReceiptNoteSerializer method*), 448

get_grnes() (*InvManage.views.GoodsReceiptNoteSerializer method*), 176

get_product_purchase_entries_view() (in module *InvManage.views*), 237

get_product_purchase_entries_view() (in module *InvManage.views.purchase_order_views*), 107

get_vendor() (in module *InvManage.views*), 316

get_vendor() (in module *InvManage.views.vendor_views*), 122

GoodsReceiptNote (class in *InvManage.models*), 64

GoodsReceiptNote (class in *InvManage.models.objects*), 44

GoodsReceiptNote (class in *InvManage.views*), 158, 210, 261, 299, 338, 376, 420

GoodsReceiptNoteFilter (class in *InvManage.filters*), 437

GoodsReceiptNoteFilter (class in *InvManage.views*), 170

GoodsReceiptNoteFilter.Meta (class in *InvManage.filters*), 438

GoodsReceiptNoteFilter.Meta (class in *InvManage.views*), 170

GoodsReceiptNoteSerializer (class in *InvManage.serializers*), 446

GoodsReceiptNoteSerializer (class in *InvManage.views*), 174

GoodsReceiptNoteSerializer.Meta (class in *InvManage.serializers*), 448

GoodsReceiptNoteSerializer.Meta (class in *InvManage.views*), 176

grn (*InvManage.models.GRNEntry attribute*), 61

grn (*InvManage.models.objects.GRNEntry attribute*), 50

grn (*InvManage.models.objects.GRNInvoice attribute*), 51, 52

grn (*InvManage.serializers.GRNEntrySerializer at-*

- tribute), 445
- grn (*InvManage.serializers.GRNEntryWithPORefSerializer* attribute), 446
- grn (*InvManage.serializers.GRNInvoiceSerializer* attribute), 457
- grn (*InvManage.views.GRNEntry* attribute), 154, 207, 258, 295, 335, 373, 417
- grn (*InvManage.views.GRNEntrySerializer* attribute), 171
- grn (*InvManage.views.GRNEntryWithPORefSerializer* attribute), 172
- grn (*InvManage.views.GRNInvoice* attribute), 181, 182
- grn (*InvManage.views.GRNInvoiceSerializer* attribute), 174
- grne (*InvManage.models.GRNEntry* attribute), 60
- grne (*InvManage.models.objects.GRNEntry* attribute), 50
- grne (*InvManage.views.GRNEntry* attribute), 154, 207, 257, 295, 334, 372, 416
- grne_id (*InvManage.forms.forms.GRNEntryForm* attribute), 15
- grne_id (*InvManage.forms.GRNEntryForm* attribute), 29, 30
- grne_id (*InvManage.serializers.GRNEntrySerializer* attribute), 445
- grne_id (*InvManage.serializers.GRNEntryWithPORefSerializer* attribute), 446
- grne_id (*InvManage.views.GRNEntryForm* attribute), 143, 144, 196, 246, 247, 284, 285, 323, 324, 361, 362, 406
- grne_id (*InvManage.views.GRNEntrySerializer* attribute), 171
- grne_id (*InvManage.views.GRNEntryWithPORefSerializer* attribute), 172
- GRNEntry (class in *InvManage.models*), 60
- GRNEntry (class in *InvManage.models.objects*), 49
- GRNEntry (class in *InvManage.views*), 154, 207, 257, 295, 334, 372, 416
- GRNEntryForm (class in *InvManage.forms*), 29
- GRNEntryForm (class in *InvManage.forms.forms*), 15
- GRNEntryForm (class in *InvManage.views*), 143, 196, 246, 284, 323, 361, 405
- GRNEntrySerializer (class in *InvManage.serializers*), 445
- GRNEntrySerializer (class in *InvManage.views*), 170
- GRNEntrySerializer.Meta (class in *InvManage.serializers*), 445
- GRNEntrySerializer.Meta (class in *InvManage.views*), 171
- GRNEntryWithPORefSerializer (class in *InvManage.serializers*), 446
- GRNEntryWithPORefSerializer (class in *InvManage.views*), 171
- GRNEntryWithPORefSerializer.Meta (class in *InvManage.serializers*), 446
- GRNEntryWithPORefSerializer.Meta (class in *InvManage.views*), 172
- grnes (*InvManage.serializers.GoodsReceiptNoteSerializer* attribute), 448
- grnes (*InvManage.views.GoodsReceiptNoteSerializer* attribute), 176
- GRNInfo (class in *InvManage.forms*), 27
- GRNInfo (class in *InvManage.forms.forms*), 11
- GRNInfo (class in *InvManage.views*), 141, 194, 244, 282, 322, 360, 404
- GRNInvoice (class in *InvManage.models.objects*), 51
- GRNInvoice (class in *InvManage.views*), 181
- GRNInvoiceSerializer (class in *InvManage.serializers*), 455
- GRNInvoiceSerializer (class in *InvManage.views*), 172
- GRNInvoiceSerializer.Meta (class in *InvManage.serializers*), 457
- GRNInvoiceSerializer.Meta (class in *InvManage.views*), 174
- grnType (*InvManage.filters.GoodsReceiptNoteFilter* attribute), 438
- grnType (*InvManage.forms.forms.GRNInfo* attribute), 11, 12
- grnType (*InvManage.forms.GRNInfo* attribute), 27, 28
- grnType (*InvManage.models.GoodsReceiptNote* attribute), 64, 65
- grnType (*InvManage.models.objects.GoodsReceiptNote* attribute), 44, 45
- grnType (*InvManage.views.GoodsReceiptNote* attribute), 158, 159, 211, 212, 261, 262, 299, 300, 338, 339, 376, 377, 420, 421
- grnType (*InvManage.views.GoodsReceiptNoteFilter* attribute), 170
- grnType (*InvManage.views.GRNInfo* attribute), 142, 143, 194, 195, 245, 246, 283, 284, 322, 323, 360, 361, 404, 405
- gstn (*InvManage.forms.CompanyForm* attribute), 32, 33
- gstn (*InvManage.forms.ConsumerForm* attribute), 31, 32
- gstn (*InvManage.forms.forms.CompanyForm* attribute), 17, 18
- gstn (*InvManage.forms.forms.ConsumerForm* attribute), 14, 15
- gstn (*InvManage.forms.vendor_forms.VendorForm* attribute), 18, 19
- gstn (*InvManage.forms.VendorForm* attribute), 34
- gstn (*InvManage.models.Company* attribute), 67, 68
- gstn (*InvManage.models.Consumer* attribute), 65, 66
- gstn (*InvManage.models.objects.Company* attribute), 40, 41
- gstn (*InvManage.models.objects.Consumer* attribute),

- 42
 gstin (*InvManage.models.objects.Vendor* attribute), 41, 42
 gstin (*InvManage.models.Vendor* attribute), 66, 67
 gstin (*InvManage.views.Company* attribute), 161, 162, 214, 215, 264, 265, 302, 303, 341, 342, 379, 380, 424
 gstin (*InvManage.views.CompanyForm* attribute), 148, 201, 251, 289, 328, 329, 366, 367, 410, 411
 gstin (*InvManage.views.Consumer* attribute), 159, 160, 212, 213, 262, 263, 300, 301, 340, 378, 422
 gstin (*InvManage.views.ConsumerForm* attribute), 147, 148, 200, 250, 251, 288, 327, 328, 365, 366, 409, 410
 gstin (*InvManage.views.Vendor* attribute), 160, 161, 213, 263, 264, 301, 302, 341, 378, 379, 423
 gstin (*InvManage.views.VendorForm* attribute), 147, 199, 250, 288, 327, 365, 409
- ## H
- height (*InvManage.forms.forms.ProductDetailedInfoForm* attribute), 8
 height (*InvManage.forms.ProductDetailedInfoForm* attribute), 23
 height (*InvManage.models.objects.Product* attribute), 47, 48
 height (*InvManage.models.Product* attribute), 58, 59
 height (*InvManage.views.Product* attribute), 152, 153, 205, 255, 256, 293, 294, 332, 333, 370, 371, 414, 415
 height (*InvManage.views.ProductDetailedInfoForm* attribute), 137, 138, 190, 240, 241, 278, 279, 318, 356, 400
 HistoryFilterState (class in *InvManage.models*), 72
 HistoryFilterState (class in *InvManage.models.misc*), 39
 HistoryFilterState (class in *InvManage.views*), 166, 218, 269, 307, 346, 384, 428, 433
 HistoryFilterStateSerializer (class in *InvManage.serializers*), 460
 HistoryFilterStateSerializer (class in *InvManage.views*), 434
 HistoryFilterStateSerializer.Meta (class in *InvManage.serializers*), 460
 HistoryFilterStateSerializer.Meta (class in *InvManage.views*), 434
 HistoryForm (class in *InvManage.forms*), 33
 HistoryForm (class in *InvManage.forms.forms*), 18
 HistoryForm (class in *InvManage.views*), 149, 201, 252, 289, 329, 367, 411, 434
- ## I
- iban (*InvManage.forms.BankAccountForm* attribute), 36, 37
 iban (*InvManage.forms.vendor_forms.BankAccountForm* attribute), 21, 22
 iban (*InvManage.models.BankAccount* attribute), 74, 75
 iban (*InvManage.models.reuse.BankAccount* attribute), 55
 iban (*InvManage.views.BankAccount* attribute), 168, 169, 221, 271, 272, 309, 310, 349, 387, 431
 iban (*InvManage.views.BankAccountForm* attribute), 150, 151, 203, 204, 254, 291, 292, 331, 369, 413
 identifier (*InvManage.filters.ConsumerFilter* attribute), 437
 identifier (*InvManage.filters.GoodsReceiptNoteFilter* attribute), 438
 identifier (*InvManage.filters.ProductFilter* attribute), 436
 identifier (*InvManage.filters.VendorFilter* attribute), 437
 identifier (*InvManage.forms.ConsumerForm* attribute), 31, 32
 identifier (*InvManage.forms.forms.ConsumerForm* attribute), 14, 15
 identifier (*InvManage.forms.forms.GRNInfo* attribute), 11, 12
 identifier (*InvManage.forms.forms.ProductStatusForm* attribute), 10
 identifier (*InvManage.forms.GRNInfo* attribute), 27, 28
 identifier (*InvManage.forms.ProductStatusForm* attribute), 25, 26
 identifier (*InvManage.forms.vendor_forms.VendorForm* attribute), 18, 19
 identifier (*InvManage.forms.VendorForm* attribute), 34
 identifier (*InvManage.models.Consumer* attribute), 65, 66
 identifier (*InvManage.models.GoodsReceiptNote* attribute), 64, 65
 identifier (*InvManage.models.misc.Object* attribute), 37
 identifier (*InvManage.models.Object* attribute), 70
 identifier (*InvManage.models.objects.Consumer* attribute), 42
 identifier (*InvManage.models.objects.GoodsReceiptNote* attribute), 44, 45
 identifier (*InvManage.models.objects.Product* attribute), 47, 48
 identifier (*InvManage.models.objects.Vendor* attribute), 41, 42
 identifier (*InvManage.models.Product* attribute), 58
 identifier (*InvManage.models.Vendor* attribute), 66, 67
 identifier (*InvManage.views.Consumer* attribute), 159, 160, 212, 213, 262, 263, 300, 301, 340,

