# VIEWS

**FILE:** views.py   
**Location:** (~/web\_interface/)  
  
Optionally consider logging the software events by including the following:

import logging

log = logging.getLogger(\_\_name\_\_)

|  |  |
| --- | --- |
| **Class:** | DefaultView(TemplateView) |
| **Import requirements:** | from **django.views.generic** import **TemplateView** |
| **Used In:** | HomeView class  RestrictedView class |
| **Purpose:** | *To create a default view of all web pages.* |
| **Definitions/nested classes** | *get() 🡪 short description to be added* |
| *dispatch() 🡪 short description to be added* |

|  |  |
| --- | --- |
| **Class:** | RestrictedView(DefaultView) |
| **Import requirements:** | from **django.utils.decorators** import **method\_decorator**  from **django.contrib.auth.decorators** import **login\_required** |
| **Used In:** | RestrictedFormView class  MyAccountView class located in account\_viewsN.py file |
| **Purpose:** | *Displays a view after registering the user or after login.* |
| **Definitions/nested classes** |  |

|  |  |
| --- | --- |
| **Class:** | RestrictedFormView(RestrictedView, FormMixin, ProcessFormView) |
| **Import requirements:** | from **django.utils.decorators** import **method\_decorator**  from **django.contrib.auth.decorators** import **login\_required**  from **django.views.generic.edit** import **FormMixin, ProcessFormView** |
| **Used In:** | RestrictedFormView class  MyAccountView class located in account\_viewsN.py file  AccountProfileView class located in account\_viewsN.py file |
| **Purpose:** | *Displays the resources used by the user.* |
| **Definitions/nested classes** | *form\_valid()*  *form\_invalid()*  *get() post()* |

|  |  |
| --- | --- |
| **Class:** | HomeView(DefaultView) |
| **Import requirements:** | - |
| **Used In:** | - |
| **Purpose:** | *Display the home page homeN.html and set the page title to Home.* |
| **Definitions/nested classes** |  |

|  |  |
| --- | --- |
| **Class:** | LandingView(RedirectView) |
| **Import requirements:** | from **django.views.generic** import **RedirectView** |
| **Used In:** | - |
| **Purpose:** | *To load the view after log in to user account.* |
| **Definitions/nested classes** | *get\_redirect\_url() 🡪 redirecting to the url pages based on if the user is logged in or out.* |

|  |  |
| --- | --- |
| **Class:** | LoginView(FormView) |
| **Import requirements:** | from **django.views.generic** import **FormView**  from **django.contrib.auth.forms** import **AuthenticationForm** |
| **Used In:** | - |
| **Purpose:** | *The login/sign in view.* |
| **Definitions/nested classes** | *get\_success\_url()*  *get\_context\_data()*  *form\_valid()* |

|  |  |
| --- | --- |
| **Class:** | LogoutView(RedirectView) |
| **Import requirements:** | from **django.views.generic** import **RedirectView** |
| **Used In:** | - |
| **Purpose:** | *View to display after logging out.* |
| **Definitions/nested classes** | *dispatch()* |

# Account Views

**FILE:** account\_viewsN.py   
**Location:** (~/web\_interface/account/)  
  
Optionally consider logging the software events by including the following:

import logging

log = logging.getLogger(\_\_name\_\_)

|  |  |
| --- | --- |
| **Class:** | MyAccountView(RestrictedView) |
| **Import requirements:** | from **django.utils.decorators** import **method\_decorator**  from **django.contrib.auth.decorators** import **login\_required** |
| **Used In:** | - |
| **Purpose:** | *Displays a page after “account\_homeN.html” after logging in.* |
| **Definitions/nested classes** | *-* |

|  |  |
| --- | --- |
| **Class:** | AcountRegisterForm(UserCreationForm) |
| **Import requirements:** | from **django.contrib.auth.forms** import **UserCreationForm**  from **django** import **forms** from **web\_interface.account** import **user\_countries**  from **django.contrib.auth.models** import **User**  from **django.urls** import **reverse\_lazy** |
| **Used In:** | AccountRegisterView class |
| **Purpose:** | *This class defines the user registration form.* |
| **Definitions/nested classes** | *\_\_init\_\_() 🡪 initialization and calling terms page.* |
| *Class Meta*  *🡪 to access and override the contents of inner class fields.* |

|  |  |
| --- | --- |
| **Class:** | AcountRegisterView(FormView) |
| **Import requirements:** | from **django.views.generic** import **FormView** from **django.urls** import **reverse\_lazy**  from **cloud\_copasi** import **settings**  from **django.http** import **HttpResponseRedirect**  AccountRegisterForm class |
| **Used In:** | - |
| **Purpose:** | *Used to create a view for the registration page.* |
| **Definitions/nested classes** | *get\_context\_data() 🡪 built-in django method to merge all the context data of parent classes with those of the current class. It is used for injecting data* |
| *dispatch()*  *🡪 looks at the request to determine whether it is a GET, POST, etc, and relays the request to a matching method if one is defined, or raises HttpResponseNotAllowed if not* |
| *form\_valid() 🡪 This method is called when valid form data has been POSTed* |

|  |  |
| --- | --- |
| **Class:** | AcountProfileForm(forms.Form) |
| **Import requirements:** | from **django** import **forms** |
| **Used In:** | AccountProfileView class |
| **Purpose:** | *Display the profile view.* |
| **Definitions/nested classes** | *-* |

|  |  |
| --- | --- |
| **Class:** | AccountProfileView(RestrictedFormView) |
| **Import requirements:** | from **django.utils.decorators** import **method\_decorator**  from **django.contrib.auth.decorators** import **login\_required**  from **django.urls** import **reverse\_lazy**  RestrictedFormView class |
| **Used In:** | - |
| **Purpose:** | *Display account profile options and calls out the html page account/profileN.html* |
| **Definitions/nested classes** | *form\_valid()* |

