

OCDE - 405 Found

Generated by Doxygen 1.8.17

1 Namespace Index	1
1.1 Packages	1
2 Hierarchical Index	3
2.1 Class Hierarchy	3
3 Class Index	5
3.1 Class List	5
4 File Index	7
4.1 File List	7
5 Namespace Documentation	9
5.1 blog Namespace Reference	9
5.2 blog.admin Namespace Reference	9
5.3 blog.apps Namespace Reference	9
5.4 blog.forms Namespace Reference	9
5.5 blog.migrations Namespace Reference	10
5.6 blog.migrations.0001_initial Namespace Reference	10
5.7 blog.migrations.0002_auto_20201206_1527 Namespace Reference	10
5.8 blog.models Namespace Reference	10
5.9 blog.tests Namespace Reference	10
5.10 blog.urls Namespace Reference	10
5.10.1 Variable Documentation	10
5.10.1.1 urlpatterns	11
5.11 blog.views Namespace Reference	11
5.11.1 Function Documentation	11
5.11.1.1 about()	11
5.11.1.2 builder()	12
5.11.1.3 create()	12
5.11.1.4 create_file_in_folder()	12
5.11.1.5 detail()	12
5.11.1.6 folderview()	12
5.11.1.7 home()	12
5.12 django_project Namespace Reference	13
5.13 django_project.asgi Namespace Reference	13
5.13.1 Detailed Description	13
5.13.2 Variable Documentation	13
5.13.2.1 application	13
5.14 django_project.settings Namespace Reference	13
5.14.1 Detailed Description	14
5.14.2 Variable Documentation	14
5.14.2.1 ALLOWED_HOSTS	14
5.14.2.2 AUTH_PASSWORD_VALIDATORS	14

5.14.2.3	BASE_DIR	15
5.14.2.4	CRISPY_TEMPLATE_PACK	15
5.14.2.5	DATABASES	15
5.14.2.6	DEBUG	15
5.14.2.7	INSTALLED_APPS	15
5.14.2.8	LANGUAGE_CODE	15
5.14.2.9	LOGIN_REDIRECT_URL	16
5.14.2.10	MEDIA_ROOT	16
5.14.2.11	MEDIA_URL	16
5.14.2.12	MIDDLEWARE	16
5.14.2.13	ROOT_URLCONF	16
5.14.2.14	SECRET_KEY	16
5.14.2.15	STATIC_URL	16
5.14.2.16	TEMPLATES	17
5.14.2.17	TIME_ZONE	17
5.14.2.18	USE_I18N	17
5.14.2.19	USE_L10N	17
5.14.2.20	USE_TZ	17
5.14.2.21	WSGI_APPLICATION	17
5.15	django_project.urls Namespace Reference	18
5.15.1	Variable Documentation	18
5.15.1.1	document_root	18
5.15.1.2	MEDIA_URL	18
5.15.1.3	urlpatterns	18
5.16	django_project.wsgi Namespace Reference	18
5.16.1	Detailed Description	19
5.16.2	Variable Documentation	19
5.16.2.1	application	19
5.17	manage Namespace Reference	19
5.17.1	Detailed Description	19
5.17.2	Function Documentation	19
5.17.2.1	main()	19
5.18	ok Namespace Reference	19
5.19	test Namespace Reference	19
5.19.1	Variable Documentation	20
5.19.1.1	a	20
5.19.1.2	b	20
5.20	time Namespace Reference	20
5.20.1	Function Documentation	20
5.20.1.1	f()	20
5.20.2	Variable Documentation	20
5.20.2.1	a	20

