OCDE - 405 Found

Generated by Doxygen 1.8.17

Chapter 1

Namespace Index

1.1 Packages

Here are the packages with brief descriptions (if available):

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

Chapter 3

Class Index

3.1 Class List

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

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

Chapter 5

Namespace Documentation

5.1 blog Namespace Reference

Namespaces

- admin
- apps
- forms
- migrations
- models
- tests
- urls
- views

5.2 blog.admin Namespace Reference

5.3 blog.apps Namespace Reference

Classes

- class BlogConfig

5.4 blog.forms Namespace Reference

Classes

- class CreateFileForm

    Custom Form for folder creation linking it to the user.

5.5 blog.migrations Namespace Reference

Namespaces

- 0001_initial
- 0002_auto_20201206_1527

5.6 blog.migrations.0001_initial Namespace Reference

Classes

- class Migration

5.7 blog.migrations.0002_auto_20201206_1527 Namespace Reference

Classes

- class Migration

5.8 blog.models Namespace Reference

Classes

- class Post

  Encapsulation of the Post Model which represents files in the CodeFiles Directory.

5.9 blog.tests Namespace Reference

5.10 blog.urls Namespace Reference

Variables

- list urlpatterns

5.10.1 Variable Documentation

5.10.1.1 urlpatterns

```
list urlpatterns
```

Initial value:

```
=  [
    path(”, views.home, name=’blog-home’),
    path(’about/’, views.about, name=’blog-about’),
    path(’folderview/<path:folderpath>’,views.folderview,name=’blog-folderview’),
    path(’create/<path:folderpath>’,views.create_file_in_folder,name=’blog-create_file_in_folder’),
]
```

5.11 blog.views Namespace Reference

Classes

- class node

  Documentation for class node class node represents a node in the tree structure of code_directory variable stat distinguishes between file and folder.

Functions

- def about (request)
- def builder (top)
- def create (request, folderpath)
- def create_file_in_folder (request, folderpath)

  Used for Creating files and sub-folders as well as deleting sub-folders this function takes two parameters , the folderpath and the request to create file or folder or delete folder According to the option creates the prompt page for name if needed to create ,else redirects to home Contains the logic for creating and deletion of folders,files the option whether to create or delete is handled accordingly by passing a special character in the folderpath.
- def detail (request)
- def folderview (request, folderpath)

  This function is called on clicking hyperlinks of folders in the directory view.
- def home (request)

  This function displays the home page.

5.11.1 Function Documentation

5.11.1.1 about()

```
def blog.views.about (
              request )
```

5.11.1.2 builder()

```
def blog.views.builder (
              top )
```

5.11.1.3 create()

```
def blog.views.create (
              request,
              folderpath )
```

5.11.1.4 create_file_in_folder()

```
def blog.views.create_file_in_folder (
              request,
              folderpath )
```

Used for Creating files and sub-folders as well as deleting sub-folders this function takes two parameters , the folderpath and the request to create file or folder or delete folder According to the option creates the prompt page for name if needed to create ,else redirects to home Contains the logic for creating and deletion of folders,files the option whether to create or delete is handled accordingly by passing a special character in the folderpath.

5.11.1.5 detail()

```
def blog.views.detail (
              request )
```

5.11.1.6 folderview()

```
def blog.views.folderview (
              request,
              folderpath )
```

This function is called on clicking hyperlinks of folders in the directory view.

It facilitates the interface for creation of sub-folders and files as well as deletion of the folder

5.11.1.7 home()

```
def blog.views.home (
              request )
```

This function displays the home page.

The Home page consists of the directory structure which is achieved through the builder function.

5.12 django_project Namespace Reference

Namespaces

- asgi
- settings
- urls
- wsgi

5.13 django_project.asgi Namespace Reference

Variables

- application = get_asgi_application()

5.13.1 Detailed Description

```
ASGI config for django_project project.

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

For more information on this file, see
https://docs.djangoproject.com/en/3.1/howto/deployment/asgi/
```

5.13.2 Variable Documentation

5.13.2.1 application

```
application = get_asgi_application()
```

5.14 django_project.settings Namespace Reference

Variables

- list ALLOWED_HOSTS = [ ]
- list AUTH_PASSWORD_VALIDATORS
- BASE_DIR = Path(__file__).resolve().parent.parent
- string CRISPY_TEMPLATE_PACK = 'bootstrap4'
- dictionary DATABASES
- bool DEBUG = True
- list INSTALLED_APPS
- string LANGUAGE_CODE = 'en-us'
- string LOGIN_REDIRECT_URL = 'blog-home'
- MEDIA_ROOT = os.path.join(BASE_DIR, 'media')

