

My Project

Generated by Doxygen 1.8.13

Contents

1	Namespace Index	1
1.1	Namespace List	1
2	Hierarchical Index	3
2.1	Class Hierarchy	3
3	Class Index	5
3.1	Class List	5
4	Namespace Documentation	7
4.1	manage Namespace Reference	7
4.1.1	Detailed Description	7
4.2	price.forms Namespace Reference	7
4.2.1	Detailed Description	7
4.3	price.urls Namespace Reference	7
4.3.1	Detailed Description	7
4.3.2	Variable Documentation	8
4.3.2.1	urlpatterns	8
4.4	price.views Namespace Reference	8
4.4.1	Detailed Description	8
4.5	price.webscraping.amazon Namespace Reference	8
4.5.1	Detailed Description	8
4.6	price.webscraping.flipkart Namespace Reference	8
4.6.1	Detailed Description	9
4.6.2	Function Documentation	9

4.6.2.1	flipkart_price()	9
4.7	price.webscraping.paytm Namespace Reference	9
4.7.1	Detailed Description	9
4.7.2	Function Documentation	9
4.7.2.1	paytm_price()	9
4.8	price.webscraping.price_comparator Namespace Reference	10
4.8.1	Detailed Description	10
4.9	price.webscraping.snapdeal Namespace Reference	10
4.9.1	Detailed Description	10
4.9.2	Function Documentation	10
4.9.2.1	snapdeal_price()	10
4.10	priceCom.settings Namespace Reference	11
4.10.1	Detailed Description	11
4.10.2	Variable Documentation	11
4.10.2.1	AUTH_PASSWORD_VALIDATORS	11
4.10.2.2	DATABASES	12
4.10.2.3	INSTALLED_APPS	12
4.10.2.4	MIDDLEWARE	12
4.10.2.5	TEMPLATES	13
4.11	priceCom.urls Namespace Reference	13
4.11.1	Detailed Description	13
4.11.2	Variable Documentation	13
4.11.2.1	urlpatterns	13
4.12	priceCom.wsgi Namespace Reference	14
4.12.1	Detailed Description	14
5	Class Documentation	15
5.1	price.webscraping.price_comparator.price_comp Class Reference	15
5.1.1	Member Function Documentation	15
5.1.1.1	comp()	15
5.1.2	Member Data Documentation	15
5.1.2.1	result	16
5.2	price.apps.PriceConfig Class Reference	16
5.3	price.forms.searching Class Reference	16
5.3.1	Member Data Documentation	16
5.3.1.1	search	16
Index		17

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

manage	7
price.forms	7
price.urls	7
price.views	8
price.webscraping.amazon	8
price.webscraping.flipkart	8
price.webscraping.paytm	9
price.webscraping.price_comparator	10
price.webscraping.snapdeal	10
priceCom.settings	11
priceCom.urls	13
priceCom.wsgi	14

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Form	
price.forms.searching	16
price.webscraping.price_comparator.price_comp	15
AppConfig	
price.apps.PriceConfig	16

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

price.webscraping.price_comparator.price_comp	15
price.apps.PriceConfig	16
price.forms.searching	16

Chapter 4

Namespace Documentation

4.1 manage Namespace Reference

Functions

- def **main** ()

4.1.1 Detailed Description

Django's command-line utility for administrative tasks.

4.2 price.forms Namespace Reference

Classes

- class [searching](#)

4.2.1 Detailed Description

it is search bar for our website created using django

4.3 price.urls Namespace Reference

Variables

- list **urlpatterns**

4.3.1 Detailed Description

If the url provided is empty then `view.index` is accessed otherwise if the url is `search/` then view of `mainSearch`

4.3.2 Variable Documentation

4.3.2.1 urlpatterns

```
list price.urls.urlpatterns
```

Initial value:

```
1 = [
2     path('', views.index, name='index'),
3     path('search/', views.mainSearch, name='mainSearch')
4 ]
```

4.4 price.views Namespace Reference

Functions

- def **index** (request)
- def **mainSearch** (request)

Variables

- **obj** = price_comparator.price_comp()

4.4.1 Detailed Description

If the request method is POST then it will check whether the form is valid . If the form is valid it render pr

4.5 price.webscraping.amazon Namespace Reference

4.5.1 Detailed Description

```
amazon.py
```

```
=====
```

This module is used to do webscraping on amazon website. It searches the product on the amazon website entered

4.6 price.webscraping.flipkart Namespace Reference

Functions

- def **flipkart_price** (item_name)

4.6.1 Detailed Description

flipkart.py

=====

This module is used to do webscraping on flikart website. It searches the product on the flipkart website ente

4.6.2 Function Documentation

4.6.2.1 flipkart_price()

```
def price.webscraping.flipkart.flipkart_price (
    item_name )
```