5.21 users Namespace Reference	21
5.22 users.admin Namespace Reference	21
5.23 users.apps Namespace Reference	21
5.24 users.forms Namespace Reference	21
5.25 users.migrations Namespace Reference	21
5.26 users.migrations.0001_initial Namespace Reference	21
5.27 users.migrations.0002_snippet Namespace Reference	22
5.28 users.models Namespace Reference	22
5.29 users.signals Namespace Reference	22
5.29.1 Function Documentation	22
5.29.1.1 create_profile()	22
5.29.1.2 save_profile()	23
5.29.2 Variable Documentation	23
5.29.2.1 post_save	23
5.29.2.2 sender	23
5.30 users.tests Namespace Reference	23
5.31 users.views Namespace Reference	23
5.31.1 Function Documentation	24
5.31.1.1 builder()	24
5.31.1.2 codeviewer()	24
5.31.1.3 profile()	24
5.31.1.4 register()	24
5.31.1.5 simple()	25
5.31.2 Variable Documentation	25
5.31.2.1 top	25
6 Class Documentation	27
6.1 BlogConfig Class Reference	27
6.1.1 Member Data Documentation	27
6.1.1.1 name	27
6.2 CreateFileForm Class Reference	27
6.2.1 Detailed Description	28
6.3 SnippetForm.Meta Class Reference	28
6.3.1 Member Data Documentation	28
6.3.1.1 exclude	28
6.3.1.2 model	28
6.3.1.3 widgets	28
6.4 UserRegisterForm.Meta Class Reference	29
6.4.1 Member Data Documentation	29
6.4.1.1 fields	29
6.4.1.2 model	29
6.5 Snippet.Meta Class Reference	29

6.5.1 Member Data Documentation	29
6.5.1.1 ordering	29
6.6 CreateFileForm.Meta Class Reference	30
6.6.1 Detailed Description	30
6.6.2 Member Data Documentation	30
6.6.2.1 fields	30
6.6.2.2 model	30
6.7 UserUpdateForm.Meta Class Reference	30
6.7.1 Member Data Documentation	30
6.7.1.1 fields	31
6.7.1.2 model	31
6.8 ProfileUpdateForm.Meta Class Reference	31
6.8.1 Member Data Documentation	31
6.8.1.1 fields	31
6.8.1.2 model	31
6.9 Migration Class Reference	31
6.9.1 Member Data Documentation	32
6.9.1.1 dependencies	32
6.9.1.2 initial	32
6.9.1.3 operations	32
6.10 Migration Class Reference	32
6.10.1 Member Data Documentation	33
6.10.1.1 dependencies	33
6.10.1.2 initial	33
6.10.1.3 operations	33
6.11 Migration Class Reference	33
6.11.1 Member Data Documentation	34
6.11.1.1 dependencies	34
6.11.1.2 operations	34
6.12 Migration Class Reference	34
6.12.1 Member Data Documentation	35
6.12.1.1 dependencies	35
6.12.1.2 operations	35
6.13 node Class Reference	35
6.13.1 Detailed Description	36
6.13.2 Constructor & Destructor Documentation	36
6.13.2.1 __init__()	36
6.13.3 Member Function Documentation	36
6.13.3.1 add_child()	36
6.13.4 Member Data Documentation	36
6.13.4.1 child	36
6.13.4.2 name	36

6.13.4.3 path	37
6.13.4.4 stat	37
6.14 node Class Reference	37
6.14.1 Detailed Description	37
6.14.2 Constructor & Destructor Documentation	37
6.14.2.1 __init__()	37
6.14.3 Member Function Documentation	38
6.14.3.1 add_child()	38
6.14.4 Member Data Documentation	38
6.14.4.1 child	38
6.14.4.2 name	38
6.14.4.3 path	38
6.14.4.4 stat	38
6.15 Post Class Reference	38
6.15.1 Detailed Description	39
6.15.2 Member Function Documentation	39
6.15.2.1 __str__()	39
6.15.2.2 get_absolute_url()	39
6.15.3 Member Data Documentation	39
6.15.3.1 author	39
6.15.3.2 content	40
6.15.3.3 date_posted	40
6.15.3.4 title	40
6.16 Profile Class Reference	40
6.16.1 Detailed Description	40
6.16.2 Member Function Documentation	40
6.16.2.1 __str__()	41
6.16.2.2 save()	41
6.16.3 Member Data Documentation	41
6.16.3.1 image	41
6.16.3.2 user	41
6.17 ProfileUpdateForm Class Reference	41
6.18 Snippet Class Reference	42
6.18.1 Detailed Description	42
6.18.2 Member Data Documentation	42
6.18.2.1 created_at	42
6.18.2.2 text	42
6.19 SnippetForm Class Reference	42
6.20 Test Class Reference	43
6.20.1 Member Function Documentation	43
6.20.1.1 main()	43
6.21 UserRegisterForm Class Reference	43