- 377, 378, 422
- identifier (*InvManage.views.ConsumerFilter attribute*), 350
- identifier (*InvManage.views.ConsumerForm attribute*), 147, 148, 200, 250, 251, 288, 327, 328, 365, 366, 409, 410
- identifier (*InvManage.views.GoodsReceiptNote attribute*), 158, 159, 211, 212, 261, 262, 299, 300, 338, 339, 376, 377, 420, 421
- identifier (*InvManage.views.GoodsReceiptNoteFilter attribute*), 170
- identifier (*InvManage.views.GRNInfo attribute*), 142, 143, 194, 195, 245, 246, 282, 284, 322, 323, 360, 361, 404, 405
- identifier (*InvManage.views.Object attribute*), 164, 216, 217, 267, 305, 344, 382, 426
- identifier (*InvManage.views.Product attribute*), 151, 152, 204, 205, 255, 292, 293, 332, 333, 370, 371, 414, 415
- identifier (*InvManage.views.ProductFilter attribute*), 133
- identifier (*InvManage.views.ProductStatusForm attribute*), 139, 140, 192, 243, 280, 281, 320, 358, 402
- identifier (*InvManage.views.Vendor attribute*), 160, 161, 213, 263, 264, 301, 302, 341, 378, 379, 423
- identifier (*InvManage.views.VendorFilter attribute*), 311
- identifier (*InvManage.views.VendorForm attribute*), 147, 199, 250, 287, 288, 327, 365, 409
- image (*InvManage.forms.forms.ThumbnailForm attribute*), 9
- image (*InvManage.forms.ThumbnailForm attribute*), 30
- image (*InvManage.models.Company attribute*), 68
- image (*InvManage.models.objects.Company attribute*), 40, 41
- image (*InvManage.models.objects.Product attribute*), 48
- image (*InvManage.models.Product attribute*), 58, 59
- image (*InvManage.views.Company attribute*), 161, 162, 214, 215, 265, 302, 303, 342, 380, 424
- image (*InvManage.views.Product attribute*), 152, 153, 205, 255, 256, 293, 294, 333, 370, 371, 415
- image (*InvManage.views.ThumbnailForm attribute*), 144, 197, 247, 285, 324, 362, 406, 407
- inspectedBy (*InvManage.filters.GoodsReceiptNoteFilter attribute*), 438
- inspectedBy (*InvManage.forms.forms.GRNInfo attribute*), 13
- inspectedBy (*InvManage.forms.GRNInfo attribute*), 29
- inspectedBy (*InvManage.models.GoodsReceiptNote attribute*), 65
- inspectedBy (*InvManage.models.objects.GoodsReceiptNote attribute*), 45
- inspectedBy (*InvManage.views.GoodsReceiptNote attribute*), 159, 212, 262, 300, 339, 377, 421
- inspectedBy (*InvManage.views.GoodsReceiptNoteFilter attribute*), 170
- inspectedBy (*InvManage.views.GRNInfo attribute*), 143, 196, 246, 284, 323, 361, 405
- InvManage
module, 7
- InvManage.filters
module, 436
- InvManage.forms
module, 7
- InvManage.forms.forms
module, 7
- InvManage.forms.vendor_forms
module, 18
- InvManage.models
module, 37
- InvManage.models.misc
module, 37
- InvManage.models.objects
module, 39
- InvManage.models.reuse
module, 52
- InvManage.models.states
module, 56
- InvManage.scripts
module, 75
- InvManage.scripts.filters
module, 75
- InvManage.serializers
module, 439
- InvManage.views
module, 76
- InvManage.views.company_views
module, 76
- InvManage.views.consumer_views
module, 80
- InvManage.views.goods_receipt_note_views
module, 84
- InvManage.views.history_views
module, 92
- InvManage.views.product_views
module, 94
- InvManage.views.purchase_order_views
module, 98
- InvManage.views.sales_order_views
module, 110
- InvManage.views.vendor_views
module, 118

is_complete() (InvManage.models.objects.ProductPurchaseEntry method), 49
 is_complete() (InvManage.models.objects.PurchaseOrder method), 43, 44
 is_complete() (InvManage.models.ProductPurchaseEntry method), 59, 60
 is_complete() (InvManage.models.PurchaseOrder method), 63, 64
 is_complete() (InvManage.views.ProductPurchaseEntry method), 153, 154, 206, 256, 257, 294, 295, 334, 372, 416
 is_complete() (InvManage.views.PurchaseOrder method), 157, 158, 210, 260, 261, 298, 299, 337, 338, 375, 376, 420
 item_type (InvManage.forms.forms.ProductBasicInfoForm attribute), 8
 item_type (InvManage.forms.ProductBasicInfoForm attribute), 23
 item_type (InvManage.models.objects.Product attribute), 47, 48
 item_type (InvManage.models.Product attribute), 57, 58
 item_type (InvManage.views.Product attribute), 151, 152, 204, 205, 254, 255, 292, 293, 332, 333, 369, 371, 414, 415
 item_type (InvManage.views.ProductBasicInfoForm attribute), 137, 189, 190, 240, 278, 317, 355, 399, 400
L
 label (InvManage.models.EventType attribute), 71
 label (InvManage.models.FilterColumn attribute), 69
 label (InvManage.models.misc.EventType attribute), 38
 label (InvManage.models.misc.ObjectModel attribute), 38
 label (InvManage.models.ObjectModel attribute), 71, 72
 label (InvManage.models.states.FilterColumn attribute), 56
 label (InvManage.views.EventType attribute), 165, 218, 268, 306, 345, 383, 427
 label (InvManage.views.FilterColumn attribute), 163, 164, 216, 266, 267, 304, 305, 344, 381, 382, 426
 label (InvManage.views.ObjectModel attribute), 165, 218, 268, 269, 306, 346, 383, 384, 428
 language (InvManage.forms.CommunicationForm attribute), 35
 language (InvManage.forms.vendor_forms.CommunicationForm attribute), 20
 language (InvManage.models.Communication attribute), 73
 language (InvManage.models.reuse.Communication attribute), 53, 54
 language (InvManage.views.Communication attribute), 167, 168, 220, 270, 271, 308, 309, 348, 386, 430
 language (InvManage.views.CommunicationForm attribute), 149, 202, 252, 290, 329, 330, 367, 411, 412
 length (InvManage.forms.forms.ProductDetailedInfoForm attribute), 8
 length (InvManage.forms.ProductDetailedInfoForm attribute), 23
 length (InvManage.models.objects.Product attribute), 47, 48
 length (InvManage.models.Product attribute), 58, 59
 length (InvManage.views.Product attribute), 152, 204, 205, 255, 256, 293, 332, 333, 370, 371, 414, 415
 length (InvManage.views.ProductDetailedInfoForm attribute), 137, 138, 190, 240, 241, 278, 279, 318, 355, 356, 400
 location (InvManage.filters.CompanyFilter attribute), 439
 location (InvManage.filters.ConsumerFilter attribute), 437
 location (InvManage.filters.ProductFilter attribute), 436
 location (InvManage.filters.VendorFilter attribute), 437
 location (InvManage.forms.CompanyForm attribute), 32, 33
 location (InvManage.forms.ConsumerForm attribute), 32
 location (InvManage.forms.forms.CompanyForm attribute), 17, 18
 location (InvManage.forms.forms.ConsumerForm attribute), 14, 15
 location (InvManage.forms.forms.ProductStatusForm attribute), 10
 location (InvManage.forms.ProductStatusForm attribute), 25, 26
 location (InvManage.models.Company attribute), 67, 68
 location (InvManage.models.Consumer attribute), 66
 location (InvManage.models.objects.Company attribute), 40, 41
 location (InvManage.models.objects.Consumer attribute), 42, 43
 location (InvManage.models.objects.Product attribute), 47, 48
 location (InvManage.models.Product attribute), 58
 location (InvManage.views.Company attribute), 161, 162, 214, 215, 264, 265, 302, 303, 342, 380,

424
location (*InvManage.views.CompanyFilter* attribute), 274
location (*InvManage.views.CompanyForm* attribute), 148, 201, 251, 252, 289, 329, 366, 367, 411
location (*InvManage.views.Consumer* attribute), 160, 212, 213, 263, 301, 340, 378, 422
location (*InvManage.views.ConsumerFilter* attribute), 350
location (*InvManage.views.ConsumerForm* attribute), 147, 148, 200, 250, 251, 288, 289, 328, 366, 410
location (*InvManage.views.Product* attribute), 152, 204, 205, 255, 293, 332, 333, 370, 371, 414, 415
location (*InvManage.views.ProductFilter* attribute), 133
location (*InvManage.views.ProductStatusForm* attribute), 140, 192, 243, 280, 281, 320, 358, 402
location (*InvManage.views.VendorFilter* attribute), 311

M

minorder (*InvManage.forms.PurchaseDataForm* attribute), 35, 36
minorder (*InvManage.forms.vendor_forms.PurchaseDataForm* attribute), 20, 21
minorder (*InvManage.models.PurchaseData* attribute), 73, 74
minorder (*InvManage.models.reuse.PurchaseData* attribute), 54
minorder (*InvManage.views.PurchaseData* attribute), 167, 220, 270, 308, 347, 348, 385, 429, 430
minorder (*InvManage.views.PurchaseDataForm* attribute), 149, 150, 202, 203, 253, 290, 291, 330, 368, 412
model (*InvManage.filters.CompanyFilter.Meta* attribute), 438
model (*InvManage.filters.ConsumerFilter.Meta* attribute), 437
model (*InvManage.filters.EventCardFilter.Meta* attribute), 439
model (*InvManage.filters.GoodsReceiptNoteFilter.Meta* attribute), 438
model (*InvManage.filters.ProductFilter.Meta* attribute), 436
model (*InvManage.filters.PurchaseOrderFilter.Meta* attribute), 437
model (*InvManage.filters.SalesOrderFilter.Meta* attribute), 438
model (*InvManage.filters.VendorFilter.Meta* attribute), 436
model (*InvManage.models.misc.Object* attribute), 37
model (*InvManage.models.Object* attribute), 70

model (*InvManage.serializers.CommunicationSerializer.Meta* attribute), 440
model (*InvManage.serializers.CompanySerializer.Meta* attribute), 451
model (*InvManage.serializers.ConsumerSerializer.Meta* attribute), 442
model (*InvManage.serializers.EventTypeSerializer.Meta* attribute), 459
model (*InvManage.serializers.GoodsReceiptNoteSerializer.Meta* attribute), 448
model (*InvManage.serializers.GRNEntrySerializer.Meta* attribute), 445
model (*InvManage.serializers.GRNEntryWithPORefSerializer.Meta* attribute), 446
model (*InvManage.serializers.GRNInvoiceSerializer.Meta* attribute), 457
model (*InvManage.serializers.HistoryFilterStateSerializer.Meta* attribute), 460
model (*InvManage.serializers.ObjectModelSerializer.Meta* attribute), 460
model (*InvManage.serializers.PPEntrySerializer.Meta* attribute), 443
model (*InvManage.serializers.ProductSerializer.Meta* attribute), 441
model (*InvManage.serializers.PSEntrySerializer.Meta* attribute), 449
model (*InvManage.serializers.PurchaseInvoiceSerializer.Meta* attribute), 455
model (*InvManage.serializers.PurchaseOrderSerializer.Meta* attribute), 445
model (*InvManage.serializers.SalesInvoiceSerializer.Meta* attribute), 459
model (*InvManage.serializers.SalesOrderSerializer.Meta* attribute), 451
model (*InvManage.serializers.ShippingAddressSerializer.Meta* attribute), 440
model (*InvManage.serializers.VendorSerializer.Meta* attribute), 441
model (*InvManage.views.CompanyFilter.Meta* attribute), 274
model (*InvManage.views.CompanySerializer.Meta* attribute), 273
model (*InvManage.views.ConsumerFilter.Meta* attribute), 350
model (*InvManage.views.ConsumerSerializer.Meta* attribute), 351
model (*InvManage.views.EventCardFilter.Meta* attribute), 433
model (*InvManage.views.GoodsReceiptNoteFilter.Meta* attribute), 170
model (*InvManage.views.GoodsReceiptNoteSerializer.Meta* attribute), 176
model (*InvManage.views.GRNEntrySerializer.Meta* attribute), 171