- string MEDIA_URL = '/media/'
- list MIDDLEWARE
- string ROOT_URLCONF = 'django_project.urls'
- string SECRET_KEY = 'x2(kig$+@riq5l8&d∗cj∗dyc&6q31io9_50!hrubfl!_=4pxb@'
- string STATIC_URL = '/static/'
- list TEMPLATES
- string TIME_ZONE = 'UTC'
- bool USE_I18N = True
- bool USE_L10N = True
- bool USE_TZ = True
- string WSGI_APPLICATION = 'django_project.wsgi.application'

5.14.1 Detailed Description

```
Django settings for django_project project.

Generated by 'django-admin startproject' using Django 3.1.2.

For more information on this file, see
https://docs.djangoproject.com/en/3.1/topics/settings/

For the full list of settings and their values, see
https://docs.djangoproject.com/en/3.1/ref/settings/
```

5.14.2 Variable Documentation

5.14.2.1 ALLOWED_HOSTS

```
list ALLOWED_HOSTS = []
```

5.14.2.2 AUTH_PASSWORD_VALIDATORS

```
list AUTH_PASSWORD_VALIDATORS
```

Initial value:

```
=  [
    {
        'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]
```

5.14.2.3 BASE_DIR

```
BASE_DIR = Path(__file__).resolve().parent.parent
```

5.14.2.4 CRISPY_TEMPLATE_PACK

```
string CRISPY_TEMPLATE_PACK = 'bootstrap4'
```

5.14.2.5 DATABASES

```
dictionary DATABASES
```

Initial value:

```
=  {
    ’default’: {
        ’ENGINE’: ’django.db.backends.sqlite3’,
        ’NAME’: BASE_DIR / ’db.sqlite3’,
    }
}
```

5.14.2.6 DEBUG

```
bool DEBUG = True
```

5.14.2.7 INSTALLED_APPS

```
list INSTALLED_APPS
```

Initial value:

```
=  [
    ’blog.apps.BlogConfig’,
    ’users.apps.UsersConfig’,
    ’crispy_forms’,
    ’django_ace’,
    ’django.contrib.admin’,
    ’django.contrib.auth’,
    ’django.contrib.contenttypes’,
    ’django.contrib.sessions’,
    ’django.contrib.messages’,
    ’django.contrib.staticfiles’,
]
```

5.14.2.8 LANGUAGE_CODE

```
string LANGUAGE_CODE = 'en-us'
```

5.14.2.9 LOGIN_REDIRECT_URL

```
string LOGIN_REDIRECT_URL = 'blog-home'
```

5.14.2.10 MEDIA_ROOT

```
MEDIA_ROOT = os.path.join(BASE_DIR, 'media')
```

5.14.2.11 MEDIA_URL

```
string MEDIA_URL = '/media/'
```

5.14.2.12 MIDDLEWARE

```
list MIDDLEWARE
```

Initial value:

```
=  [
    ’django.middleware.security.SecurityMiddleware’,
    ’django.contrib.sessions.middleware.SessionMiddleware’,
    ’django.middleware.common.CommonMiddleware’,
    ’django.middleware.csrf.CsrfViewMiddleware’,
    ’django.contrib.auth.middleware.AuthenticationMiddleware’,
    ’django.contrib.messages.middleware.MessageMiddleware’,
    ’django.middleware.clickjacking.XFrameOptionsMiddleware’,
]
```

5.14.2.13 ROOT_URLCONF

```
string ROOT_URLCONF = 'django_project.urls'
```

5.14.2.14 SECRET_KEY

```
string SECRET_KEY = 'x2(kig$+@riq5l8&d*cj*dyc&6q31io9_50!hrubfl!_=4pxb@'
```

5.14.2.15 STATIC_URL

```
string STATIC_URL = '/static/'
```

5.14.2.16 TEMPLATES

```
list TEMPLATES
```

Initial value:

```
=  [
    {
        ’BACKEND’: ’django.template.backends.django.DjangoTemplates’,
        ’DIRS’: [],
        ’APP_DIRS’: True,
        ’OPTIONS’: {
            ’context_processors’: [
                ’django.template.context_processors.debug’,
                ’django.template.context_processors.request’,
                ’django.contrib.auth.context_processors.auth’,
                ’django.contrib.messages.context_processors.messages’,
            ],
        },
    },
]
```