6.21.1 Member Data Documentation	44
6.21.1.1 email	44
6.22 UsersConfig Class Reference	44
6.22.1 Member Function Documentation	44
6.22.1.1 ready()	44
6.22.2 Member Data Documentation	44
6.22.2.1 name	45
6.23 UserUpdateForm Class Reference	45
6.23.1 Member Data Documentation	45
6.23.1.1 email	45
7 File Documentation	47
7.1 blog/__init__.py File Reference	47
7.2 blog/migrations/__init__.py File Reference	47
7.3 django_project/__init__.py File Reference	47
7.4 users/__init__.py File Reference	47
7.5 users/migrations/__init__.py File Reference	47
7.6 blog/admin.py File Reference	48
7.7 users/admin.py File Reference	48
7.8 blog/apps.py File Reference	48
7.9 users/apps.py File Reference	48
7.10 blog/forms.py File Reference	48
7.11 users/forms.py File Reference	49
7.12 blog/migrations/0001_initial.py File Reference	49
7.13 users/migrations/0001_initial.py File Reference	49
7.14 blog/migrations/0002_auto_20201206_1527.py File Reference	50
7.15 blog/models.py File Reference	50
7.16 users/models.py File Reference	50
7.17 blog/tests.py File Reference	50
7.18 users/tests.py File Reference	51
7.19 blog/urls.py File Reference	51
7.20 django_project/urls.py File Reference	51
7.21 blog/views.py File Reference	51
7.22 users/views.py File Reference	52
7.23 CodeFiles/20/abc.cpp File Reference	53
7.24 CodeFiles/20/ele.cpp File Reference	53
7.24.1 Function Documentation	53
7.24.1.1 main()	53
7.25 CodeFiles/20/hello.cpp File Reference	53
7.26 CodeFiles/21/hello.cpp File Reference	53
7.27 CodeFiles/20/hello2.cpp File Reference	53
7.28 CodeFiles/20/test.cpp File Reference	53

7.28.1 Function Documentation	53
7.28.1.1 main()	54
7.29 CodeFiles/21/test.cpp File Reference	54
7.29.1 Function Documentation	54
7.29.1.1 main()	54
7.30 CodeFiles/test.cpp File Reference	54
7.30.1 Function Documentation	54
7.30.1.1 main()	54
7.31 CodeFiles/20/time.py File Reference	55
7.32 CodeFiles/21/ok.py File Reference	55
7.33 CodeFiles/21/sample.cpp File Reference	55
7.33.1 Function Documentation	55
7.33.1.1 main()	55
7.34 CodeFiles/21/subfolder/hello3.cpp File Reference	56
7.35 CodeFiles/21/subfolder/sub.cpp File Reference	56
7.36 CodeFiles/21/subfolder/subfile.cpp File Reference	56
7.37 CodeFiles/21/subfolder/try.cpp File Reference	56
7.38 CodeFiles/21/test.py File Reference	56
7.39 CodeFiles/22/coool.cpp File Reference	56
7.40 CodeFiles/22/Test.java File Reference	56
7.41 django_project/asgi.py File Reference	56
7.42 django_project/settings.py File Reference	56
7.43 django_project/wsgi.py File Reference	57
7.44 manage.py File Reference	57
7.45 users/migrations/0002_snippet.py File Reference	58
7.46 users/signals.py File Reference	58
Index	59

Chapter 1

Namespace Index

1.1 Packages

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

blog	9
blog.admin	9
blog.apps	9
blog.forms	9
blog.migrations	10
blog.migrations.0001_initial	10
blog.migrations.0002_auto_20201206_1527	10
blog.models	10
blog.tests	10
blog.urls	10
blog.views	11
django_project	13
django_project.asgi	13
django_project.settings	13
django_project.urls	18
django_project.wsgi	18
manage	19
ok	19
test	19
time	20
users	21
users.admin	21
users.apps	21
users.forms	21
users.migrations	21
users.migrations.0001_initial	21
users.migrations.0002_snippet	22
users.models	22
users.signals	22
users.tests	23
users.views	23