- `model` (*InvManage.views.GRNEntryWithPORefSerializer.Meta attribute*), 172
 - `model` (*InvManage.views.GRNInvoiceSerializer.Meta attribute*), 174
 - `model` (*InvManage.views.HistoryFilterStateSerializer.Meta attribute*), 434
 - `model` (*InvManage.views.Object attribute*), 164, 217, 267, 305, 344, 382, 426
 - `model` (*InvManage.views.PPEntrySerializer.Meta attribute*), 177, 224
 - `model` (*InvManage.views.ProductFilter.Meta attribute*), 132
 - `model` (*InvManage.views.ProductSerializer.Meta attribute*), 177, 223, 389
 - `model` (*InvManage.views.PSEntrySerializer.Meta attribute*), 389
 - `model` (*InvManage.views.PurchaseInvoiceSerializer.Meta attribute*), 181, 228
 - `model` (*InvManage.views.PurchaseOrderFilter.Meta attribute*), 223
 - `model` (*InvManage.views.SalesInvoiceSerializer.Meta attribute*), 392
 - `model` (*InvManage.views.SalesOrderFilter.Meta attribute*), 388
 - `model` (*InvManage.views.ShippingAddressSerializer.Meta attribute*), 274
 - `model` (*InvManage.views.VendorFilter.Meta attribute*), 311
 - `model` (*InvManage.views.VendorSerializer.Meta attribute*), 312
 - `MODEL_CHOICES` (*InvManage.filters.EventCardFilter attribute*), 439
 - `MODEL_CHOICES` (*InvManage.views.EventCardFilter attribute*), 433
 - `modelName` (*InvManage.models.misc.ObjectModel attribute*), 38
 - `modelName` (*InvManage.models.ObjectModel attribute*), 71, 72
 - `modelName` (*InvManage.views.ObjectModel attribute*), 165, 218, 268, 269, 306, 346, 384, 428
 - `module`
 - `InvManage`, 7
 - `InvManage.filters`, 436
 - `InvManage.forms`, 7
 - `InvManage.forms.forms`, 7
 - `InvManage.forms.vendor_forms`, 18
 - `InvManage.models`, 37
 - `InvManage.models.misc`, 37
 - `InvManage.models.objects`, 39
 - `InvManage.models.reuse`, 52
 - `InvManage.models.states`, 56
 - `InvManage.scripts`, 75
 - `InvManage.scripts.filters`, 75
 - `InvManage.serializers`, 439
 - `InvManage.views`, 76
 - `InvManage.views.company_views`, 76
 - `InvManage.views.consumer_views`, 80
 - `InvManage.views.goods_receipt_note_views`, 84
 - `InvManage.views.history_views`, 92
 - `InvManage.views.product_views`, 94
 - `InvManage.views.purchase_order_views`, 98
 - `InvManage.views.sales_order_views`, 110
 - `InvManage.views.vendor_views`, 118
- ## N
- `name` (*InvManage.filters.CompanyFilter attribute*), 439
 - `name` (*InvManage.filters.ConsumerFilter attribute*), 437
 - `name` (*InvManage.filters.ProductFilter attribute*), 436
 - `name` (*InvManage.filters.VendorFilter attribute*), 437
 - `name` (*InvManage.forms.BankAccountForm attribute*), 36
 - `name` (*InvManage.forms.CompanyForm attribute*), 32, 33
 - `name` (*InvManage.forms.ConsumerForm attribute*), 31, 32
 - `name` (*InvManage.forms.forms.CompanyForm attribute*), 17, 18
 - `name` (*InvManage.forms.forms.ConsumerForm attribute*), 14, 15
 - `name` (*InvManage.forms.forms.ProductBasicInfoForm attribute*), 8
 - `name` (*InvManage.forms.ProductBasicInfoForm attribute*), 23
 - `name` (*InvManage.forms.ShippingAddressForm attribute*), 33, 34
 - `name` (*InvManage.forms.vendor_forms.BankAccountForm attribute*), 21, 22
 - `name` (*InvManage.forms.vendor_forms.ShippingAddressForm attribute*), 19, 20
 - `name` (*InvManage.forms.vendor_forms.VendorForm attribute*), 18, 19
 - `name` (*InvManage.forms.VendorForm attribute*), 34
 - `name` (*InvManage.models.BankAccount attribute*), 74, 75
 - `name` (*InvManage.models.Company attribute*), 67, 68
 - `name` (*InvManage.models.Consumer attribute*), 65, 66
 - `name` (*InvManage.models.EventType attribute*), 71
 - `name` (*InvManage.models.FilterColumn attribute*), 69
 - `name` (*InvManage.models.FilterState attribute*), 70
 - `name` (*InvManage.models.misc.EventType attribute*), 38
 - `name` (*InvManage.models.misc.Object attribute*), 37
 - `name` (*InvManage.models.misc.ObjectModel attribute*), 38
 - `name` (*InvManage.models.Object attribute*), 70
 - `name` (*InvManage.models.ObjectModel attribute*), 71, 72
 - `name` (*InvManage.models.objects.Company attribute*), 40, 41
 - `name` (*InvManage.models.objects.Consumer attribute*), 42
 - `name` (*InvManage.models.objects.Product attribute*), 47, 48

name (*InvManage.models.objects.Vendor* attribute), 41, 42

name (*InvManage.models.Product* attribute), 57, 58

name (*InvManage.models.reuse.BankAccount* attribute), 55

name (*InvManage.models.reuse.ShippingAddress* attribute), 53

name (*InvManage.models.ShippingAddress* attribute), 72, 73

name (*InvManage.models.states.FilterColumn* attribute), 56

name (*InvManage.models.states.FilterState* attribute), 56

name (*InvManage.models.Vendor* attribute), 66, 67

name (*InvManage.views.BankAccount* attribute), 168, 169, 221, 271, 272, 309, 310, 348, 349, 386, 387, 430, 431

name (*InvManage.views.BankAccountForm* attribute), 150, 151, 203, 253, 254, 291, 292, 330, 331, 368, 369, 412, 413

name (*InvManage.views.Company* attribute), 161, 162, 214, 264, 265, 302, 303, 341, 342, 379, 380, 423, 424

name (*InvManage.views.CompanyFilter* attribute), 274

name (*InvManage.views.CompanyForm* attribute), 148, 200, 201, 251, 289, 328, 329, 366, 367, 410, 411

name (*InvManage.views.Consumer* attribute), 159, 160, 212, 213, 262, 263, 300, 301, 340, 377, 378, 422

name (*InvManage.views.ConsumerFilter* attribute), 350

name (*InvManage.views.ConsumerForm* attribute), 147, 200, 250, 251, 288, 327, 328, 365, 366, 409, 410

name (*InvManage.views.EventType* attribute), 165, 218, 268, 306, 345, 383, 427

name (*InvManage.views.FilterColumn* attribute), 163, 216, 266, 267, 304, 343, 344, 381, 382, 426

name (*InvManage.views.FilterState* attribute), 163, 216, 266, 304, 343, 381, 425

name (*InvManage.views.Object* attribute), 164, 217, 267, 305, 344, 382, 426

name (*InvManage.views.ObjectModel* attribute), 165, 218, 268, 269, 306, 346, 383, 384, 428

name (*InvManage.views.Product* attribute), 151, 152, 204, 205, 254, 255, 292, 293, 331, 333, 369, 370, 413, 415

name (*InvManage.views.ProductBasicInfoForm* attribute), 137, 189, 190, 240, 278, 317, 355, 399, 400

name (*InvManage.views.ProductFilter* attribute), 133

name (*InvManage.views.ShippingAddress* attribute), 166, 219, 269, 270, 307, 346, 347, 384, 385, 428, 429

name (*InvManage.views.ShippingAddressForm* attribute), 145, 146, 198, 199, 249, 286, 287, 326, 327,

364, 408, 409

name (*InvManage.views.Vendor* attribute), 160, 161, 213, 263, 264, 301, 302, 340, 341, 378, 379, 422, 423

name (*InvManage.views.VendorFilter* attribute), 311

name (*InvManage.views.VendorForm* attribute), 146, 147, 199, 250, 287, 288, 327, 365, 409

number (*InvManage.forms.BankAccountForm* attribute), 36, 37

number (*InvManage.forms.vendor_forms.BankAccountForm* attribute), 21, 22

number (*InvManage.models.BankAccount* attribute), 74, 75

number (*InvManage.models.reuse.BankAccount* attribute), 55

number (*InvManage.views.BankAccount* attribute), 168, 169, 221, 271, 272, 309, 310, 349, 386, 387, 431

number (*InvManage.views.BankAccountForm* attribute), 150, 151, 203, 204, 253, 254, 291, 292, 331, 369, 413

O

obj (*InvManage.models.EventCard* attribute), 70, 71

obj (*InvManage.models.misc.EventCard* attribute), 38, 39

obj (*InvManage.views.EventCard* attribute), 164, 165, 217, 267, 268, 305, 306, 344, 345, 382, 383, 427, 432, 433

Object (class in *InvManage.models*), 70

Object (class in *InvManage.models.misc*), 37

Object (class in *InvManage.views*), 164, 216, 267, 305, 344, 382, 426

ObjectModel (class in *InvManage.models*), 71

ObjectModel (class in *InvManage.models.misc*), 38

ObjectModel (class in *InvManage.views*), 165, 218, 268, 306, 345, 383, 428

ObjectModelSerializer (class in *InvManage.serializers*), 460

ObjectModelSerializer.Meta (class in *InvManage.serializers*), 460

objId (*InvManage.models.EventCard* attribute), 70, 71

objId (*InvManage.models.misc.EventCard* attribute), 38, 39

objId (*InvManage.views.EventCard* attribute), 164, 165, 217, 267, 268, 305, 306, 345, 382, 383, 427, 432, 433

objmodel (*InvManage.filters.EventCardFilter* attribute), 439

objmodel (*InvManage.models.EventCard* attribute), 70, 71

objmodel (*InvManage.models.misc.EventCard* attribute), 39

objmodel (*InvManage.views.EventCard* attribute), 164, 165, 217, 267, 268, 305, 306, 345, 383, 427,

433
objmodel (*InvManage.views.EventCardFilter* attribute), 433
objModels (*InvManage.serializers.HistoryFilterStateSerializer* attribute), 460
objModels (*InvManage.views.HistoryFilterStateSerializer* attribute), 434
objname (*InvManage.filters.EventCardFilter* attribute), 439
objname (*InvManage.models.EventCard* attribute), 70, 71
objname (*InvManage.models.misc.EventCard* attribute), 39
objname (*InvManage.views.EventCard* attribute), 164, 165, 217, 267, 268, 305, 306, 345, 382, 383, 427, 432, 433
objname (*InvManage.views.EventCardFilter* attribute), 433
operation (*InvManage.filters.EventCardFilter* attribute), 439
operation (*InvManage.models.EventCard* attribute), 71
operation (*InvManage.models.misc.EventCard* attribute), 39
operation (*InvManage.views.EventCard* attribute), 164, 165, 217, 268, 305, 306, 345, 383, 427, 433
operation (*InvManage.views.EventCardFilter* attribute), 434
OPERATION_CHOICES (*InvManage.filters.EventCardFilter* attribute), 439
OPERATION_CHOICES (*InvManage.views.EventCardFilter* attribute), 433
order (*InvManage.models.objects.ProductPurchaseEntry* attribute), 49
order (*InvManage.models.objects.ProductSalesEntry* attribute), 51
order (*InvManage.models.ProductPurchaseEntry* attribute), 59, 60
order (*InvManage.models.ProductSalesEntry* attribute), 61
order (*InvManage.views.ProductPurchaseEntry* attribute), 153, 154, 206, 256, 257, 294, 295, 333, 334, 371, 372, 416
order (*InvManage.views.ProductSalesEntry* attribute), 155, 208, 258, 296, 335, 336, 373, 417, 418
order_by_field (*InvManage.filters.CompanyFilter* attribute), 439
order_by_field (*InvManage.filters.ConsumerFilter* attribute), 437
order_by_field (*InvManage.filters.ProductFilter* attribute), 436
order_by_field (*InvManage.filters.PurchaseOrderFilter* attribute), 437
order_by_field (*InvManage.filters.SalesOrderFilter* attribute), 438
order_by_field (*InvManage.filters.VendorFilter* attribute), 437
order_by_field (*InvManage.views.CompanyFilter* attribute), 274
order_by_field (*InvManage.views.ConsumerFilter* attribute), 350
order_by_field (*InvManage.views.ProductFilter* attribute), 133
order_by_field (*InvManage.views.PurchaseOrderFilter* attribute), 223
order_by_field (*InvManage.views.SalesOrderFilter* attribute), 388
order_by_field (*InvManage.views.VendorFilter* attribute), 311
ordering (*InvManage.filters.CompanyFilter* attribute), 439
ordering (*InvManage.filters.ConsumerFilter* attribute), 437
ordering (*InvManage.filters.GoodsReceiptNoteFilter* attribute), 438
ordering (*InvManage.filters.ProductFilter* attribute), 436
ordering (*InvManage.filters.PurchaseOrderFilter* attribute), 437
ordering (*InvManage.filters.SalesOrderFilter* attribute), 438
ordering (*InvManage.filters.VendorFilter* attribute), 437
ordering (*InvManage.views.CompanyFilter* attribute), 274
ordering (*InvManage.views.ConsumerFilter* attribute), 351
ordering (*InvManage.views.GoodsReceiptNoteFilter* attribute), 170
ordering (*InvManage.views.ProductFilter* attribute), 133
ordering (*InvManage.views.PurchaseOrderFilter* attribute), 223
ordering (*InvManage.views.SalesOrderFilter* attribute), 388
ordering (*InvManage.views.VendorFilter* attribute), 311
ordertotal (*InvManage.filters.PurchaseOrderFilter* attribute), 437
ordertotal (*InvManage.filters.SalesOrderFilter* attribute), 438
ordertotal (*InvManage.forms.forms.PurchaseOrderBasicInfo* attribute), 11
ordertotal (*InvManage.forms.forms.SalesOrderBasicInfo* attribute), 13, 14
ordertotal (*InvManage.forms.PurchaseOrderBasicInfo* attribute), 27
ordertotal (*InvManage.forms.SalesOrderBasicInfo* attribute), 31