This function store the important information related to the searched product
Parameters

item_name is the name of the product entered by the user to be searched

4.7 price.webscraping.paytm Namespace Reference

Functions

- def [paytm_price](#) (item_name)

4.7.1 Detailed Description

paytm.py

=====

This module is used to do webscraping on paytm mall website. It searches the product on the paytm mall website

4.7.2 Function Documentation

4.7.2.1 paytm_price()

```
def price.webscraping.paytm.paytm_price (
    item_name )
```

This function store the important information related to the searched product
Parameters

item_name is the name of the product entered by the user to be searched

4.8 price.webscraping.price_comparator Namespace Reference

Classes

- class [price_comp](#)

4.8.1 Detailed Description

price_comparator.py

=====

This is the main program which call the function from amazon.py, flipkart.py, paytm.py and snapdeal.py to extr

4.9 price.webscraping.snapdeal Namespace Reference

Functions

- def [snapdeal_price](#) (item_name)

4.9.1 Detailed Description

snapdeal.py

=====

This module is used to do webscraping on snapdeal website. It searches the product on the sandeal website ente

4.9.2 Function Documentation

4.9.2.1 snapdeal_price()

```
def price.webscraping.snapdeal.snapdeal_price (
    item_name )
```

This function store the important information related to the searched product
Parameters

item_name is the name of the product entered by the user to be searched

4.10 priceCom.settings Namespace Reference

Variables

- **BASE_DIR** = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
- string **SECRET_KEY** = 'j_3ro*cdei6_f%b-cl^#8j_ket4iiigtwzuel+s7*o4sx\$w+=@'
- bool **DEBUG** = True
- list **ALLOWED_HOSTS** = []
- list **INSTALLED_APPS**
- list **MIDDLEWARE**
- string **ROOT_URLCONF** = 'priceCom.urls'
- list **TEMPLATES**
- string **WSGI_APPLICATION** = 'priceCom.wsgi.application'
- dictionary **DATABASES**
- list **AUTH_PASSWORD_VALIDATORS**
- string **LANGUAGE_CODE** = 'en-us'
- string **TIME_ZONE** = 'UTC'
- bool **USE_I18N** = True
- bool **USE_L10N** = True
- bool **USE_TZ** = True
- string **STATIC_URL** = '/static/'

4.10.1 Detailed Description

Django settings for priceCom project.

Generated by 'django-admin startproject' using Django 2.2.6.

For more information on this file, see
<https://docs.djangoproject.com/en/2.2/topics/settings/>

For the full list of settings and their values, see
<https://docs.djangoproject.com/en/2.2/ref/settings/>

4.10.2 Variable Documentation

4.10.2.1 AUTH_PASSWORD_VALIDATORS

list priceCom.settings.AUTH_PASSWORD_VALIDATORS

Initial value:

```

1 = [
2     {
3         'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
4     },
5     {
6         'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
7     },
8     {
9         'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
10    },
11    {
12        'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
13    },
14 ]

```

4.10.2.2 DATABASES

dictionary priceCom.settings.DATABASES

Initial value:

```
1 = {  
2     'default': {  
3         'ENGINE': 'django.db.backends.sqlite3',  
4         'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),  
5     }  
6 }
```

4.10.2.3 INSTALLED_APPS

list priceCom.settings.INSTALLED_APPS

Initial value:

```
1 = [  
2     'price',  
3     'django.contrib.admin',  
4     'django.contrib.auth',  
5     'django.contrib.contenttypes',  
6     'django.contrib.sessions',  
7     'django.contrib.messages',  
8     'django.contrib.staticfiles',  
9 ]
```

4.10.2.4 MIDDLEWARE

list priceCom.settings.MIDDLEWARE

Initial value:

```
1 = [  
2     'django.middleware.security.SecurityMiddleware',  
3     'django.contrib.sessions.middleware.SessionMiddleware',  
4     'django.middleware.common.CommonMiddleware',  
5     'django.middleware.csrf.CsrfViewMiddleware',  
6     'django.contrib.auth.middleware.AuthenticationMiddleware',  
7     'django.contrib.messages.middleware.MessageMiddleware',  
8     'django.middleware.clickjacking.XFrameOptionsMiddleware',  
9 ]
```


4.10.2.5 TEMPLATES

```
list priceCom.settings.TEMPLATES
```

Initial value:

```
1 = [
2     {
3         'BACKEND': 'django.template.backends.django.DjangoTemplates',
4         'DIRS': [],
5         'APP_DIRS': True,
6         'OPTIONS': {
7             'context_processors': [
8                 'django.template.context_processors.debug',
9                 'django.template.context_processors.request',
10                'django.contrib.auth.context_processors.auth',
11                'django.contrib.messages.context_processors.messages',
12            ],
13        },
14    },
15 ]
```