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

SnippetForm.Meta	28
UserRegisterForm.Meta	29
Snippet.Meta	29
CreateFileForm.Meta	30
UserUpdateForm.Meta	30
ProfileUpdateForm.Meta	31
Migration	
Migration	31
Migration	34
Migration	32
Migration	33
Model	
Post	38
Profile	40
Snippet	42
ModelForm	
CreateFileForm	27
ProfileUpdateForm	41
SnippetForm	42
UserUpdateForm	45
node	35
node	37
Test	43
AppConfig	
BlogConfig	27
UsersConfig	44
UserCreationForm	
UserRegisterForm	43

Chapter 3

Class Index

3.1 Class List

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

BlogConfig	27
CreateFileForm	
Custom Form for folder creation linking it to the user	27
SnippetForm.Meta	28
UserRegisterForm.Meta	29
Snippet.Meta	29
CreateFileForm.Meta	
Standard Entries	30
UserUpdateForm.Meta	30
ProfileUpdateForm.Meta	31
Migration	31
Migration	32
Migration	33
Migration	34
node	
Documentation for class node Class node represents a node in the tree structure of code_↔ directory	35
node	
Documentation for class node class node represents a node in the tree structure of code_↔ directory variable stat distinguishes between file and folder	37
Post	
Encapsulation of the Post Model which represents files in the CodeFiles Directory	38
Profile	
This is an encapsulation of the User Profile Model which represents each user in the server uniquely	40
ProfileUpdateForm	41
Snippet	
This class is an encapsulation of the editor	42
SnippetForm	42
Test	43
UserRegisterForm	43
UsersConfig	44
UserUpdateForm	45

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

manage.py	57
blog/__init__.py	47
blog/admin.py	48
blog/apps.py	48
blog/forms.py	48
blog/models.py	50
blog/tests.py	50
blog/urls.py	51
blog/views.py	51
blog/migrations/0001_initial.py	49
blog/migrations/0002_auto_20201206_1527.py	50
blog/migrations/__init__.py	47
CodeFiles/test.cpp	54
CodeFiles/20/abc.cpp	53
CodeFiles/20/ele.cpp	53
CodeFiles/20/hello.cpp	53
CodeFiles/20/hello2.cpp	53
CodeFiles/20/test.cpp	53
CodeFiles/20/time.py	55
CodeFiles/21/hello.cpp	53
CodeFiles/21/ok.py	55
CodeFiles/21/sample.cpp	55
CodeFiles/21/test.cpp	54
CodeFiles/21/test.py	56
CodeFiles/21/subfolder/hello3.cpp	56
CodeFiles/21/subfolder/sub.cpp	56
CodeFiles/21/subfolder/subfile.cpp	56
CodeFiles/21/subfolder/try.cpp	56
CodeFiles/22/coool.cpp	56
CodeFiles/22/Test.java	56
django_project/__init__.py	47
django_project/asgi.py	56
django_project/settings.py	56
django_project/urls.py	51
django_project/wsgi.py	57

users/__init__.py	47
users/admin.py	48
users/apps.py	48
users/forms.py	49
users/models.py	50
users/signals.py	58
users/tests.py	51
users/views.py	52
users/migrations/0001_initial.py	49
users/migrations/0002_snippet.py	58
users/migrations/__init__.py	47

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:

```
1 = [
2     path("", views.home, name='blog-home'),
3     path('about/', views.about, name='blog-about'),
4     path('folderview/<path:folderpath>', views.folderview, name='blog-folderview'),
5     path('create/<path:folderpath>', views.create_file_in_folder, name='blog-create_file_in_folder'),
6 ]
```

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:

```
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 ]
```


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:

```
1 = {  
2     'default': {  
3         'ENGINE': 'django.db.backends.sqlite3',  
4         'NAME': BASE_DIR / 'db.sqlite3',  
5     }  
6 }
```

5.14.2.6 DEBUG

```
bool DEBUG = True
```

5.14.2.7 INSTALLED_APPS

```
list INSTALLED_APPS
```

Initial value:

```
1 = [  
2     'blog.apps.BlogConfig',  
3     'users.apps.UsersConfig',  
4     'crispy_forms',  
5     'django_ace',  
6     'django.contrib.admin',  
7     'django.contrib.auth',  
8     'django.contrib.contenttypes',  
9     'django.contrib.sessions',  
10    'django.contrib.messages',  
11    'django.contrib.staticfiles',  
12 ]
```