ordertotal (*InvManage.models.objects.PurchaseOrder attribute*), 43, 44

ordertotal (*InvManage.models.objects.SalesOrder attribute*), 46

ordertotal (*InvManage.models.PurchaseOrder attribute*), 63, 64

ordertotal (*InvManage.models.SalesOrder attribute*), 62

ordertotal (*InvManage.views.PurchaseOrder attribute*), 157, 210, 260, 261, 298, 337, 338, 375, 376, 419, 420

ordertotal (*InvManage.views.PurchaseOrderBasicInfo attribute*), 141, 194, 244, 282, 321, 322, 359, 360, 403, 404

ordertotal (*InvManage.views.PurchaseOrderFilter attribute*), 223

ordertotal (*InvManage.views.SalesOrder attribute*), 156, 209, 259, 297, 336, 337, 374, 375, 418, 419

ordertotal (*InvManage.views.SalesOrderBasicInfo attribute*), 145, 198, 248, 286, 325, 326, 363, 364, 407, 408

ordertotal (*InvManage.views.SalesOrderFilter attribute*), 388

owner (*InvManage.filters.CompanyFilter attribute*), 439

owner (*InvManage.forms.CompanyForm attribute*), 32, 33

owner (*InvManage.forms.forms.CompanyForm attribute*), 17, 18

owner (*InvManage.models.Company attribute*), 67, 68

owner (*InvManage.models.objects.Company attribute*), 40, 41

owner (*InvManage.views.Company attribute*), 161, 162, 214, 264, 265, 302, 303, 341, 342, 379, 380, 423, 424

owner (*InvManage.views.CompanyFilter attribute*), 274

owner (*InvManage.views.CompanyForm attribute*), 148, 201, 251, 289, 328, 329, 366, 367, 410, 411

P

paginate() (*in module InvManage.scripts.filters*), 76

paginate() (*in module InvManage.views*), 170, 222, 273, 311, 350, 388, 432

paid (*InvManage.forms.forms.PurchaseOrderBasicInfo attribute*), 10, 11

paid (*InvManage.forms.forms.SalesOrderBasicInfo attribute*), 13, 14

paid (*InvManage.forms.PurchaseOrderBasicInfo attribute*), 26, 27

paid (*InvManage.forms.SalesOrderBasicInfo attribute*), 30, 31

paid (*InvManage.models.objects.PurchaseOrder attribute*), 43, 44

paid (*InvManage.models.objects.SalesOrder attribute*), 46

paid (*InvManage.models.PurchaseOrder attribute*), 63

paid (*InvManage.models.SalesOrder attribute*), 62

paid (*InvManage.views.PurchaseOrder attribute*), 157, 210, 260, 298, 337, 338, 375, 376, 419, 420

paid (*InvManage.views.PurchaseOrderBasicInfo attribute*), 141, 193, 194, 244, 281, 282, 321, 322, 359, 403, 404

paid (*InvManage.views.SalesOrder attribute*), 156, 208, 209, 259, 297, 336, 337, 374, 418, 419

paid (*InvManage.views.SalesOrderBasicInfo attribute*), 145, 197, 198, 248, 285, 286, 325, 326, 363, 407, 408

params (*InvManage.models.HistoryFilterState attribute*), 72

params (*InvManage.models.misc.HistoryFilterState attribute*), 39

params (*InvManage.views.HistoryFilterState attribute*), 166, 218, 269, 307, 346, 384, 428, 433

pending_ppes() (*InvManage.models.objects.PurchaseOrder method*), 43, 44

pending_ppes() (*InvManage.models.PurchaseOrder method*), 63, 64

pending_ppes() (*InvManage.views.PurchaseOrder method*), 157, 158, 210, 260, 261, 298, 299, 338, 375, 376, 420

pending_quantity() (*InvManage.models.objects.ProductPurchaseEntry method*), 49

pending_quantity() (*InvManage.models.objects.ProductPurchaseEntry property*), 49

pending_quantity() (*InvManage.models.ProductPurchaseEntry method*), 59

pending_quantity() (*InvManage.models.ProductPurchaseEntry property*), 60

pending_quantity() (*InvManage.views.ProductPurchaseEntry method*), 153, 206, 256, 294, 334, 372, 416

pending_quantity() (*InvManage.views.ProductPurchaseEntry property*), 154, 206, 257, 295, 334, 372, 416

pendingQty (*InvManage.serializers.PPEntrySerializer attribute*), 443

pendingQty (*InvManage.views.PPEntrySerializer attribute*), 177, 224

phone (*InvManage.filters.CompanyFilter attribute*), 439

phone (*InvManage.filters.ConsumerFilter attribute*), 437

phone (*InvManage.filters.VendorFilter attribute*), 437

phone (*InvManage.forms.CommunicationForm attribute*), 437

- tribute), 35
- phone (*InvManage.forms.CompanyForm* attribute), 32, 33
- phone (*InvManage.forms.ConsumerForm* attribute), 31, 32
- phone (*InvManage.forms.forms.CompanyForm* attribute), 17, 18
- phone (*InvManage.forms.forms.ConsumerForm* attribute), 14, 15
- phone (*InvManage.forms.ShippingAddressForm* attribute), 33, 34
- phone (*InvManage.forms.vendor_forms.CommunicationForm* attribute), 20
- phone (*InvManage.forms.vendor_forms.ShippingAddressForm* attribute), 19, 20
- phone (*InvManage.models.Communication* attribute), 73
- phone (*InvManage.models.Company* attribute), 67, 68
- phone (*InvManage.models.Consumer* attribute), 65, 66
- phone (*InvManage.models.objects.Company* attribute), 40, 41
- phone (*InvManage.models.objects.Consumer* attribute), 42
- phone (*InvManage.models.reuse.Communication* attribute), 54
- phone (*InvManage.models.reuse.ShippingAddress* attribute), 53
- phone (*InvManage.models.ShippingAddress* attribute), 72, 73
- phone (*InvManage.serializers.VendorSerializer.Meta* attribute), 441
- phone (*InvManage.views.Communication* attribute), 167, 168, 220, 270, 271, 308, 309, 348, 386, 430
- phone (*InvManage.views.CommunicationForm* attribute), 149, 202, 252, 290, 329, 330, 367, 411, 412
- phone (*InvManage.views.Company* attribute), 161, 162, 214, 215, 264, 265, 302, 303, 342, 379, 380, 424
- phone (*InvManage.views.CompanyFilter* attribute), 274
- phone (*InvManage.views.CompanyForm* attribute), 148, 201, 251, 289, 328, 329, 366, 367, 410, 411
- phone (*InvManage.views.Consumer* attribute), 159, 160, 212, 213, 262, 263, 300, 301, 340, 378, 422
- phone (*InvManage.views.ConsumerFilter* attribute), 350
- phone (*InvManage.views.ConsumerForm* attribute), 147, 148, 200, 250, 251, 288, 289, 327, 328, 365, 366, 410
- phone (*InvManage.views.ShippingAddress* attribute), 166, 167, 219, 269, 270, 307, 308, 346, 347, 384, 385, 428, 429
- phone (*InvManage.views.ShippingAddressForm* attribute), 146, 198, 199, 249, 286, 287, 326, 327, 364, 408, 409
- phone (*InvManage.views.VendorFilter* attribute), 311
- phone (*InvManage.views.VendorSerializer.Meta* attribute), 312
- po (*InvManage.filters.PurchaseOrderFilter* attribute), 437
- po (*InvManage.forms.forms.PurchaseOrderBasicInfo* attribute), 10, 11
- po (*InvManage.forms.PurchaseOrderBasicInfo* attribute), 26, 27
- po (*InvManage.models.objects.PurchaseInvoice* attribute), 51
- po (*InvManage.models.objects.PurchaseOrder* attribute), 43, 44
- po (*InvManage.models.PurchaseInvoice* attribute), 68
- po (*InvManage.models.PurchaseOrder* attribute), 63
- po (*InvManage.serializers.PurchaseInvoiceSerializer* attribute), 455
- po (*InvManage.views.PurchaseInvoice* attribute), 162, 215, 265, 303, 343, 380, 381, 425
- po (*InvManage.views.PurchaseInvoiceSerializer* attribute), 181, 228
- po (*InvManage.views.PurchaseOrder* attribute), 157, 209, 210, 260, 298, 337, 338, 375, 376, 419, 420
- po (*InvManage.views.PurchaseOrderBasicInfo* attribute), 140, 141, 193, 194, 243, 244, 281, 282, 321, 359, 403, 404
- po (*InvManage.views.PurchaseOrderFilter* attribute), 223
- po_id (*InvManage.serializers.GRNEntryWithPORefSerializer* attribute), 446
- po_id (*InvManage.views.GRNEntryWithPORefSerializer* attribute), 172
- poRef (*InvManage.filters.GoodsReceiptNoteFilter* attribute), 438
- poRef (*InvManage.forms.forms.GRNInfo* attribute), 11, 12
- poRef (*InvManage.forms.GRNInfo* attribute), 27, 28
- poRef (*InvManage.models.GoodsReceiptNote* attribute), 64, 65
- poRef (*InvManage.models.objects.GoodsReceiptNote* attribute), 44, 45
- poRef (*InvManage.views.GoodsReceiptNote* attribute), 158, 159, 211, 212, 261, 262, 299, 300, 338, 339, 376, 377, 420, 421
- poRef (*InvManage.views.GoodsReceiptNoteFilter* attribute), 170
- poRef (*InvManage.views.GRNInfo* attribute), 141, 143, 194, 195, 245, 246, 282, 284, 322, 323, 360, 361, 404, 405
- position (*InvManage.models.FilterColumn* attribute), 69
- position (*InvManage.models.states.FilterColumn* attribute), 56
- position (*InvManage.views.FilterColumn* attribute), 163, 164, 216, 266, 267, 304, 305, 344, 382, 426