5.14.2.17 TIME_ZONE

```
string TIME_ZONE = 'UTC'
```

5.14.2.18 USE_I18N

```
bool USE_I18N = True
```

5.14.2.19 USE_L10N

```
bool USE_L10N = True
```

5.14.2.20 USE_TZ

```
bool USE_TZ = True
```

5.14.2.21 WSGI_APPLICATION

```
string WSGI_APPLICATION = 'django_project.wsgi.application'
```

5.15 django_project.urls Namespace Reference

Variables

- document_root
- MEDIA_URL
- list urlpatterns

5.15.1 Variable Documentation

5.15.1.1 document_root

```
document_root
```

5.15.1.2 MEDIA_URL

```
MEDIA_URL
```

5.15.1.3 urlpatterns

```
list urlpatterns
```

Initial value:

```
=  [
    path(’admin/’, admin.site.urls),
    path(’register/’, user_views.register, name=’register’),
    path(’profile/’, user_views.profile, name=’profile’),
    path(’login/’, auth_views.LoginView.as_view(template_name=’users/login.html’), name=’login’),
    path(’logout/’, auth_views.LogoutView.as_view(template_name=’users/logout.html’), name=’logout’),
    path(’ide/’, user_views.simple, name=’ide’),
    path(”, include(’blog.urls’)),
    path(’CodeFiles/<path:filepath>’,user_views.codeviewer, name=’codeviewer’),
]
```

5.16 django_project.wsgi Namespace Reference

Variables

- application = get_wsgi_application()

5.16.1 Detailed Description

```
WSGI config for django_project 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/3.1/howto/deployment/wsgi/
```

5.16.2 Variable Documentation

5.16.2.1 application

```
application = get_wsgi_application()
```

5.17 manage Namespace Reference

Functions

- def main ()

5.17.1 Detailed Description

```
Django's command-line utility for administrative tasks.
```

5.17.2 Function Documentation

5.17.2.1 main()

```
def manage.main ( )
```

```
Run administrative tasks.
```

5.18 ok Namespace Reference

5.19 test Namespace Reference

Variables

- a
- b

5.19.1 Variable Documentation

5.19.1.1 a

```
a
```

5.19.1.2 b

```
b
```

5.20 time Namespace Reference

Functions

- def f (a)

Variables

- a = input()

5.20.1 Function Documentation

5.20.1.1 f()

```
def time.f (
              a )
```

5.20.2 Variable Documentation

5.20.2.1 a

```
a = input()
```

5.21 users Namespace Reference

Namespaces

- admin
- apps
- forms
- migrations
- models
- signals
- tests
- views

5.22 users.admin Namespace Reference

5.23 users.apps Namespace Reference

Classes

- class UsersConfig

5.24 users.forms Namespace Reference

Classes

- class ProfileUpdateForm
- class SnippetForm
- class UserRegisterForm
- class UserUpdateForm

5.25 users.migrations Namespace Reference

Namespaces

- 0001_initial
- 0002_snippet

5.26 users.migrations.0001_initial Namespace Reference

Classes

- class Migration

5.27 users.migrations.0002_snippet Namespace Reference

Classes

- class Migration

5.28 users.models Namespace Reference

Classes

- class Profile

  This is an encapsulation of the User Profile Model which represents each user in the server uniquely.
- class Snippet

  This class is an encapsulation of the editor.

5.29 users.signals Namespace Reference

Functions

- def create_profile (sender, instance, created, **kwargs)
- def save_profile (sender, instance, **kwargs)

Variables

- post_save
- sender

5.29.1 Function Documentation

5.29.1.1 create_profile()

```
def users.signals.create_profile (
              sender,
              instance,
              created,
             ** kwargs )
```

5.29.1.2 save_profile()

```
def users.signals.save_profile (
              sender,
              instance,
             ** kwargs )
```

5.29.2 Variable Documentation

5.29.2.1 post_save

```
post_save
```

5.29.2.2 sender

```
sender
```

5.30 users.tests Namespace Reference

5.31 users.views Namespace Reference

Classes

- class node

  Documentation for class node Class node represents a node in the tree structure of code_directory.

Functions

- def builder (top)

  This function builds the whole directory structure in a bredth-first search manner where we iterate through the tree of directory with each file representing a leaf in the graph.
- def codeviewer (request, filepath)

  This function is called when the user clicks on file in the tree structure of the directory.
- def profile (request)

  This Function lets the user update its personal information including username and profile picture.
- def register (request)

  This function registers the details of a new user in the database via a POST request.
- def simple (request)

  This function is called when the user clicks on the editor option.