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:

```
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 ]
```

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&6q3lio9_50!hrubfl!_=4pxb@'
```

5.14.2.15 STATIC_URL

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

5.14.2.16 TEMPLATES

list 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 ]
```

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:

```
1 = [  
2     path('admin/', admin.site.urls),  
3     path('register/', user_views.register, name='register'),  
4     path('profile/', user_views.profile, name='profile'),  
5     path('login/', auth_views.LoginView.as_view(template_name='users/login.html'), name='login'),  
6     path('logout/', auth_views.LogoutView.as_view(template_name='users/logout.html'), name='logout'),  
7     path('ide/', user_views.simple, name='ide'),  
8     path("", include('blog.urls')),  
9     path('CodeFiles/<path:filepath>', user_views.codeviewer, name='codeviewer'),  
10 ]
```

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 breadth-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 breadth-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, **kwargs)`

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,
    ** kwargs )
```

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 breadth-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

- `__init__`
 - node, [36, 37](#)
 - `__str__`
 - Post, [39](#)
 - Profile, [40](#)
- a
 - test, [20](#)
 - time, [20](#)
- about
 - blog.views, [11](#)
- add_child
 - node, [36, 38](#)
- ALLOWED_HOSTS
 - django_project.settings, [14](#)
- application
 - django_project.asgi, [13](#)
 - django_project.wsgi, [19](#)
- AUTH_PASSWORD_VALIDATORS
 - django_project.settings, [14](#)
- author
 - Post, [39](#)
- b
 - test, [20](#)
- BASE_DIR
 - django_project.settings, [14](#)
- blog, [9](#)
- blog.admin, [9](#)
- blog.apps, [9](#)
- blog.forms, [9](#)
- blog.migrations, [10](#)
- blog.migrations.0001_initial, [10](#)
- blog.migrations.0002_auto_20201206_1527, [10](#)
- blog.models, [10](#)
- blog.tests, [10](#)
- blog.urls, [10](#)
 - urlpatterns, [10](#)
- blog.views, [11](#)
 - about, [11](#)
 - builder, [11](#)
 - create, [12](#)
 - create_file_in_folder, [12](#)
 - detail, [12](#)
 - folderview, [12](#)
 - home, [12](#)
- blog/___init__.py, [47](#)
- blog/admin.py, [48](#)
- blog/apps.py, [48](#)
- blog/forms.py, [48](#)
- blog/migrations/0001_initial.py, [49](#)
- blog/migrations/0002_auto_20201206_1527.py, [50](#)
- blog/migrations/___init__.py, [47](#)
- blog/models.py, [50](#)
- blog/tests.py, [50](#)
- blog/urls.py, [51](#)
- blog/views.py, [51](#)
- BlogConfig, [27](#)
 - name, [27](#)
- builder
 - blog.views, [11](#)
 - users.views, [24](#)
- child
 - node, [36, 38](#)
- CodeFiles/20/abc.cpp, [53](#)
- CodeFiles/20/ele.cpp, [53](#)
- CodeFiles/20/hello.cpp, [53](#)
- CodeFiles/20/hello2.cpp, [53](#)
- CodeFiles/20/test.cpp, [53](#)
- CodeFiles/20/time.py, [55](#)
- CodeFiles/21/hello.cpp, [53](#)
- CodeFiles/21/ok.py, [55](#)
- CodeFiles/21/sample.cpp, [55](#)