post (*InvManage.forms.ShippingAddressForm* attribute), 34
 post (*InvManage.forms.vendor_forms.ShippingAddressForm* attribute), 19, 20
 post (*InvManage.models.reuse.ShippingAddress* attribute), 53
 post (*InvManage.models.ShippingAddress* attribute), 72, 73
 post (*InvManage.views.ShippingAddress* attribute), 166, 167, 219, 269, 270, 307, 308, 347, 385, 429
 post (*InvManage.views.ShippingAddressForm* attribute), 146, 199, 249, 287, 326, 327, 364, 365, 408, 409
 ppe (*InvManage.models.GRNEntry* attribute), 60
 ppe (*InvManage.models.objects.GRNEntry* attribute), 50
 ppe (*InvManage.views.GRNEntry* attribute), 154, 207, 257, 295, 334, 372, 416
 ppe_id (*InvManage.forms.forms.GRNEntryForm* attribute), 15
 ppe_id (*InvManage.forms.forms.ProductPurchaseEntryForm* attribute), 16
 ppe_id (*InvManage.forms.GRNEntryForm* attribute), 29, 30
 ppe_id (*InvManage.forms.ProductPurchaseEntryForm* attribute), 24
 ppe_id (*InvManage.serializers.GRNEntryWithPORefSerializer* attribute), 446
 ppe_id (*InvManage.serializers.PPEntrySerializer* attribute), 443
 ppe_id (*InvManage.views.GRNEntryForm* attribute), 143, 144, 196, 246, 247, 284, 285, 324, 361, 362, 406
 ppe_id (*InvManage.views.GRNEntryWithPORefSerializer* attribute), 172
 ppe_id (*InvManage.views.PPEntrySerializer* attribute), 177, 224
 ppe_id (*InvManage.views.ProductPurchaseEntryForm* attribute), 138, 139, 191, 241, 242, 279, 318, 319, 356, 357, 401
 PPEntrySerializer (class in *InvManage.serializers*), 442
 PPEntrySerializer (class in *InvManage.views*), 176, 223
 PPEntrySerializer.Meta (class in *InvManage.serializers*), 443
 PPEntrySerializer.Meta (class in *InvManage.views*), 177, 224
 ppes (*InvManage.models.GRNEntry* attribute), 61
 ppes (*InvManage.models.objects.GRNEntry* attribute), 50
 ppes (*InvManage.serializers.PurchaseOrderSerializer* attribute), 445
 ppes (*InvManage.views.GRNEntry* attribute), 154, 207, 258, 295, 335, 373, 417
 prefix (*InvManage.forms.BankAccountForm* attribute), 36
 prefix (*InvManage.forms.CommunicationForm* attribute), 35
 prefix (*InvManage.forms.CompanyForm* attribute), 33
 prefix (*InvManage.forms.ConsumerForm* attribute), 32
 prefix (*InvManage.forms.forms.CompanyForm* attribute), 18
 prefix (*InvManage.forms.forms.ConsumerForm* attribute), 14
 prefix (*InvManage.forms.forms.GRNEntryForm* attribute), 15
 prefix (*InvManage.forms.forms.GRNInfo* attribute), 12
 prefix (*InvManage.forms.forms.HistoryForm* attribute), 18
 prefix (*InvManage.forms.forms.ProductBasicInfoForm* attribute), 8
 prefix (*InvManage.forms.forms.ProductDetailedInfoForm* attribute), 8
 prefix (*InvManage.forms.forms.ProductPricingForm* attribute), 9
 prefix (*InvManage.forms.forms.ProductPurchaseEntryForm* attribute), 16
 prefix (*InvManage.forms.forms.ProductSalesEntryForm* attribute), 17
 prefix (*InvManage.forms.forms.ProductStatusForm* attribute), 10
 prefix (*InvManage.forms.forms.ProductStorageInfoForm* attribute), 9
 prefix (*InvManage.forms.forms.PurchaseOrderBasicInfo* attribute), 11
 prefix (*InvManage.forms.forms.SalesOrderBasicInfo* attribute), 13
 prefix (*InvManage.forms.forms.ThumbnailForm* attribute), 9
 prefix (*InvManage.forms.GRNEntryForm* attribute), 29
 prefix (*InvManage.forms.GRNInfo* attribute), 28
 prefix (*InvManage.forms.HistoryForm* attribute), 33
 prefix (*InvManage.forms.ProductBasicInfoForm* attribute), 23
 prefix (*InvManage.forms.ProductDetailedInfoForm* attribute), 23
 prefix (*InvManage.forms.ProductPricingForm* attribute), 24
 prefix (*InvManage.forms.ProductPurchaseEntryForm* attribute), 24
 prefix (*InvManage.forms.ProductSalesEntryForm* attribute), 25
 prefix (*InvManage.forms.ProductStatusForm* attribute), 25
 prefix (*InvManage.forms.ProductStorageInfoForm* attribute), 26
 prefix (*InvManage.forms.PurchaseDataForm* attribute), 35
 prefix (*InvManage.forms.PurchaseOrderBasicInfo* at-

tribute), 27

prefix (InvManage.forms.SalesOrderBasicInfo attribute), 31

prefix (InvManage.forms.ShippingAddressForm attribute), 34

prefix (InvManage.forms.ThumbnailForm attribute), 30

prefix (InvManage.forms.vendor_forms.BankAccountForm attribute), 22

prefix (InvManage.forms.vendor_forms.CommunicationForm attribute), 20

prefix (InvManage.forms.vendor_forms.PurchaseDataForm attribute), 21

prefix (InvManage.forms.vendor_forms.ShippingAddressForm attribute), 19

prefix (InvManage.forms.vendor_forms.VendorForm attribute), 19

prefix (InvManage.forms.VendorForm attribute), 34

prefix (InvManage.views.BankAccountForm attribute), 151, 203, 254, 292, 331, 369, 413

prefix (InvManage.views.CommunicationForm attribute), 149, 202, 252, 290, 329, 367, 412

prefix (InvManage.views.CompanyForm attribute), 148, 201, 251, 289, 329, 366, 411

prefix (InvManage.views.ConsumerForm attribute), 147, 200, 251, 288, 328, 366, 410

prefix (InvManage.views.GRNEntryForm attribute), 144, 196, 247, 285, 324, 362, 406

prefix (InvManage.views.GRNInfo attribute), 142, 195, 246, 283, 323, 361, 405

prefix (InvManage.views.HistoryForm attribute), 149, 201, 252, 290, 329, 367, 411, 434

prefix (InvManage.views.ProductBasicInfoForm attribute), 137, 190, 240, 278, 317, 355, 399

prefix (InvManage.views.ProductDetailedInfoForm attribute), 138, 190, 241, 278, 318, 356, 400

prefix (InvManage.views.ProductPricingForm attribute), 138, 191, 241, 279, 318, 356, 400

prefix (InvManage.views.ProductPurchaseEntryForm attribute), 139, 191, 242, 279, 319, 357, 401

prefix (InvManage.views.ProductSalesEntryForm attribute), 139, 192, 242, 280, 319, 357, 402

prefix (InvManage.views.ProductStatusForm attribute), 140, 192, 243, 281, 320, 358, 402

prefix (InvManage.views.ProductStorageInfoForm attribute), 140, 193, 243, 281, 320, 358, 402

prefix (InvManage.views.PurchaseDataForm attribute), 150, 202, 253, 291, 330, 368, 412

prefix (InvManage.views.PurchaseOrderBasicInfo attribute), 141, 194, 244, 282, 321, 359, 403

prefix (InvManage.views.SalesOrderBasicInfo attribute), 145, 198, 248, 286, 325, 363, 407

prefix (InvManage.views.ShippingAddressForm attribute), 146, 199, 249, 287, 326, 364, 409

prefix (InvManage.views.ThumbnailForm attribute), 144, 197, 247, 285, 324, 362, 407

prefix (InvManage.views.VendorForm attribute), 147, 199, 250, 288, 327, 365, 409

preparedBy (InvManage.filters.GoodsReceiptNoteFilter attribute), 438

preparedBy (InvManage.forms.forms.GRNInfo attribute), 12, 13

preparedBy (InvManage.forms.GRNInfo attribute), 28, 29

preparedBy (InvManage.models.GoodsReceiptNote attribute), 64, 65

preparedBy (InvManage.models.objects.GoodsReceiptNote attribute), 45

preparedBy (InvManage.views.GoodsReceiptNote attribute), 158, 159, 211, 212, 261, 262, 299, 300, 339, 377, 421

preparedBy (InvManage.views.GoodsReceiptNoteFilter attribute), 170

preparedBy (InvManage.views.GRNInfo attribute), 142, 143, 195, 196, 245, 246, 283, 284, 322, 323, 360, 361, 405

price (InvManage.filters.ProductFilter attribute), 436

price (InvManage.forms.forms.ProductPricingForm attribute), 9

price (InvManage.forms.forms.ProductPurchaseEntryForm attribute), 16

price (InvManage.forms.forms.ProductSalesEntryForm attribute), 17

price (InvManage.forms.ProductPricingForm attribute), 24

price (InvManage.forms.ProductPurchaseEntryForm attribute), 24

price (InvManage.forms.ProductSalesEntryForm attribute), 25

price (InvManage.models.objects.Product attribute), 47, 48

price (InvManage.models.objects.ProductPurchaseEntry attribute), 48, 49

price (InvManage.models.objects.ProductSalesEntry attribute), 50, 51

price (InvManage.models.Product attribute), 57, 58

price (InvManage.models.ProductPurchaseEntry attribute), 59, 60

price (InvManage.models.ProductSalesEntry attribute), 61

price (InvManage.views.Product attribute), 151, 152, 204, 205, 254, 255, 292, 293, 332, 333, 370, 371, 414, 415

price (InvManage.views.ProductFilter attribute), 133

price (InvManage.views.ProductPricingForm attribute), 138, 190, 191, 241, 279, 318, 356, 400

price (InvManage.views.ProductPurchaseEntry attribute), 153, 154, 206, 256, 257, 294, 295, 333, 334, 371, 372, 415, 416

price (*InvManage.views.ProductPurchaseEntryForm* attribute), 138, 139, 191, 241, 242, 279, 280, 319, 357, 401

price (*InvManage.views.ProductSalesEntry* attribute), 155, 208, 258, 296, 335, 336, 373, 417, 418

price (*InvManage.views.ProductSalesEntryForm* attribute), 139, 192, 242, 280, 319, 320, 357, 358, 401, 402

print_grn_view() (in module *InvManage.views*), 187

print_grn_view() (in module *InvManage.views.goods_receipt_note_views*), 90

print_purchase_order_view() (in module *InvManage.views*), 233

print_purchase_order_view() (in module *InvManage.views.purchase_order_views*), 103

print_sales_order_view() (in module *InvManage.views*), 397

print_sales_order_view() (in module *InvManage.views.sales_order_views*), 115

prod_id (*InvManage.serializers.ProductSerializer* attribute), 441

prod_id (*InvManage.views.ProductSerializer* attribute), 177, 223, 389

Product (class in *InvManage.models*), 57

Product (class in *InvManage.models.objects*), 47

Product (class in *InvManage.views*), 151, 204, 254, 292, 331, 369, 413

product (*InvManage.forms.forms.GRNEntryForm* attribute), 15

product (*InvManage.forms.forms.ProductPurchaseEntryForm* attribute), 16

product (*InvManage.forms.forms.ProductSalesEntryForm* attribute), 16, 17

product (*InvManage.forms.GRNEntryForm* attribute), 29

product (*InvManage.forms.ProductPurchaseEntryForm* attribute), 24

product (*InvManage.forms.ProductSalesEntryForm* attribute), 25

product (*InvManage.models.GRNEntry* attribute), 60

product (*InvManage.models.objects.GRNEntry* attribute), 49, 50

product (*InvManage.models.objects.ProductPurchaseEntryForm* attribute), 48, 49

product (*InvManage.models.objects.ProductSalesEntryForm* attribute), 50, 51

product (*InvManage.models.ProductPurchaseEntryForm* attribute), 59, 60

product (*InvManage.models.ProductSalesEntryForm* attribute), 61

product (*InvManage.serializers.GRNEntrySerializer* attribute), 445

product (*InvManage.serializers.GRNEntryWithPORefSerializer* attribute), 446

product (*InvManage.serializers.PPEntrySerializer* attribute), 443

product (*InvManage.serializers.PSEntrySerializer* attribute), 449

product (*InvManage.views.GRNEntry* attribute), 154, 207, 257, 295, 334, 335, 372, 373, 416, 417

product (*InvManage.views.GRNEntryForm* attribute), 143, 144, 196, 246, 247, 284, 285, 323, 324, 361, 362, 405, 406

product (*InvManage.views.GRNEntrySerializer* attribute), 171

product (*InvManage.views.GRNEntryWithPORefSerializer* attribute), 172

product (*InvManage.views.PPEntrySerializer* attribute), 177, 224

product (*InvManage.views.ProductPurchaseEntryForm* attribute), 153, 206, 256, 257, 294, 333, 334, 371, 372, 415, 416

product (*InvManage.views.ProductPurchaseEntryForm* attribute), 138, 139, 191, 241, 242, 279, 319, 356, 357, 401

product (*InvManage.views.ProductSalesEntryForm* attribute), 155, 207, 208, 258, 296, 335, 373, 417, 418

product (*InvManage.views.ProductSalesEntryForm* attribute), 139, 192, 242, 280, 319, 320, 357, 401, 402

product (*InvManage.views.PSEntrySerializer* attribute), 389

product_choices (*InvManage.forms.forms.GRNInfoForm* attribute), 12

product_choices (*InvManage.forms.forms.PurchaseOrderBasicInfoForm* attribute), 11

product_choices (*InvManage.forms.forms.SalesOrderBasicInfoForm* attribute), 14

product_choices (*InvManage.forms.GRNInfoForm* attribute), 28

product_choices (*InvManage.forms.PurchaseOrderBasicInfoForm* attribute), 27

product_choices (*InvManage.forms.SalesOrderBasicInfoForm* attribute), 31

product_choices (*InvManage.views.GRNInfoForm* attribute), 143, 195, 246, 283, 323, 361, 405

product_choices (*InvManage.views.PurchaseOrderBasicInfoForm* attribute), 141, 194, 244, 282, 321, 359, 403

product_choices (*InvManage.views.SalesOrderBasicInfoForm* attribute), 145, 198, 248, 286, 325, 363, 407