Variables

- top = node('c_directory','dir')

5.31.1 Function Documentation

5.31.1.1 builder()

```
def users.views.builder (
              top )
```

This function builds the whole directory structure in a bredth-first search manner where we iterate through the tree of directory with each file representing a leaf in the graph.

5.31.1.2 codeviewer()

```
def users.views.codeviewer (
              request,
              filepath )
```

This function is called when the user clicks on file in the tree structure of the directory.

It then redirects to the editor with the respective code file opened. Here, the user has both options of saving and running the code. This function supports the execution of programs in Java,C++ and Python.

5.31.1.3 profile()

```
def users.views.profile (
              request )
```

This Function lets the user update its personal information including username and profile picture.

5.31.1.4 register()

```
def users.views.register (
              request )
```

This function registers the details of a new user in the database via a POST request.

5.31.1.5 simple()

```
def users.views.simple (
              request )
```

This function is called when the user clicks on the editor option.

This function is for the compilation and running of temporary files in C++ only. The save functionality does not work here and the code is deleted after the user exits the site.

5.31.2 Variable Documentation

5.31.2.1 top

```
top = node('c_directory','dir')
```

Chapter 6

Class Documentation

6.1 BlogConfig Class Reference

Inheritance diagram for BlogConfig:

Collaboration diagram for BlogConfig:

Static Public Attributes

- string name = 'blog'

6.1.1 Member Data Documentation

6.1.1.1 name

```
string name = 'blog'  [static]
```

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

- blog/apps.py

6.2 CreateFileForm Class Reference

Custom Form for folder creation linking it to the user.

Inheritance diagram for CreateFileForm:

Collaboration diagram for CreateFileForm:

Classes

- class Meta

  Standard Entries.

6.2.1 Detailed Description

Custom Form for folder creation linking it to the user.

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

- blog/forms.py

6.3 SnippetForm.Meta Class Reference

Static Public Attributes

- tuple exclude = ()
- model = Snippet
- dictionary widgets

6.3.1 Member Data Documentation

6.3.1.1 exclude

```
tuple exclude = ()  [static]
```

6.3.1.2 model

```
model = Snippet  [static]
```

6.3.1.3 widgets

```
dictionary widgets  [static]
```

Initial value:

```
=  {
             "text": AceWidget(mode='javascript', theme='monokai'),
        }
```

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

- users/forms.py

6.4 UserRegisterForm.Meta Class Reference

Static Public Attributes

- list fields = ['username', 'email', 'password1', 'password2']
- model = User

6.4.1 Member Data Documentation

6.4.1.1 fields

```
list fields = ['username', 'email', 'password1', 'password2']  [static]
```

6.4.1.2 model

```
model = User  [static]
```

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

- users/forms.py

6.5 Snippet.Meta Class Reference

Static Public Attributes

- tuple ordering = ('-created_at',)

6.5.1 Member Data Documentation

6.5.1.1 ordering

```
tuple ordering = ('-created_at', )  [static]
```

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

- users/models.py

6.6 CreateFileForm.Meta Class Reference

Standard Entries.

Static Public Attributes

- list fields = ['title', 'author']
- model = Post

6.6.1 Detailed Description

Standard Entries.

6.6.2 Member Data Documentation

6.6.2.1 fields

```
list fields = ['title', 'author']  [static]
```

6.6.2.2 model

```
model = Post  [static]
```

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

- blog/forms.py

6.7 UserUpdateForm.Meta Class Reference

Static Public Attributes

- list fields = ['username', 'email']
- model = User

6.7.1 Member Data Documentation

6.7.1.1 fields

```
list fields = ['username', 'email']  [static]
```

6.7.1.2 model

```
model = User  [static]
```

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

- users/forms.py

6.8 ProfileUpdateForm.Meta Class Reference

Static Public Attributes

- list fields = ['image']
- model = Profile

6.8.1 Member Data Documentation

6.8.1.1 fields

```
list fields = ['image']  [static]
```

6.8.1.2 model

```
model = Profile  [static]
```

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

- users/forms.py

6.9 Migration Class Reference

Inheritance diagram for Migration:

Collaboration diagram for Migration:

Static Public Attributes

- list dependencies
- bool initial = True
- list operations

6.9.1 Member Data Documentation

6.9.1.1 dependencies

```
list dependencies [static]
```

Initial value:

```
= [
        migrations.swappable_dependency(settings.AUTH_USER_MODEL),
    ]
```