4.11 priceCom.urls Namespace Reference

Variables

- list `urlpatterns`

4.11.1 Detailed Description

mysite URL Configuration

Function views

1. Add an import: `from my_app import views`
2. Add a URL to urlpatterns: `path('', views.home, name='home')`

Class-based views

1. Add an import: `from other_app.views import Home`
2. Add a URL to urlpatterns: `path('', Home.as_view(), name='home')`

Including another URLconf

1. Import the `include()` function: `from django.urls import include, path`
2. Add a URL to urlpatterns: `path('blog/', include('blog.urls'))`

If the user does not provide a url then the user redirected to home page.If admin/ is provided with some url i

4.11.2 Variable Documentation

4.11.2.1 urlpatterns

```
list priceCom.urls.urlpatterns
```

Initial value:

```
1 = [
2     path('admin/', admin.site.urls),
3     path('', include('price.urls')),
4
5 ]
```

4.12 priceCom.wsgi Namespace Reference

Variables

- **application** = get_wsgi_application()

4.12.1 Detailed Description

WSGI config for priceCom project.

It exposes the WSGI callable as a module-level variable named `application`.

For more information on this file, see
<https://docs.djangoproject.com/en/2.2/howto/deployment/wsgi/>

Chapter 5

Class Documentation

5.1 price.webscraping.price_comparator.price_comp Class Reference

Public Member Functions

- `def __init__(self)`
- `def comp(self, item_name)`

Static Public Attributes

- `result`
- `price_sort = sorted(final_dict.items(), key=lambda x: x[1]['price'])`

5.1.1 Member Function Documentation

5.1.1.1 comp()

```
def price.webscraping.price_comparator.price_comp.comp (
    self,
    item_name )
```

this

parameter

it take item_name as input parameter and passes this item to all the functions (amazon_price, flipkart_price, pa and sort them item according to the lowest price.

5.1.2 Member Data Documentation

5.1.2.1 result

```
price.webscraping.price_comparator.price_comp.result [static]
```

it initialise the result list with null.

The documentation for this class was generated from the following file:

- price/webscraping/price_comparator.py

5.2 price.apps.PriceConfig Class Reference

Inheritance diagram for price.apps.PriceConfig:

5.3 price.forms.searching Class Reference

Inheritance diagram for price.forms.searching:

Collaboration diagram for price.forms.searching:

Static Public Attributes

- **search**

5.3.1 Member Data Documentation

5.3.1.1 search

```
price.forms.searching.search [static]
```

Initial value:

```
= forms.CharField(max_length=200,  
    widget=forms.TextInput(attrs={'id': 'mySearch', 'placeholder': 'Search your product...'}))
```

The documentation for this class was generated from the following file:

- price/forms.py

Index

AUTH_PASSWORD_VALIDATORS

priceCom::settings, [11](#)

comp

price::webscraping::price_comparator::price_↔
comp, [15](#)

DATABASES

priceCom::settings, [11](#)

flipkart_price

price::webscraping::flipkart, [9](#)

INSTALLED_APPS

priceCom::settings, [12](#)

MIDDLEWARE

priceCom::settings, [12](#)

manage, [7](#)

paytm_price

price::webscraping::paytm, [9](#)

price.apps.PriceConfig, [16](#)

price.forms, [7](#)

price.forms.searching, [16](#)

price.urls, [7](#)

price.views, [8](#)

price.webscraping.amazon, [8](#)

price.webscraping.flipkart, [8](#)

price.webscraping.paytm, [9](#)

price.webscraping.price_comparator, [10](#)

price.webscraping.price_comparator.price_comp, [15](#)

price.webscraping.snapdeal, [10](#)

price::forms::searching

search, [16](#)

price::urls

urlpatterns, [8](#)

price::webscraping::flipkart

flipkart_price, [9](#)

price::webscraping::paytm

paytm_price, [9](#)

price::webscraping::price_comparator::price_comp

comp, [15](#)

result, [15](#)

price::webscraping::snapdeal

snapdeal_price, [10](#)

priceCom.settings, [11](#)

priceCom.urls, [13](#)

priceCom.wsgi, [14](#)

priceCom::settings

AUTH_PASSWORD_VALIDATORS, [11](#)

DATABASES, [11](#)

INSTALLED_APPS, [12](#)

MIDDLEWARE, [12](#)

TEMPLATES, [12](#)

priceCom::urls

urlpatterns, [13](#)

result

price::webscraping::price_comparator::price_↔
comp, [15](#)

search

price::forms::searching, [16](#)

snapdeal_price

price::webscraping::snapdeal, [10](#)

TEMPLATES

priceCom::settings, [12](#)

urlpatterns

price::urls, [8](#)

priceCom::urls, [13](#)