ProductBasicInfoForm (class in *InvManage.forms*), 22

- ProductBasicInfoForm (class in *InvManage.forms.forms*), 7
- ProductBasicInfoForm (class in *InvManage.views*), 137, 189, 240, 278, 317, 355, 399
- ProductDetailedInfoForm (class in *InvManage.forms*), 23
- ProductDetailedInfoForm (class in *InvManage.forms.forms*), 8
- ProductDetailedInfoForm (class in *InvManage.views*), 137, 190, 240, 278, 318, 355, 400
- ProductFilter (class in *InvManage.filters*), 436
- ProductFilter (class in *InvManage.views*), 132
- ProductFilter.Meta (class in *InvManage.filters*), 436
- ProductFilter.Meta (class in *InvManage.views*), 132
- ProductPricingForm (class in *InvManage.forms*), 24
- ProductPricingForm (class in *InvManage.forms.forms*), 9
- ProductPricingForm (class in *InvManage.views*), 138, 190, 241, 279, 318, 356, 400
- ProductPurchaseEntry (class in *InvManage.models*), 59
- ProductPurchaseEntry (class in *InvManage.models.objects*), 48
- ProductPurchaseEntry (class in *InvManage.views*), 153, 205, 256, 294, 333, 371, 415
- ProductPurchaseEntryForm (class in *InvManage.forms*), 24
- ProductPurchaseEntryForm (class in *InvManage.forms.forms*), 16
- ProductPurchaseEntryForm (class in *InvManage.views*), 138, 191, 241, 279, 318, 356, 401
- ProductSalesEntry (class in *InvManage.models*), 61
- ProductSalesEntry (class in *InvManage.models.objects*), 50
- ProductSalesEntry (class in *InvManage.views*), 155, 207, 258, 296, 335, 373, 417
- ProductSalesEntryForm (class in *InvManage.forms*), 25
- ProductSalesEntryForm (class in *InvManage.forms.forms*), 16
- ProductSalesEntryForm (class in *InvManage.views*), 139, 191, 242, 280, 319, 357, 401
- ProductSerializer (class in *InvManage.serializers*), 441
- ProductSerializer (class in *InvManage.views*), 177, 223, 388
- ProductSerializer.Meta (class in *InvManage.serializers*), 441
- ProductSerializer.Meta (class in *InvManage.views*), 177, 223, 389
- ProductStatusForm (class in *InvManage.forms*), 25
- ProductStatusForm (class in *InvManage.forms.forms*), 9
- ProductStatusForm (class in *InvManage.views*), 139, 192, 242, 280, 320, 358, 402
- ProductStorageInfoForm (class in *InvManage.forms*), 26
- ProductStorageInfoForm (class in *InvManage.forms.forms*), 9
- ProductStorageInfoForm (class in *InvManage.views*), 140, 192, 243, 281, 320, 358, 402
- pse_id (*InvManage.forms.forms.ProductSalesEntryForm* attribute), 16, 17
- pse_id (*InvManage.forms.ProductSalesEntryForm* attribute), 25
- pse_id (*InvManage.serializers.PSEntrySerializer* attribute), 449
- pse_id (*InvManage.views.ProductSalesEntryForm* attribute), 139, 191, 192, 242, 280, 319, 320, 357, 401, 402
- pse_id (*InvManage.views.PSEntrySerializer* attribute), 389
- PSEntrySerializer (class in *InvManage.serializers*), 448
- PSEntrySerializer (class in *InvManage.views*), 389
- PSEntrySerializer.Meta (class in *InvManage.serializers*), 449
- PSEntrySerializer.Meta (class in *InvManage.views*), 389
- pses (*InvManage.serializers.SalesOrderSerializer* attribute), 451
- PurchaseData (class in *InvManage.models*), 73
- PurchaseData (class in *InvManage.models.reuse*), 54
- PurchaseData (class in *InvManage.views*), 167, 220, 270, 308, 347, 385, 429
- purchasedata (*InvManage.models.objects.Vendor* attribute), 41, 42
- purchasedata (*InvManage.models.Vendor* attribute), 67
- purchasedata (*InvManage.views.Vendor* attribute), 161, 213, 214, 264, 302, 341, 379, 423
- PurchaseDataForm (class in *InvManage.forms*), 35
- PurchaseDataForm (class in *InvManage.forms.vendor_forms*), 20
- PurchaseDataForm (class in *InvManage.views*), 149, 202, 252, 290, 330, 368, 412
- PurchaseInvoice (class in *InvManage.models*), 68
- PurchaseInvoice (class in *InvManage.models.objects*), 51
- PurchaseInvoice (class in *InvManage.views*), 162, 215, 265, 303, 342, 380, 424
- PurchaseInvoiceSerializer (class in *InvManage.serializers*), 451
- PurchaseInvoiceSerializer (class in *InvManage.views*), 178, 224
- PurchaseInvoiceSerializer.Meta (class in *InvManage.serializers*), 455

`PurchaseInvoiceSerializer.Meta` (class in `InvManage.views`), 181, 228

`PurchaseOrder` (class in `InvManage.models`), 62

`PurchaseOrder` (class in `InvManage.models.objects`), 43

`PurchaseOrder` (class in `InvManage.views`), 156, 209, 259, 297, 337, 375, 419

`PurchaseOrderBasicInfo` (class in `InvManage.forms`), 26

`PurchaseOrderBasicInfo` (class in `InvManage.forms.forms`), 10

`PurchaseOrderBasicInfo` (class in `InvManage.views`), 140, 193, 243, 281, 320, 358, 403

`PurchaseOrderFilter` (class in `InvManage.filters`), 437

`PurchaseOrderFilter` (class in `InvManage.views`), 223

`PurchaseOrderFilter.Meta` (class in `InvManage.filters`), 437

`PurchaseOrderFilter.Meta` (class in `InvManage.views`), 223

`PurchaseOrderSerializer` (class in `InvManage.serializers`), 443

`PurchaseOrderSerializer.Meta` (class in `InvManage.serializers`), 445

Q

`qlen` (`InvManage.forms.forms.HistoryForm` attribute), 18

`qlen` (`InvManage.forms.HistoryForm` attribute), 33

`qlen` (`InvManage.views.HistoryForm` attribute), 149, 201, 252, 290, 329, 367, 411, 434

`quantity` (`InvManage.filters.ProductFilter` attribute), 436

`quantity` (`InvManage.forms.forms.GRNEntryForm` attribute), 15, 16

`quantity` (`InvManage.forms.forms.ProductPurchaseEntryForm` attribute), 16

`quantity` (`InvManage.forms.forms.ProductSalesEntryForm` attribute), 17

`quantity` (`InvManage.forms.forms.ProductStatusForm` attribute), 9, 10

`quantity` (`InvManage.forms.GRNEntryForm` attribute), 29, 30

`quantity` (`InvManage.forms.ProductPurchaseEntryForm` attribute), 24

`quantity` (`InvManage.forms.ProductSalesEntryForm` attribute), 25

`quantity` (`InvManage.forms.ProductStatusForm` attribute), 25, 26

`quantity` (`InvManage.models.GRNEntry` attribute), 60, 61

`quantity` (`InvManage.models.objects.GRNEntry` attribute), 49, 50

`quantity` (`InvManage.models.objects.Product` attribute), 47, 48

`quantity` (`InvManage.models.objects.ProductPurchaseEntry` attribute), 48, 49

`quantity` (`InvManage.models.objects.ProductSalesEntry` attribute), 50, 51

`quantity` (`InvManage.models.Product` attribute), 57, 58

`quantity` (`InvManage.models.ProductPurchaseEntry` attribute), 59, 60

`quantity` (`InvManage.models.ProductSalesEntry` attribute), 61

`quantity` (`InvManage.views.GRNEntry` attribute), 154, 207, 257, 258, 295, 334, 335, 372, 373, 416, 417

`quantity` (`InvManage.views.GRNEntryForm` attribute), 143, 144, 196, 246, 247, 284, 285, 324, 361, 362, 406

`quantity` (`InvManage.views.Product` attribute), 151, 152, 204, 205, 254, 255, 292, 293, 332, 333, 370, 371, 414, 415

`quantity` (`InvManage.views.ProductFilter` attribute), 133

`quantity` (`InvManage.views.ProductPurchaseEntry` attribute), 153, 206, 256, 257, 294, 333, 334, 371, 372, 415, 416

`quantity` (`InvManage.views.ProductPurchaseEntryForm` attribute), 138, 139, 191, 241, 242, 279, 280, 319, 357, 401

`quantity` (`InvManage.views.ProductSalesEntry` attribute), 155, 208, 258, 296, 335, 373, 417, 418

`quantity` (`InvManage.views.ProductSalesEntryForm` attribute), 139, 192, 242, 280, 319, 320, 357, 401, 402

`quantity` (`InvManage.views.ProductStatusForm` attribute), 139, 140, 192, 242, 243, 280, 281, 320, 358, 402

R

`receivedQty` (`InvManage.forms.forms.GRNEntryForm` attribute), 15, 16

`receivedQty` (`InvManage.forms.GRNEntryForm` attribute), 29, 30

`receivedQty` (`InvManage.models.GRNEntry` attribute), 60, 61

`receivedQty` (`InvManage.models.objects.GRNEntry` attribute), 50

`receivedQty` (`InvManage.models.objects.ProductPurchaseEntry` attribute), 49

`receivedQty` (`InvManage.models.ProductPurchaseEntry` attribute), 59, 60

`receivedQty` (`InvManage.views.GRNEntry` attribute), 154, 155, 207, 257, 258, 295, 296, 335, 372, 373, 417

`receivedQty` (`InvManage.views.GRNEntryForm` attribute), 143, 144, 196, 197, 246, 247, 284,