6.9.1.2 initial

```
bool initial = True [static]
```

6.9.1.3 operations

```
list operations [static]
```

Initial value:

```
= [
        migrations.CreateModel(
            name='Post',
            fields=[
                ('id', models.AutoField(auto_created=True, primary_key=True, serialize=False,
       verbose_name='ID')),
                ('title', models.CharField(max_length=100)),
                #('content', models.TextField()),
                #('date_posted', models.DateTimeField(default=django.utils.timezone.now)),
                ('author', models.ForeignKey(on_delete=django.db.models.deletion.CASCADE,
       to=settings.AUTH_USER_MODEL)),
            ],
        ),
    ]
```

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

- blog/migrations/0001_initial.py

6.10 Migration Class Reference

Inheritance diagram for Migration:

Collaboration diagram for Migration:

Static Public Attributes

- list dependencies
- bool initial = True
- list operations

6.10.1 Member Data Documentation

6.10.1.1 dependencies

```
list dependencies  [static]
```

Initial value:

```
=  [
        migrations.swappable_dependency(settings.AUTH_USER_MODEL),
    ]
```

6.10.1.2 initial

```
bool initial = True  [static]
```

6.10.1.3 operations

```
list operations  [static]
```

Initial value:

```
=  [
        migrations.CreateModel(
            name='Profile',
            fields=[
                ('id', models.AutoField(auto_created=True, primary_key=True, serialize=False,
       verbose_name='ID')),
                ('image', models.ImageField(default='default.jpg', upload_to='profile_pics')),
                ('user', models.OneToOneField(on_delete=django.db.models.deletion.CASCADE,
       to=settings.AUTH_USER_MODEL)),
            ],
        ),
    ]
```

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

- users/migrations/0001_initial.py

6.11 Migration Class Reference

Inheritance diagram for Migration:

Collaboration diagram for Migration:

Static Public Attributes

- list dependencies
- list operations

6.11.1 Member Data Documentation

6.11.1.1 dependencies

```
list dependencies  [static]
```

Initial value:

```
=  [
        ('users', '0001_initial'),
    ]
```

6.11.1.2 operations

```
list operations  [static]
```

Initial value:

```
=  [
        migrations.CreateModel(
            name='Snippet',
            fields=[
                ('id', models.AutoField(auto_created=True, primary_key=True, serialize=False,
       verbose_name='ID')),
                ('text', models.TextField()),
                ('created_at', models.DateTimeField(auto_now_add=True)),
            ],
            options={
                'ordering': ('-created_at',),
            },
        ),
    ]
```

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

- users/migrations/0002_snippet.py

6.12 Migration Class Reference

Inheritance diagram for Migration:

Collaboration diagram for Migration:

Static Public Attributes

- list dependencies
- list operations

6.12.1 Member Data Documentation

6.12.1.1 dependencies

```
list dependencies [static]
```

Initial value:

```
= [
        ('blog', '0001_initial'),
    ]
```

6.12.1.2 operations

```
list operations [static]
```

Initial value:

```
= [
        migrations.AddField(
            model_name='post',
            name='content',
            field=models.TextField(default=0),
            preserve_default=False,
        ),
        migrations.AddField(
            model_name='post',
            name='date_posted',
            field=models.DateTimeField(default=django.utils.timezone.now),
        ),
    ]
```

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

- blog/migrations/0002_auto_20201206_1527.py

6.13 node Class Reference

Documentation for class node Class node represents a node in the tree structure of code_directory.

Public Member Functions

- def __init__ (self, name, stat)
- def add_child (self, child_node)

  adding files as children to the parent folder

Public Attributes

- child
- name
- path
- stat

6.13.1 Detailed Description

Documentation for class node Class node represents a node in the tree structure of code_directory.

Variable stat distinguishes between file and folder.

6.13.2 Constructor & Destructor Documentation

6.13.2.1 __init__()

```
def __init__ (
              self,
              name,
              stat )
```

6.13.3 Member Function Documentation

6.13.3.1 add_child()

```
def add_child (
              self,
              child_node )
```

adding files as children to the parent folder

6.13.4 Member Data Documentation

6.13.4.1 child

```
child
```

6.13.4.2 name

```
name
```

6.13.4.3 path

```
path
```

6.13.4.4 stat

```
stat
```

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

- users/views.py

6.14 node Class Reference

Documentation for class node class node represents a node in the tree structure of code_directory variable stat distinguishes between file and folder.

