**Create a project**

1. Create project folder
2. Set up python virtual env
3. Install
   1. django
   2. djangorestdramework
   3. django-cors-headers
4. Create src folder under project folder

**Create Django project in the src folder**

1. django-admin startproject <Orella\_APIs >

The created files are:

* The outer <**Orella\_APIs** >**/** root directory is just a container for your project. Its name doesn’t matter to Django; you can rename it to anything you like.
* **manage.py**: A command-line utility that lets you interact with this Django project in various ways. You can read all the details about **manage.py** in [django-admin and manage.py](https://docs.djangoproject.com/en/2.1/ref/django-admin/).
* The inner **mysite/** directory is the actual Python package for your project. Its name is the Python package name you’ll need to use to import anything inside it (e.g. <**Orella\_APIs** >**.urls**).
* <**Orella\_APIs** >**/\_\_init\_\_.py**: An empty file that tells Python that this directory should be considered a Python package. If you’re a Python beginner, read [more about packages](https://docs.python.org/3/tutorial/modules.html#tut-packages) in the official Python docs.
* <**Orella\_APIs** >**/settings.py**: Settings/configuration for this Django project. [Django settings](https://docs.djangoproject.com/en/2.1/topics/settings/) will tell you all about how settings work.
* <**Orella\_APIs** >**/urls.py**: The URL declarations for this Django project; a “table of contents” of your Django-powered site. You can read more about URLs in [URL dispatcher](https://docs.djangoproject.com/en/2.1/topics/http/urls/).
* <**Orella\_APIs** >**/wsgi.py**: An entry-point for WSGI-compatible web servers to serve your project. See [How to deploy with WSGI](https://docs.djangoproject.com/en/2.1/howto/deployment/wsgi/) for more details.

**The development server**

1. cd ../src/<**Orella\_APIs** >
2. python manage.py runserver 3000 (will start the dev server on <http://127.0.0.1:3000/>)

a rocket launch at that ip address confirms everything OK

**Create django app**

1. navigate to **<Orella\_APIs** **>**
2. type py manage.py startapp **<API>**

**<Orella>/settings.py**

1. add ‘rest\_framework’ to end of *INSTALLED\_APPS* = […] section
   1. Optional add 'rest\_framework.authtoken',
2. add ‘<API>’ to end of *INSTALLED\_APPS* = […] section
3. add ‘corsheaders’ to end of *INSTALLED\_APPS* = […] section
4. add ‘corsheaders.middleware.CorsMiddleware’, to TOP of *MIDDLEWARE* = […] section (note caps and comma)
5. add the following new section immediately under the MIDDLEWARE= [] section
   1. CORS\_ORIGIN\_ALLOW\_ALL = True
   2. CORS\_ALLOW\_CREDENTIALS = True
   3. CORS\_ORIGIN\_WHITELIST = # NOTE might not need these as ALLOW\_ALL = TRUE
   4. 'localhost:3000/<endpoint>/',
   5. 'localhost:8080/<endpoint>/',
   6. )

**Database (<API>/models.py)**

1. Create migrations initial migrations
   1. py manage.py makemigrations (this will create modifications to the database)
   2. py manage.py migrate (this will apply all migrations)
2. Set up django models (NOTE not implemented for simple <orella.API>)
3. Set custom user model (NOTE not implemented for simple <orella.API>)
4. Create migrations
   1. py manage.py makemigrations (this will create modifications to the database)
   2. py manage.py migrate (this will apply all migrations)
5. add views to views.py

**Setup django admin (NOT IMPLEMENTED in <orella.API>)**

**Set up an APIView**

1. add APIView to views.py
   1. add the following to views.py
      1. from rest\_framework.views import APIView
      2. from rest\_framework. response import Response
   2. create a class for the API view:
      1. e.g. class WebGrab(APIView):
      2. add get, post, put, patch, and delete methods as necessary
      3. e.g. for get:
         1. def get(self,request,format=None):
         2. return Response()

**Add url for APIView**

1. Create a new urls.py file under the <API> folder and point to this from the base urls.py files (auto created by django and stored at <orella>/<orella>.urls.py. Doc string starts <orella> ULR Configuration)
   1. Create new <API>/urls.py
      1. Add
         1. from django.urls import path
         2. from . import views
         3. urlpatterns = [
         4. path(r'WebGrab/', views.WebGrab.as\_view()), NOTE use of regex VERY IMPORTANT THIS (WebGrab) IS THE NAME THE CLIENT WILL USE TO ACCESS THE API
         5. ]
   2. Point to it from existing <orella>/<orella>.urls.py
      1. add from django.conf.urls import include
      2. add path(r'api/', include('<API>.urls')), NOTE the api sets up a root url for all api calls to be made via. i.e. to reach the WebGrabApi we need to type 127.0.0.1:3000/api/WebGrab. Alternatively use path('', include('<API>.urls')), and then just 127.0.0.1:3000/WebGrab