# Models

**FILE:** models.py   
**Location:** (~/web\_interface/)  
  
Optionally consider logging the software events by including the following:

import logging

log = logging.getLogger(\_\_name\_\_)

|  |  |
| --- | --- |
| **Class:** | Profile(models.Model) |
| **Import requirements:** | from django.db import models  from django.contrib.auth.models import User |
| **Used In:** | - |
| **Purpose:** | *Store additional profile information for a user* |
| **Definitions/nested classes** |  |

|  |  |
| --- | --- |
| **Class:** | AWSAccessKey(models.Model) |
| **Import requirements:** | from django.db import models  from django.contrib.auth.models import User  from django.core.validators import RegexValidator |
| **Used In:** | - |
| **Purpose:** | *Store AWS access key information.* |
| **Definitions/nested classes** | *\_\_unicode\_\_*  *Class Meta:* |

|  |  |
| --- | --- |
| **Class:** | VPC(models.Model) |
| **Import requirements:** | from django.db import models  AWSAccessKey class |
| **Used In:** | - |
| **Purpose:** | *Represents AWS VPC where jobs run.* |
| **Definitions/nested classes** | *Class Meta*  *\_\_unicode\_\_() get\_vpc()*  *get\_internet\_gateway() get\_route\_table() get\_master\_groupl()*  *get\_worker\_groupl()*  *get\_status()*  *get\_keypair()* |

|  |  |
| --- | --- |
| **Class:** | CondolPool(models.Model) |
| **Import requirements:** | from django.db import models  from django.contrib.auth.models import User  from cloud\_copasi.web\_interface.fields import UUIDField |
| **Used In:** | EC2Pool class |
| **Purpose:** | *Abstract place holder for either an EC2 condor pool, or some other Bosco pool.* |
| **Definitions/nested classes** | *Class Meta*  *Get\_pool\_type()*  *\_\_unicode\_\_() get\_pool\_type\_display\_true() get\_returning\_tasks() get\_recast\_pool()* |

|  |  |
| --- | --- |
| **Class:** | EC2Pool(CondorPool) |
| **Import requirements:** | VPC class  CondorPool class |
| **Used In:** | - |
| **Purpose:** | *Stores info about all the EC2 instances making up a condor pool.* |
| **Definitions/nested classes** | *Class Meta*  *get\_key\_pair()*  *get\_count() get\_queue\_name() get\_alarm\_notify\_topic() get\_health()* |

# CSS

**FILE:** baseN.css   
**Location:** (~/html5up/css/)

|  |  |
| --- | --- |
| **Name:** | header |
| **Type:** | Built-in HTML Tag |
| **Used In:** | baseN.html |
| **Purpose:** | *It encloses the navigation bar and keep its position to remain on top of all other components while scrolling down the page.* |

|  |  |
| --- | --- |
| **Name:** | navbar-light |
| **Type:** | Class |
| **Used In:** | baseN.html |
| **Purpose:** | *This is a Bootstrap class of navigation bar. It defines the styling of navigation bar. It requires Bootsrap to be included in the head of html file.* |

|  |  |
| --- | --- |
| **Name:** | container |
| **Type:** | Class |
| **Used In:** | baseN.html |
| **Purpose:** | *This is a Bootstrap class to keep the data at the center of page. The minimum width of the content size is set to 69em (em refers to the CSS unit. 69em means means 69 times the current value).*  *It requires Bootsrap to be included in the head of html file.* |

|  |  |
| --- | --- |
| **Name:** | banner |
| **Type:** | ID |
| **Used In:** | homeN.html |
| **Purpose:** | It includes styling of the banner image (cloud image) display on the homeN.html page. The image is retrieved from ~/html5up/static/images/ directory.  *Styling for h1 and h3 are done separately.* |

|  |  |
| --- | --- |
| **Name:** | features |
| **Type:** | Class |
| **Used In:** | homeN.html |
| **Purpose:** | *This block includes the styling of three main features mentioned at the home page, between the cloud copasi banner and the description.* |

|  |  |
| --- | --- |
| **Name:** | description |
| **Type:** | Class |
| **Used In:** | homeN.html |
| **Purpose:** | *This block includes the styling of cloud copasi description mentioned at the homepage.* |

|  |  |
| --- | --- |
| **Name:** | footer |
| **Type:** | Class |
| **Used In:** | baseN.html |
| **Purpose:** | *This block includes the styling of footer at the bottom of page. Footer appears on all of html pages.* |

|  |  |
| --- | --- |
| **Name:** |  |
| **Type:** |  |
| **Used In:** |  |
| **Purpose:** |  |