Public Member Functions

- def __init__ (self, name, stat, path)
- def add_child (self, child_node)

Public Attributes

- child
- name
- path
- stat

6.14.1 Detailed Description

Documentation for class node class node represents a node in the tree structure of code_directory variable stat distinguishes between file and folder.

6.14.2 Constructor & Destructor Documentation

6.14.2.1 __init__()

```
def __init__ (
              self,
              name,
              stat,
              path )
```

6.14.3 Member Function Documentation

6.14.3.1 add_child()

```
def add_child (
              self,
              child_node )
```

6.14.4 Member Data Documentation

6.14.4.1 child

```
child
```

6.14.4.2 name

```
name
```

6.14.4.3 path

```
path
```

6.14.4.4 stat

```
stat
```

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

- blog/views.py

6.15 Post Class Reference

Encapsulation of the Post Model which represents files in the CodeFiles Directory.

Inheritance diagram for Post:

Collaboration diagram for Post:

Public Member Functions

- def __str__ (self)

  String Representation.
- def get_absolute_url (self)

  url representation for the file detail view

Static Public Attributes

- author = models.ForeignKey(User, on_delete=models.CASCADE)
- content = models.TextField()
- date_posted = models.DateTimeField(default=timezone.now)
- title = models.CharField(max_length=100)

6.15.1 Detailed Description

Encapsulation of the Post Model which represents files in the CodeFiles Directory.

6.15.2 Member Function Documentation

6.15.2.1 __str__()

```
def __str__ (
              self )
```

String Representation.

6.15.2.2 get_absolute_url()

```
def get_absolute_url (
              self )
```

url representation for the file detail view

6.15.3 Member Data Documentation

6.15.3.1 author

```
author = models.ForeignKey(User, on_delete=models.CASCADE)  [static]
```

6.15.3.2 content

```
content = models.TextField()  [static]
```

6.15.3.3 date_posted

```
date_posted = models.DateTimeField(default=timezone.now)  [static]
```

6.15.3.4 title

```
title = models.CharField(max_length=100)  [static]
```

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

- blog/models.py

6.16 Profile Class Reference

This is an encapsulation of the User Profile Model which represents each user in the server uniquely.

Inheritance diagram for Profile:

Collaboration diagram for Profile:

Public Member Functions

- def __str__ (self)
- def save (self, ∗args, ∗∗kawrgs)

Static Public Attributes

- image = models.ImageField(default='default.jpg', upload_to='profile_pics')
- user = models.OneToOneField(User, on_delete=models.CASCADE)

6.16.1 Detailed Description

This is an encapsulation of the User Profile Model which represents each user in the server uniquely.

6.16.2 Member Function Documentation

6.16.2.1 __str__()

```
def __str__ (
              self )
```

6.16.2.2 save()

```
def save (
              self,
             * args,
             ** kawrgs )
```

6.16.3 Member Data Documentation

6.16.3.1 image

```
image = models.ImageField(default='default.jpg', upload_to='profile_pics')  [static]
```

6.16.3.2 user

```
user = models.OneToOneField(User, on_delete=models.CASCADE)  [static]
```

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

- users/models.py

6.17 ProfileUpdateForm Class Reference

Inheritance diagram for ProfileUpdateForm:

Collaboration diagram for ProfileUpdateForm:

Classes

- class Meta

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

- users/forms.py

6.18 Snippet Class Reference

This class is an encapsulation of the editor.

Inheritance diagram for Snippet:

Collaboration diagram for Snippet:

Classes

- class Meta

Static Public Attributes

- created_at = models.DateTimeField(auto_now_add=True)
- text = models.TextField()

6.18.1 Detailed Description

This class is an encapsulation of the editor.

6.18.2 Member Data Documentation

6.18.2.1 created_at

```
created_at = models.DateTimeField(auto_now_add=True)  [static]
```

6.18.2.2 text

```
text = models.TextField()  [static]
```

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

- users/models.py

6.19 SnippetForm Class Reference

Inheritance diagram for SnippetForm:

Collaboration diagram for SnippetForm:

Classes

- class Meta

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

- users/forms.py

6.20 Test Class Reference

Static Public Member Functions

- static void main (String[ ] args)

6.20.1 Member Function Documentation

6.20.1.1 main()

```
static void main (
             String[ ] args )  [static]
```

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

- CodeFiles/22/Test.java

6.21 UserRegisterForm Class Reference

Inheritance diagram for UserRegisterForm:

Collaboration diagram for UserRegisterForm:

Classes

- class Meta

Static Public Attributes

- email = forms.EmailField()