- CodeFiles/21/subfolder/hello3.cpp, [56](#)
- CodeFiles/21/subfolder/sub.cpp, [56](#)
- CodeFiles/21/subfolder/subfile.cpp, [56](#)
- CodeFiles/21/subfolder/try.cpp, [56](#)
- CodeFiles/21/test.cpp, [54](#)
- CodeFiles/21/test.py, [56](#)
- CodeFiles/22/coool.cpp, [56](#)
- CodeFiles/22/Test.java, [56](#)
- CodeFiles/test.cpp, [54](#)
- codeviewer
 - users.views, [24](#)
- content
 - Post, [39](#)
- create
 - blog.views, [12](#)
- create_file_in_folder
 - blog.views, [12](#)
- create_profile
 - users.signals, [22](#)
- created_at
 - Snippet, [42](#)
- CreateFileForm, [27](#)
- CreateFileForm.Meta, [30](#)
 - fields, [30](#)
 - model, [30](#)
- CRISPY_TEMPLATE_PACK

- django_project.settings, 15
- DATABASES
 - django_project.settings, 15
- date_posted
 - Post, 40
- DEBUG
 - django_project.settings, 15
- dependencies
 - Migration, 32–35
- detail
 - blog.views, 12
- django_project, 13
- django_project.asgi, 13
 - application, 13
- django_project.settings, 13
 - ALLOWED_HOSTS, 14
 - AUTH_PASSWORD_VALIDATORS, 14
 - BASE_DIR, 14
 - CRISPY_TEMPLATE_PACK, 15
 - DATABASES, 15
 - DEBUG, 15
 - INSTALLED_APPS, 15
 - LANGUAGE_CODE, 15
 - LOGIN_REDIRECT_URL, 15
 - MEDIA_ROOT, 16
 - MEDIA_URL, 16
 - MIDDLEWARE, 16
 - ROOT_URLCONF, 16
 - SECRET_KEY, 16
 - STATIC_URL, 16
 - TEMPLATES, 16
 - TIME_ZONE, 17
 - USE_I18N, 17
 - USE_L10N, 17
 - USE_TZ, 17
 - WSGI_APPLICATION, 17
- django_project.urls, 18
 - document_root, 18
 - MEDIA_URL, 18
 - urlpatterns, 18
- django_project.wsgi, 18
 - application, 19
- django_project/__init__.py, 47
- django_project/asgi.py, 56
- django_project/settings.py, 56
- django_project/urls.py, 51
- django_project/wsgi.py, 57
- document_root
 - django_project.urls, 18
- ele.cpp
 - main, 53
- email
 - UserRegisterForm, 44
 - UserUpdateForm, 45
- exclude
 - SnippetForm.Meta, 28
- f
 - time, 20
- fields
 - CreateFileForm.Meta, 30
 - ProfileUpdateForm.Meta, 31
 - UserRegisterForm.Meta, 29
 - UserUpdateForm.Meta, 30
- folderview
 - blog.views, 12
- get_absolute_url
 - Post, 39
- home
 - blog.views, 12
- image
 - Profile, 41
- initial
 - Migration, 32, 33
- INSTALLED_APPS
 - django_project.settings, 15
- LANGUAGE_CODE
 - django_project.settings, 15
- LOGIN_REDIRECT_URL
 - django_project.settings, 15
- main
 - ele.cpp, 53
 - manage, 19
 - sample.cpp, 55
 - Test, 43
 - test.cpp, 53, 54
- manage, 19
 - main, 19
- manage.py, 57
- MEDIA_ROOT
 - django_project.settings, 16
- MEDIA_URL
 - django_project.settings, 16
 - django_project.urls, 18
- MIDDLEWARE
 - django_project.settings, 16
- Migration, 31–34
 - dependencies, 32–35
 - initial, 32, 33
 - operations, 32–35
- model
 - CreateFileForm.Meta, 30
 - ProfileUpdateForm.Meta, 31
 - SnippetForm.Meta, 28
 - UserRegisterForm.Meta, 29
 - UserUpdateForm.Meta, 31
- name
 - BlogConfig, 27
 - node, 36, 38
 - UsersConfig, 44
- node, 35, 37

- `__init__`, [36, 37](#)
 - `add_child`, [36, 38](#)
 - `child`, [36, 38](#)
 - `name`, [36, 38](#)
 - `path`, [36, 38](#)
 - `stat`, [37, 38](#)
- `ok`, [19](#)
- `operations`
 - `Migration`, [32–35](#)
- `ordering`