285, 324, 362, 406
 receivedQty (*InvManage.views.ProductPurchaseEntry* attribute), 153, 154, 206, 256, 257, 294, 295, 334, 371, 372, 416
 refcode (*InvManage.forms.PurchaseDataForm* attribute), 35, 36
 refcode (*InvManage.forms.vendor_forms.PurchaseDataForm* attribute), 21
 refcode (*InvManage.models.PurchaseData* attribute), 74
 refcode (*InvManage.models.reuse.PurchaseData* attribute), 54
 refcode (*InvManage.views.PurchaseData* attribute), 167, 220, 270, 308, 347, 348, 385, 429, 430
 refcode (*InvManage.views.PurchaseDataForm* attribute), 150, 202, 203, 253, 291, 330, 368, 412
 region (*InvManage.forms.BankAccountForm* attribute), 36, 37
 region (*InvManage.forms.vendor_forms.BankAccountForm* attribute), 21, 22
 region (*InvManage.models.BankAccount* attribute), 74, 75
 region (*InvManage.models.reuse.BankAccount* attribute), 55
 region (*InvManage.views.BankAccount* attribute), 168, 169, 221, 271, 272, 309, 310, 348, 349, 386, 387, 430, 431
 region (*InvManage.views.BankAccountForm* attribute), 150, 151, 203, 204, 253, 254, 291, 292, 331, 368, 369, 413
 rejectedQty (*InvManage.forms.forms.GRNEntryForm* attribute), 15, 16
 rejectedQty (*InvManage.forms.GRNEntryForm* attribute), 29, 30
 rejectedQty (*InvManage.models.GRNEntry* attribute), 60, 61
 rejectedQty (*InvManage.models.objects.GRNEntry* attribute), 50
 rejectedQty (*InvManage.models.objects.ProductPurchaseEntry* attribute), 49
 rejectedQty (*InvManage.models.ProductPurchaseEntry* attribute), 59, 60
 rejectedQty (*InvManage.views.GRNEntry* attribute), 154, 155, 207, 257, 258, 295, 296, 335, 373, 417
 rejectedQty (*InvManage.views.GRNEntryForm* attribute), 144, 196, 197, 247, 284, 285, 324, 362, 406
 rejectedQty (*InvManage.views.ProductPurchaseEntry* attribute), 153, 154, 206, 256, 257, 294, 295, 334, 372, 416
 remark (*InvManage.forms.forms.GRNEntryForm* attribute), 15, 16
 remark (*InvManage.forms.GRNEntryForm* attribute), 29, 30
 remark (*InvManage.models.GRNEntry* attribute), 60, 61
 remark (*InvManage.models.objects.GRNEntry* attribute), 50
 remark (*InvManage.views.GRNEntry* attribute), 154, 155, 207, 257, 258, 295, 296, 334, 335, 372, 373, 417
 remark (*InvManage.views.GRNEntryForm* attribute), 143, 144, 196, 197, 246, 247, 284, 285, 324, 362, 406
 route (*InvManage.forms.BankAccountForm* attribute), 36, 37
 route (*InvManage.forms.vendor_forms.BankAccountForm* attribute), 21, 22
 route (*InvManage.models.BankAccount* attribute), 74, 75
 route (*InvManage.models.reuse.BankAccount* attribute), 55
 route (*InvManage.views.BankAccount* attribute), 168, 169, 221, 271, 272, 309, 310, 349, 386, 387, 431
 route (*InvManage.views.BankAccountForm* attribute), 150, 151, 203, 204, 253, 254, 291, 292, 331, 368, 369, 413

S

SalesInvoice (class in *InvManage.models*), 69
 SalesInvoice (class in *InvManage.models.objects*), 52
 SalesInvoice (class in *InvManage.views*), 162, 215, 266, 303, 343, 381, 425
 SalesInvoiceSerializer (class in *InvManage.serializers*), 457
 SalesInvoiceSerializer (class in *InvManage.views*), 390
 SalesInvoiceSerializer.Meta (class in *InvManage.serializers*), 459
 SalesInvoiceSerializer.Meta (class in *InvManage.views*), 392
 SalesOrder (class in *InvManage.models*), 61
 SalesOrder (class in *InvManage.models.objects*), 45
 SalesOrder (class in *InvManage.views*), 155, 208, 258, 296, 336, 373, 418
 SalesOrderBasicInfo (class in *InvManage.forms*), 30
 SalesOrderBasicInfo (class in *InvManage.forms.forms*), 13
 SalesOrderBasicInfo (class in *InvManage.views*), 144, 197, 247, 285, 324, 362, 407
 SalesOrderFilter (class in *InvManage.filters*), 438
 SalesOrderFilter (class in *InvManage.views*), 388
 SalesOrderFilter.Meta (class in *InvManage.filters*), 438

SalesOrderFilter.Meta (class in *InvManage.views*), 388
 SalesOrderSerializer (class in *InvManage.serializers*), 449
 SalesOrderSerializer.Meta (class in *InvManage.serializers*), 450
 ShippingAddress (class in *InvManage.models*), 72
 ShippingAddress (class in *InvManage.models.reuse*), 52
 ShippingAddress (class in *InvManage.views*), 166, 219, 269, 307, 346, 384, 428
 shippingaddress (*InvManage.models.Company* attribute), 68
 shippingaddress (*InvManage.models.objects.Company* attribute), 40, 41
 shippingaddress (*InvManage.models.objects.GRNInvoice* attribute), 51, 52
 shippingaddress (*InvManage.models.objects.PurchaseInvoice* attribute), 51
 shippingaddress (*InvManage.models.objects.SalesInvoice* attribute), 52
 shippingaddress (*InvManage.models.PurchaseInvoice* attribute), 68
 shippingaddress (*InvManage.models.SalesInvoice* attribute), 69
 shippingaddress (*InvManage.serializers.GRNInvoiceSerializer* attribute), 457
 shippingaddress (*InvManage.serializers.PurchaseInvoiceSerializer* attribute), 455
 shippingaddress (*InvManage.serializers.SalesInvoiceSerializer* attribute), 459
 shippingaddress (*InvManage.views.Company* attribute), 162, 214, 215, 265, 303, 342, 380, 424
 shippingaddress (*InvManage.views.GRNInvoice* attribute), 181, 182
 shippingaddress (*InvManage.views.GRNInvoiceSerializer* attribute), 174
 shippingaddress (*InvManage.views.PurchaseInvoice* attribute), 162, 215, 265, 303, 343, 380, 381, 425
 shippingaddress (*InvManage.views.PurchaseInvoiceSerializer* attribute), 181, 228
 shippingaddress (*InvManage.views.SalesInvoice* attribute), 163, 215, 216, 266, 304, 343, 381, 425
 shippingaddress (*InvManage.views.SalesInvoiceSerializer* attribute), 392
 ShippingAddressForm (class in *InvManage.forms*), 33
 ShippingAddressForm (class in *InvManage.forms.vendor_forms*), 19
 ShippingAddressForm (class in *InvManage.views*), 145, 198, 248, 286, 326, 364, 408
 ShippingAddressSerializer (class in *InvManage.serializers*), 440
 ShippingAddressSerializer (class in *InvManage.views*), 273
 ShippingAddressSerializer.Meta (class in *InvManage.serializers*), 440
 ShippingAddressSerializer.Meta (class in *InvManage.views*), 274
 so (*InvManage.filters.SalesOrderFilter* attribute), 438
 so (*InvManage.forms.forms.SalesOrderBasicInfo* attribute), 13, 14
 so (*InvManage.forms.SalesOrderBasicInfo* attribute), 30, 31
 so (*InvManage.models.objects.SalesInvoice* attribute), 52
 so (*InvManage.models.objects.SalesOrder* attribute), 46
 so (*InvManage.models.SalesInvoice* attribute), 69
 so (*InvManage.models.SalesOrder* attribute), 62
 so (*InvManage.serializers.SalesInvoiceSerializer* attribute), 459
 so (*InvManage.views.SalesInvoice* attribute), 163, 215, 216, 266, 304, 343, 381, 425
 so (*InvManage.views.SalesInvoiceSerializer* attribute), 392
 so (*InvManage.views.SalesOrder* attribute), 155, 156, 208, 209, 259, 296, 297, 336, 374, 418, 419
 so (*InvManage.views.SalesOrderBasicInfo* attribute), 144, 145, 197, 198, 247, 248, 285, 286, 325, 363, 407, 408
 so (*InvManage.views.SalesOrderFilter* attribute), 388
 sort_ascending_descending() (in module *InvManage.scripts.filters*), 75
 sort_ascending_descending() (in module *InvManage.views*), 169, 222, 272, 310, 349, 387, 431
 state (*InvManage.forms.ShippingAddressForm* attribute), 33, 34
 state (*InvManage.forms.vendor_forms.ShippingAddressForm* attribute), 19, 20
 state (*InvManage.models.FilterColumn* attribute), 69
 state (*InvManage.models.reuse.ShippingAddress* attribute), 53
 state (*InvManage.models.ShippingAddress* attribute), 72, 73
 state (*InvManage.models.states.FilterColumn* attribute), 56
 state (*InvManage.views.FilterColumn* attribute), 163, 164, 216, 266, 267, 304, 305, 344, 382, 426

- state (*InvManage.views.ShippingAddress* attribute), 166, 167, 219, 269, 270, 307, 308, 347, 384, 385, 429
- state (*InvManage.views.ShippingAddressForm* attribute), 146, 198, 199, 249, 287, 326, 327, 364, 408, 409
- subtotal (*InvManage.forms.forms.PurchaseOrderBasicInfo* attribute), 10, 11
- subtotal (*InvManage.forms.forms.SalesOrderBasicInfo* attribute), 13, 14
- subtotal (*InvManage.forms.PurchaseOrderBasicInfo* attribute), 27
- subtotal (*InvManage.forms.SalesOrderBasicInfo* attribute), 31
- subtotal (*InvManage.models.objects.PurchaseOrder* attribute), 43, 44
- subtotal (*InvManage.models.objects.SalesOrder* attribute), 46
- subtotal (*InvManage.models.PurchaseOrder* attribute), 63, 64
- subtotal (*InvManage.models.SalesOrder* attribute), 62
- subtotal (*InvManage.views.PurchaseOrder* attribute), 157, 210, 260, 261, 298, 337, 338, 375, 376, 419, 420
- subtotal (*InvManage.views.PurchaseOrderBasicInfo* attribute), 141, 193, 194, 244, 282, 321, 322, 359, 403, 404
- subtotal (*InvManage.views.SalesOrder* attribute), 156, 209, 259, 297, 336, 337, 374, 418, 419
- subtotal (*InvManage.views.SalesOrderBasicInfo* attribute), 145, 197, 198, 248, 286, 325, 326, 363, 407, 408
- T**
- tax (*InvManage.forms.forms.PurchaseOrderBasicInfo* attribute), 10, 11
- tax (*InvManage.forms.forms.SalesOrderBasicInfo* attribute), 13, 14
- tax (*InvManage.forms.PurchaseOrderBasicInfo* attribute), 26, 27
- tax (*InvManage.forms.SalesOrderBasicInfo* attribute), 30, 31
- tax (*InvManage.models.objects.PurchaseOrder* attribute), 43, 44
- tax (*InvManage.models.objects.SalesOrder* attribute), 46
- tax (*InvManage.models.PurchaseOrder* attribute), 63
- tax (*InvManage.models.SalesOrder* attribute), 62
- tax (*InvManage.views.PurchaseOrder* attribute), 157, 209, 210, 260, 298, 337, 338, 375, 376, 419, 420
- tax (*InvManage.views.PurchaseOrderBasicInfo* attribute), 140, 141, 193, 194, 244, 281, 282, 321, 322, 359, 403, 404
- tax (*InvManage.views.SalesOrder* attribute), 156, 209, 259, 297, 336, 337, 374, 418, 419
- tax (*InvManage.views.SalesOrderBasicInfo* attribute), 145, 197, 198, 248, 285, 286, 325, 326, 363, 407, 408
- taxtotal (*InvManage.forms.forms.PurchaseOrderBasicInfo* attribute), 11
- taxtotal (*InvManage.forms.forms.SalesOrderBasicInfo* attribute), 13, 14
- taxtotal (*InvManage.forms.PurchaseOrderBasicInfo* attribute), 27
- taxtotal (*InvManage.forms.SalesOrderBasicInfo* attribute), 31
- taxtotal (*InvManage.models.objects.PurchaseOrder* attribute), 43, 44
- taxtotal (*InvManage.models.objects.SalesOrder* attribute), 46
- taxtotal (*InvManage.models.PurchaseOrder* attribute), 63, 64
- taxtotal (*InvManage.models.SalesOrder* attribute), 62
- taxtotal (*InvManage.views.PurchaseOrder* attribute), 157, 210, 260, 261, 298, 337, 338, 375, 376, 419, 420
- taxtotal (*InvManage.views.PurchaseOrderBasicInfo* attribute), 141, 193, 194, 244, 282, 321, 322, 359, 403, 404
- taxtotal (*InvManage.views.SalesOrder* attribute), 156, 209, 259, 297, 336, 337, 374, 375, 418, 419
- taxtotal (*InvManage.views.SalesOrderBasicInfo* attribute), 145, 197, 198, 248, 286, 325, 326, 363, 407, 408
- ThumbnailForm (class in *InvManage.forms*), 30
- ThumbnailForm (class in *InvManage.forms.forms*), 9
- ThumbnailForm (class in *InvManage.views*), 144, 197, 247, 285, 324, 362, 406
- title (*InvManage.forms.ShippingAddressForm* attribute), 33, 34
- title (*InvManage.forms.vendor_forms.ShippingAddressForm* attribute), 19
- title (*InvManage.models.reuse.ShippingAddress* attribute), 52, 53
- title (*InvManage.models.ShippingAddress* attribute), 72, 73
- title (*InvManage.views.ShippingAddress* attribute), 166, 219, 269, 270, 307, 346, 347, 384, 385, 428, 429
- title (*InvManage.views.ShippingAddressForm* attribute), 145, 146, 198, 199, 248, 249, 286, 287, 326, 364, 408, 409
- to_representation() (*InvManage.serializers.ConsumerSerializer* method), 442
- to_representation() (*InvManage.serializers.GoodsReceiptNoteSerializer* method), 442