6.21.1 Member Data Documentation

6.21.1.1 email

```
email = forms.EmailField()  [static]
```

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

- users/forms.py

6.22 UsersConfig Class Reference

Inheritance diagram for UsersConfig:

Collaboration diagram for UsersConfig:

Public Member Functions

- def ready (self)

Static Public Attributes

- string name = 'users'

6.22.1 Member Function Documentation

6.22.1.1 ready()

```
def ready (
              self )
```

6.22.2 Member Data Documentation

6.22.2.1 name

```
string name = 'users'  [static]
```

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

- users/apps.py

6.23 UserUpdateForm Class Reference

Inheritance diagram for UserUpdateForm:

Collaboration diagram for UserUpdateForm:

Classes

- class Meta

Static Public Attributes

- email = forms.EmailField()

6.23.1 Member Data Documentation

6.23.1.1 email

```
email = forms.EmailField()  [static]
```

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

- users/forms.py

Chapter 7

File Documentation

7.1 blog/__init__.py File Reference

Namespaces

- blog

7.2 blog/migrations/__init__.py File Reference

Namespaces

- blog.migrations

7.3 django_project/__init__.py File Reference

Namespaces

- django_project

7.4 users/__init__.py File Reference

Namespaces

- users

7.5 users/migrations/__init__.py File Reference

Namespaces

- users.migrations

7.6 blog/admin.py File Reference

Namespaces

- blog.admin

7.7 users/admin.py File Reference

Namespaces

- users.admin

7.8 blog/apps.py File Reference

Classes

- class BlogConfig

Namespaces

- blog.apps

7.9 users/apps.py File Reference

Classes

- class UsersConfig

Namespaces

- users.apps

7.10 blog/forms.py File Reference

Classes

- class CreateFileForm

  Custom Form for folder creation linking it to the user.
- class CreateFileForm.Meta

  Standard Entries.

Namespaces

- blog.forms

7.11 users/forms.py File Reference

Classes

- class SnippetForm.Meta
- class ProfileUpdateForm.Meta
- class UserUpdateForm.Meta
- class UserRegisterForm.Meta
- class ProfileUpdateForm
- class SnippetForm
- class UserRegisterForm
- class UserUpdateForm

Namespaces

- users.forms

7.12 blog/migrations/0001_initial.py File Reference

Classes

- class Migration

Namespaces

- blog.migrations.0001_initial

7.13 users/migrations/0001_initial.py File Reference

Classes

- class Migration

Namespaces

- users.migrations.0001_initial

7.14 blog/migrations/0002_auto_20201206_1527.py File Reference

Classes

- class Migration

Namespaces

- blog.migrations.0002_auto_20201206_1527

7.15 blog/models.py File Reference

Classes

- class Post

  Encapsulation of the Post Model which represents files in the CodeFiles Directory.

Namespaces

- blog.models

7.16 users/models.py File Reference

Classes

- class Snippet.Meta
- class Profile

  This is an encapsulation of the User Profile Model which represents each user in the server uniquely.
- class Snippet

  This class is an encapsulation of the editor.

Namespaces

- users.models

7.17 blog/tests.py File Reference

Namespaces

- blog.tests

7.18 users/tests.py File Reference

Namespaces

- users.tests

7.19 blog/urls.py File Reference

Namespaces

- blog.urls

Variables

- list urlpatterns

7.20 django_project/urls.py File Reference

Namespaces

- django_project.urls

Variables

- document_root
- MEDIA_URL
- list urlpatterns

7.21 blog/views.py File Reference

Classes

- class node

  Documentation for class node class node represents a node in the tree structure of code_directory variable stat distinguishes between file and folder.

Namespaces

- blog.views

Functions

- def about (request)
- def builder (top)
- def create (request, folderpath)
- def create_file_in_folder (request, folderpath)

  Used for Creating files and sub-folders as well as deleting sub-folders this function takes two parameters , the folderpath and the request to create file or folder or delete folder According to the option creates the prompt page for name if needed to create ,else redirects to home Contains the logic for creating and deletion of folders,files the option whether to create or delete is handled accordingly by passing a special character in the folderpath.
- def detail (request)
- def folderview (request, folderpath)

  This function is called on clicking hyperlinks of folders in the directory view.
- def home (request)

  This function displays the home page.

7.22 users/views.py File Reference

Classes

- class node

  Documentation for class node Class node represents a node in the tree structure of code_directory.