 - `Snippet.Meta`, [29](#)
- `path`
 - `node`, [36, 38](#)
- `Post`, [38](#)
 - `__str__`, [39](#)
 - `author`, [39](#)
 - `content`, [39](#)
 - `date_posted`, [40](#)
 - `get_absolute_url`, [39](#)
 - `title`, [40](#)
- `post_save`
 - `users.signals`, [23](#)
- `Profile`, [40](#)
 - `__str__`, [40](#)
 - `image`, [41](#)
 - `save`, [41](#)
 - `user`, [41](#)
- `profile`
 - `users.views`, [24](#)
- `ProfileUpdateForm`, [41](#)
- `ProfileUpdateForm.Meta`, [31](#)
 - `fields`, [31](#)
 - `model`, [31](#)
- `ready`
 - `UsersConfig`, [44](#)
- `register`
 - `users.views`, [24](#)
- `ROOT_URLCONF`
 - `django_project.settings`, [16](#)
- `sample.cpp`
 - `main`, [55](#)
- `save`
 - `Profile`, [41](#)
- `save_profile`
 - `users.signals`, [22](#)
- `SECRET_KEY`
 - `django_project.settings`, [16](#)
- `sender`
 - `users.signals`, [23](#)
- `simple`
 - `users.views`, [24](#)
- `Snippet`, [42](#)
 - `created_at`, [42](#)
 - `text`, [42](#)
- `Snippet.Meta`, [29](#)
 - `ordering`, [29](#)
- `SnippetForm`, [42](#)
- `SnippetForm.Meta`, [28](#)
 - `exclude`, [28](#)
 - `model`, [28](#)
 - `widgets`, [28](#)
- `stat`
 - `node`, [37, 38](#)
- `STATIC_URL`
 - `django_project.settings`, [16](#)
- `TEMPLATES`
 - `django_project.settings`, [16](#)
- `Test`, [43](#)
 - `main`, [43](#)
- `test`, [19](#)
 - `a`, [20](#)
 - `b`, [20](#)
- `test.cpp`
 - `main`, [53, 54](#)
- `text`
 - `Snippet`, [42](#)
- `time`, [20](#)
 - `a`, [20](#)
 - `f`, [20](#)
- `TIME_ZONE`
 - `django_project.settings`, [17](#)
- `title`
 - `Post`, [40](#)
- `top`
 - `users.views`, [25](#)
- `urlpatterns`
 - `blog.urls`, [10](#)
 - `django_project.urls`, [18](#)
- `USE_I18N`
 - `django_project.settings`, [17](#)
- `USE_L10N`
 - `django_project.settings`, [17](#)
- `USE_TZ`
 - `django_project.settings`, [17](#)
- `user`
 - `Profile`, [41](#)
 - `UserRegisterForm`, [43](#)
 - `email`, [44](#)
 - `UserRegisterForm.Meta`, [29](#)
 - `fields`, [29](#)
 - `model`, [29](#)
- `users`, [21](#)
- `users.admin`, [21](#)
- `users.apps`, [21](#)
- `users.forms`, [21](#)
- `users.migrations`, [21](#)
- `users.migrations.0001_initial`, [21](#)
- `users.migrations.0002_snippet`, [22](#)
- `users.models`, [22](#)
- `users.signals`, [22](#)
 - `create_profile`, [22](#)
 - `post_save`, [23](#)

- save_profile, [22](#)
 - sender, [23](#)
- users.tests, [23](#)
- users.views, [23](#)
 - builder, [24](#)
 - codeviewer, [24](#)
 - profile, [24](#)
 - register, [24](#)
 - simple, [24](#)
 - top, [25](#)
- users/__init__.py, [47](#)
- users/admin.py, [48](#)
- users/apps.py, [48](#)
- users/forms.py, [49](#)
- users/migrations/0001_initial.py, [49](#)
- users/migrations/0002_snippet.py, [58](#)
- users/migrations/__init__.py, [47](#)
- users/models.py, [50](#)
- users/signals.py, [58](#)
- users/tests.py, [51](#)
- users/views.py, [52](#)
- UsersConfig, [44](#)
 - name, [44](#)
 - ready, [44](#)
- UserUpdateForm, [45](#)
 - email, [45](#)
- UserUpdateForm.Meta, [30](#)
 - fields, [30](#)
 - model, [31](#)
- widgets
 - SnippetForm.Meta, [28](#)
- WSGI_APPLICATION
 - django_project.settings, [17](#)