method), 448

to_representation() (InvManage.serializers.GREntrySerializer method), 445

to_representation() (InvManage.serializers.GREntryWithPORefSerializer method), 446

to_representation() (InvManage.serializers.PPEntrySerializer method), 443

to_representation() (InvManage.serializers.ProductSerializer method), 441

to_representation() (InvManage.serializers.PSEntrySerializer method), 449

to_representation() (InvManage.serializers.PurchaseOrderSerializer method), 445

to_representation() (InvManage.serializers.SalesOrderSerializer method), 451

to_representation() (InvManage.serializers.VendorSerializer method), 442

to_representation() (InvManage.views.ConsumerSerializer method), 351

to_representation() (InvManage.views.GoodsReceiptNoteSerializer method), 176

to_representation() (InvManage.views.GREntrySerializer method), 171

to_representation() (InvManage.views.GREntryWithPORefSerializer method), 172

to_representation() (InvManage.views.PPEntrySerializer method), 177, 224

to_representation() (InvManage.views.ProductSerializer method), 178, 223, 389

to_representation() (InvManage.views.PSEntrySerializer method), 390

to_representation() (InvManage.views.VendorSerializer method), 312

transporter (InvManage.forms.forms.GRNInfo attribute), 12

transporter (InvManage.forms.GRNInfo attribute), 28, 29

transporter (InvManage.models.GoodsReceiptNote attribute), 64, 65

transporter (InvManage...

age.models.objects.GoodsReceiptNote attribute), 44, 45

transporter (InvManage.views.GoodsReceiptNote attribute), 158, 159, 211, 212, 261, 262, 299, 300, 339, 376, 377, 421

transporter (InvManage.views.GRNInfo attribute), 142, 143, 195, 245, 246, 283, 284, 322, 323, 360, 361, 404, 405

transportmode (InvManage.forms.PurchaseDataForm attribute), 35, 36

transportmode (InvManage.forms.vendor_forms.PurchaseDataForm attribute), 21

transportmode (InvManage.models.PurchaseData attribute), 74

transportmode (InvManage.models.reuse.PurchaseData attribute), 54

transportmode (InvManage.views.PurchaseData attribute), 167, 220, 270, 308, 348, 385, 386, 430

transportmode (InvManage.views.PurchaseDataForm attribute), 150, 202, 203, 253, 291, 330, 368, 412

TYPE_CHOICES (InvManage.forms.forms.GRNInfo attribute), 12

TYPE_CHOICES (InvManage.forms.GRNInfo attribute), 28

TYPE_CHOICES (InvManage.models.GoodsReceiptNote attribute), 65

TYPE_CHOICES (InvManage.models.objects.GoodsReceiptNote attribute), 45

TYPE_CHOICES (InvManage.views.GoodsReceiptNote attribute), 159, 211, 262, 300, 339, 377, 421

TYPE_CHOICES (InvManage.views.GRNInfo attribute), 143, 195, 246, 284, 323, 361, 405

U

update() (InvManage.serializers.GREntrySerializer method), 446

update() (InvManage.serializers.GREntryWithPORefSerializer method), 446

update() (InvManage.serializers.HistoryFilterStateSerializer method), 461

update() (InvManage.serializers.PPEntrySerializer method), 443

update() (InvManage.serializers.PSEntrySerializer method), 449

update() (InvManage.views.GREntrySerializer method), 171

update() (InvManage.views.GREntryWithPORefSerializer method), 172

[update\(\)](#) (*InvManage.views.HistoryFilterStateSerializer method*), 434
[update\(\)](#) (*InvManage.views.PPEntrySerializer method*), 177, 224
[update\(\)](#) (*InvManage.views.PSEntrySerializer method*), 390
[update_company_view\(\)](#) (*in module InvManage.views*), 275
[update_company_view\(\)](#) (*in module InvManage.views.company_views*), 78
[update_consumer_view\(\)](#) (*in module InvManage.views*), 352
[update_consumer_view\(\)](#) (*in module InvManage.views.consumer_views*), 81
[update_grn_view\(\)](#) (*in module InvManage.views*), 184
[update_grn_view\(\)](#) (*in module InvManage.views.goods_receipt_note_views*), 87
[update_product_view\(\)](#) (*in module InvManage.views*), 134
[update_product_view\(\)](#) (*in module InvManage.views.product_views*), 95
[update_purchase_order_view\(\)](#) (*in module InvManage.views*), 230
[update_purchase_order_view\(\)](#) (*in module InvManage.views.purchase_order_views*), 100
[update_sales_order_view\(\)](#) (*in module InvManage.views*), 394
[update_sales_order_view\(\)](#) (*in module InvManage.views.sales_order_views*), 112
[update_vendor_view\(\)](#) (*in module InvManage.views*), 313
[update_vendor_view\(\)](#) (*in module InvManage.views.vendor_views*), 120
[uploadCSV\(\)](#) (*in module InvManage.views*), 136
[uploadCSV\(\)](#) (*in module InvManage.views.product_views*), 97

V

[vehicleNumber](#) (*InvManage.filters.GoodsReceiptNoteFilter attribute*), 438
[vehicleNumber](#) (*InvManage.forms.forms.GRNInfo attribute*), 12, 13
[vehicleNumber](#) (*InvManage.forms.GRNInfo attribute*), 28, 29
[vehicleNumber](#) (*InvManage.models.GoodsReceiptNote attribute*), 64, 65
[vehicleNumber](#) (*InvManage.models.objects.GoodsReceiptNote attribute*), 45
[vehicleNumber](#) (*InvManage.views.GoodsReceiptNote attribute*), 158, 159, 211, 212, 261, 262, 299, 300, 339, 376, 377, 421
[vehicleNumber](#) (*InvManage.views.GoodsReceiptNoteFilter attribute*), 170
[vehicleNumber](#) (*InvManage.views.GRNInfo attribute*), 142, 143, 195, 196, 245, 246, 283, 284, 322, 323, 360, 361, 404, 405
[Vendor](#) (*class in InvManage.models*), 66
[Vendor](#) (*class in InvManage.models.objects*), 41
[Vendor](#) (*class in InvManage.views*), 160, 213, 263, 301, 340, 378, 422
[vendor](#) (*InvManage.filters.GoodsReceiptNoteFilter attribute*), 438
[vendor](#) (*InvManage.filters.PurchaseOrderFilter attribute*), 437
[vendor](#) (*InvManage.forms.forms.GRNInfo attribute*), 11, 12
[vendor](#) (*InvManage.forms.forms.PurchaseOrderBasicInfo attribute*), 10, 11
[vendor](#) (*InvManage.forms.GRNInfo attribute*), 27, 28
[vendor](#) (*InvManage.forms.PurchaseOrderBasicInfo attribute*), 26, 27
[vendor](#) (*InvManage.models.GoodsReceiptNote attribute*), 64, 65
[vendor](#) (*InvManage.models.objects.GoodsReceiptNote attribute*), 44, 45
[vendor](#) (*InvManage.models.objects.PurchaseOrder attribute*), 43, 44
[vendor](#) (*InvManage.models.PurchaseOrder attribute*), 62, 63
[vendor](#) (*InvManage.serializers.GoodsReceiptNoteSerializer attribute*), 448
[vendor](#) (*InvManage.serializers.PurchaseOrderSerializer attribute*), 445
[vendor](#) (*InvManage.views.GoodsReceiptNote attribute*), 158, 159, 210, 212, 261, 262, 299, 300, 338, 339, 376, 377, 420, 421
[vendor](#) (*InvManage.views.GoodsReceiptNoteFilter attribute*), 170
[vendor](#) (*InvManage.views.GoodsReceiptNoteSerializer attribute*), 176
[vendor](#) (*InvManage.views.GRNInfo attribute*), 141, 143, 194, 195, 244, 246, 282, 283, 322, 323, 360, 361, 404, 405
[vendor](#) (*InvManage.views.PurchaseOrder attribute*), 156, 157, 209, 210, 259, 260, 297, 298, 337, 338, 375, 419, 420
[vendor](#) (*InvManage.views.PurchaseOrderBasicInfo attribute*), 140, 141, 193, 194, 243, 244, 281, 282, 321, 358, 359, 403
[vendor](#) (*InvManage.views.PurchaseOrderFilter attribute*), 223
[vendor_choices](#) (*InvManage.forms.forms.GRNInfo attribute*), 12
[vendor_choices](#) (*InvManage*

age.forms.forms.PurchaseOrderBasicInfo attribute), 11

vendor_choices (*InvManage.forms.GRNInfo* attribute), 28

vendor_choices (*InvManage.forms.PurchaseOrderBasicInfo* attribute), 27

vendor_choices (*InvManage.views.GRNInfo* attribute), 143, 195, 246, 283, 323, 361, 405

vendor_choices (*InvManage.views.PurchaseOrderBasicInfo* attribute), 141, 194, 244, 282, 321, 359, 403

VendorFilter (class in *InvManage.filters*), 436

VendorFilter (class in *InvManage.views*), 311

VendorFilter.Meta (class in *InvManage.filters*), 436

VendorFilter.Meta (class in *InvManage.views*), 311

VendorForm (class in *InvManage.forms*), 34

VendorForm (class in *InvManage.forms.vendor_forms*), 18

VendorForm (class in *InvManage.views*), 146, 199, 249, 287, 327, 365, 409

VendorSerializer (class in *InvManage.serializers*), 441

VendorSerializer (class in *InvManage.views*), 311

VendorSerializer.Meta (class in *InvManage.serializers*), 441

VendorSerializer.Meta (class in *InvManage.views*), 311

visible (*InvManage.models.FilterColumn* attribute), 69

visible (*InvManage.models.states.FilterColumn* attribute), 56

visible (*InvManage.views.FilterColumn* attribute), 163, 164, 216, 266, 267, 304, 305, 344, 381, 382, 426

weight (*InvManage.models.objects.Product* attribute), 47, 48

weight (*InvManage.models.Product* attribute), 58, 59

weight (*InvManage.views.Product* attribute), 152, 153, 205, 255, 256, 293, 294, 332, 333, 370, 371, 414, 415

weight (*InvManage.views.ProductDetailedInfoForm* attribute), 137, 138, 190, 241, 278, 279, 318, 356, 400

width (*InvManage.forms.forms.ProductDetailedInfoForm* attribute), 8

width (*InvManage.forms.ProductDetailedInfoForm* attribute), 23

width (*InvManage.models.objects.Product* attribute), 47, 48

width (*InvManage.models.Product* attribute), 58, 59

width (*InvManage.views.Product* attribute), 152, 204, 205, 255, 256, 293, 332, 333, 370, 371, 414, 415

width (*InvManage.views.ProductDetailedInfoForm* attribute), 137, 138, 190, 240, 241, 278, 279, 318, 356, 400

W

website (*InvManage.forms.ShippingAddressForm* attribute), 34

website (*InvManage.forms.vendor_forms.ShippingAddressForm* attribute), 19, 20

website (*InvManage.models.reuse.ShippingAddress* attribute), 53

website (*InvManage.models.ShippingAddress* attribute), 72, 73

website (*InvManage.views.ShippingAddress* attribute), 166, 167, 219, 269, 270, 307, 308, 347, 385, 429

website (*InvManage.views.ShippingAddressForm* attribute), 146, 199, 249, 287, 326, 327, 364, 365, 408, 409

weight (*InvManage.forms.forms.ProductDetailedInfoForm* attribute), 8

weight (*InvManage.forms.ProductDetailedInfoForm* attribute), 23, 24