Create a Django REST Framework API Project

Step by step setup instructions for your Django REST project!

Last updated on May 4, 2022

Use this guide to start up a Django REST Framework API project from scratch. There's a checklist of setup steps, as well as a video that walks you through the process! For instructions on using Simple JWT as a part of your project, see the last section.

Basic project setup 🔨

- Create a new folder for the project
- `pipenv install` to create Pipfile and create virtual environment
- `pipenv shell` to activate virtual environment
  - NOTE: May need to select virtual environment interpreter in Visual Studio Code
- `pipenv install django djangorestframework mysql-connector-python django-cors-headers`
- `django-admin startproject <project-name>` or `django-admin startproject <projectname> .`

> ❗ NOTE: Running the `startproject` command without a name will add the project files in the current working directory (without nesting into a new directory)

- Run `CREATE DATABASE` query in MySQL Workbench
- Create `local_settings.py` inside of project folder

```
# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = 'g@02@8x89*loc1tmyc_jq(op9mzv1x((kp-^kddyha(qgcp^nl'
```

```
DATABASES = {
    'default': {
        'ENGINE': 'mysql.connector.django',
        'NAME': '<name of database here>',
        'USER': 'root',
        'PASSWORD': '<password here>',
        'HOST': '127.0.0.1',
        'PORT': '3306',
        'OPTIONS': {
            'autocommit': True
        }
    }
}
```

- Import local settings to `settings.py`

```
try:
    from project_name.local_settings import *
except ImportError:
    raise ImportError("local settings not found")
```

  - Add `corsheaders` and `rest_framework` to `INSTALLED_APPS` and `corsheaders.middleware.CorsMiddleware` to `MIDDLEWARE` in `settings.py`
  - Add `CORS_ALLOW_ALL_ORIGINS = True` below `MIDDLEWARE` in `settings.py`

- Migrate
  - Change directory into project (if `django-admin startproject` run with a nested directory for the project)
  - `python manage.py migrate`

- Create a GitHub repository and commit project to GitHub. BE SURE TO ADD A Python `.gitignore` TO ENSURE `local_settings.py` IS IGNORED!

- Create a superuser for the Project: `python manage.py createsuperuser`

- Create apps for project
  - `python manage.py startapp <name of app>`
  - Add to `INSTALLED_APPS` in `settings.py`
  - Add url paths to project `urls.py`

```
"""heroes_villains URL Configuration

The `urlpatterns` list routes URLs to views. For more information please see:
```

```
    https://docs.djangoproject.com/en/3.1/topics/http/urls/
Examples:
Function views
    1. Add an import:  from my_app import views
    2. Add a URL to urlpatterns:  path('', views.home, name='home')
Class-based views
    1. Add an import:  from other_app.views import Home
    2. Add a URL to urlpatterns:  path('', Home.as_view(), name='home')
Including another URLconf
    1. Import the include() function: from django.urls import include, path
    2. Add a URL to urlpatterns:  path('blog/', include('blog.urls'))
"""
from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path('admin/', admin.site.urls),
    path('api/supers/', include('supers.urls')),
    path('api/super_types/', include('super_types.urls'))
]
```

Video Instructions

Add Authentication/Authorization to a Django REST Framework API with Simple-JWT

- Run the command `pipenv install djangorestframework-simplejwt` in a Django project
- Add the following to `settings.py`:

```
REST_FRAMEWORK = {
    'DEFAULT_AUTHENTICATION_CLASSES': (
        'rest_framework_simplejwt.authentication.JWTAuthentication',
    )
}
```

- Add `rest_framework_simplejwt` to `INSTALLED_APPS`
- Add the following paths to the base `url.py` file

```
...
from rest_framework_simplejwt.views import TokenObtainPairView, TokenRefreshView

urlpatterns = [
        ...
    path('api/token/', TokenObtainPairView.as_view(), name='token_obtain_pair'),
    path('api/token/refresh/', TokenRefreshView.as_view(), name='token_refresh'),
        ...
]
```

- Test by running your Django REST API (`python manage.py runserver`) and navigating to `http://localhost:8000/api/token/` and submitting the username and password for the superuser. You should get a 200 OK response with a set of tokens (`refresh` and `access`) as the body of the response.

For additional configuration, like customizing token claims and adjusting default settings, see official documentation: https://django-rest-framework-simplejwt.readthedocs.io/en/latest/index.html

← Create a Python project

Create an HTML/CSS project →