Namespaces

- users.views

Functions

- def builder (top)

  This function builds the whole directory structure in a bredth-first search manner where we iterate through the tree of directory with each file representing a leaf in the graph.
- def codeviewer (request, filepath)

  This function is called when the user clicks on file in the tree structure of the directory.
- def profile (request)

  This Function lets the user update its personal information including username and profile picture.
- def register (request)

  This function registers the details of a new user in the database via a POST request.
- def simple (request)

  This function is called when the user clicks on the editor option.

Variables

- top = node('c_directory','dir')

7.23 CodeFiles/20/abc.cpp File Reference

```
#include <iostream>
```

Include dependency graph for abc.cpp:

7.24 CodeFiles/20/ele.cpp File Reference

```
#include <iostream>
```

Include dependency graph for ele.cpp:

Functions

- int main ()

7.24.1 Function Documentation

7.24.1.1 main()

```
int main ( )
```

7.25 CodeFiles/20/hello.cpp File Reference

7.26 CodeFiles/21/hello.cpp File Reference

7.27 CodeFiles/20/hello2.cpp File Reference

7.28 CodeFiles/20/test.cpp File Reference

```
#include <iostream>
```

Include dependency graph for test.cpp:

Functions

- int main ()

7.28.1 Function Documentation

7.28.1.1 main()

```
int main ( )
```

7.29 CodeFiles/21/test.cpp File Reference

`#include <iostream>`
Include dependency graph for test.cpp:

Functions

- int main ()

7.29.1 Function Documentation

7.29.1.1 main()

```
int main ( )
```

7.30 CodeFiles/test.cpp File Reference

`#include <iostream>`
Include dependency graph for test.cpp:

Functions

- int main ()

7.30.1 Function Documentation

7.30.1.1 main()

```
int main ( )
```

7.31 CodeFiles/20/time.py File Reference

Namespaces

- time

Functions

- def f (a)

Variables

- a = input()

7.32 CodeFiles/21/ok.py File Reference

Namespaces

- ok

7.33 CodeFiles/21/sample.cpp File Reference

```
#include <iostream>
```

Include dependency graph for sample.cpp:

Functions

- int main ()

7.33.1 Function Documentation

7.33.1.1 main()

```
int main ( )
```

7.34 CodeFiles/21/subfolder/heloo3.cpp File Reference

7.35 CodeFiles/21/subfolder/sub.cpp File Reference

7.36 CodeFiles/21/subfolder/subfile.cpp File Reference

7.37 CodeFiles/21/subfolder/try.cpp File Reference

7.38 CodeFiles/21/test.py File Reference

Namespaces

- test

Variables

- a
- b

7.39 CodeFiles/22/coool.cpp File Reference

7.40 CodeFiles/22/Test.java File Reference

Classes

- class Test

7.41 django_project/asgi.py File Reference

Namespaces

- django_project.asgi

Variables

- application = get_asgi_application()

7.42 django_project/settings.py File Reference

Namespaces

- django_project.settings

Variables

- list ALLOWED_HOSTS = [ ]
- list AUTH_PASSWORD_VALIDATORS
- BASE_DIR = Path(__file__).resolve().parent.parent
- string CRISPY_TEMPLATE_PACK = 'bootstrap4'
- dictionary DATABASES
- bool DEBUG = True
- list INSTALLED_APPS
- string LANGUAGE_CODE = 'en-us'
- string LOGIN_REDIRECT_URL = 'blog-home'
- MEDIA_ROOT = os.path.join(BASE_DIR, 'media')
- string MEDIA_URL = '/media/'
- list MIDDLEWARE
- string ROOT_URLCONF = 'django_project.urls'
- string SECRET_KEY = 'x2(kig$+@riq5l8&d∗cj∗dyc&6q31io9_50!hrubfl!_=4pxb@'
- string STATIC_URL = '/static/'
- list TEMPLATES
- string TIME_ZONE = 'UTC'
- bool USE_I18N = True
- bool USE_L10N = True
- bool USE_TZ = True
- string WSGI_APPLICATION = 'django_project.wsgi.application'

7.43 django_project/wsgi.py File Reference

Namespaces

- django_project.wsgi

Variables

- application = get_wsgi_application()

7.44 manage.py File Reference

Namespaces

- manage

Functions

- def main ()

7.45 users/migrations/0002_snippet.py File Reference

Classes

- class Migration

Namespaces

- users.migrations.0002_snippet

7.46 users/signals.py File Reference

Namespaces

- users.signals

Functions

- def create_profile (sender, instance, created, ∗∗kwargs)
- def save_profile (sender, instance, ∗∗kwargs)

Variables

- post_save
- sender

Index