

## DJANGO TUTORIAL – GEOSQL JOURNEY

First of all, it is highlight recommended to use a virtual environment where you have installed one of the latest version of Python3, and Django.

Then, if you want to start a new Django project, just open a terminal and inside the virtual environment run this command: ***django-admin startproject project***

After that, you should create an application inside the project, you can create as many applications as you want, and the command to run this is the following: ***python3 manage.py startapp application***

It is quite important to add the application created in the project/project/seetings.py file as here is shown:

```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'aplicacion.apps.ApplicationConfig',  
]
```

Then, in the seetings.py file it is necessary to connect a database, it can be SQLite, PostgreSQL... and in that database should be different models of the tables of the dataset, so to register them, we need to do it in the admin.py file:

```
admin.site.register(Model)
```

In this point, we need to execute two commands in order to migrate the data to the Django administration site, this basically is to create the databases script and execute it in our database:

```
python3 manage.py makemigrations
```

```
python3 manage.py migrate
```

And we are close to see the results, but before it is necessary to create a superuser in Django, so the administration site has an admin with user and password set, after this command you can populate your models with the data:

```
python3 manage.py createsuperuser
```

Now, we need to create a folder in which we are going to create our templates (html files), and it will be created at the same level as manage.py file, after it, we need to add this code to the settings.py file:

```
'DIRS': [os.path.join(BASE_DIR, 'templates'),],
```

After configuring the templates properly, an important file to take into consideration is the urls.py file, there are two, one in the application and other in the project:

In the project:

```
path('application/', include('application.urls')),
```

In the application:

```
urlpatterns = [  
    path(url/, views.url, name=url),  
]
```

And then, we are ready to start creating the functions in the views.py file, and once we have created them properly, rendering the returns to the templates, and linking the urls with the functions, we can execute the command to make run the Django server:

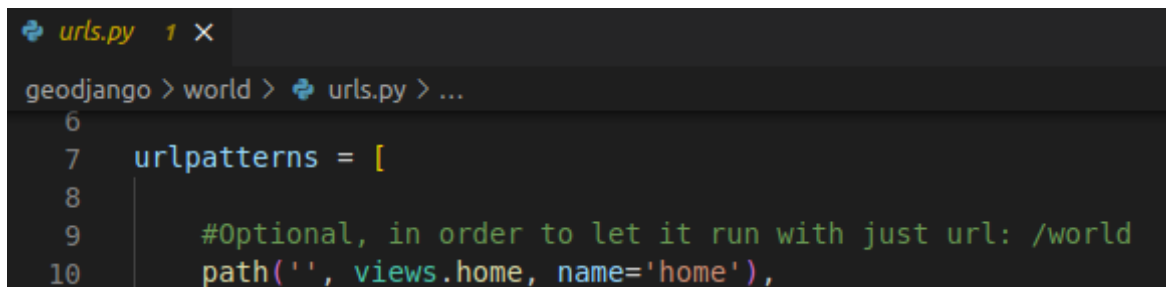
```
python3 manage.py runserver
```

And then we can access to the localhost address: <http://127.0.0.1:8000/> to see our project output in the browser.

## GEOSQL JOURNEY

Let's see now how this works within geoSQL Journey project in a simple “action” putting a basic example:

This is the urls.py file, where we can see that if the url in the localhost after the / is empty, the system needs to go to the home function implemented in the views.py



```
urls.py 1 x  
geodjango > world > urls.py > ...  
6  
7  urlpatterns = [  
8  
9      #Optional, in order to let it run with just url: /world  
10     path('', views.home, name='home'),
```

This is the views.py file showing how the home function is implemented, it is just an empty context and a rendering return to the base.html template:

```
views.py x
geodjango > world > views.py

41
42 # This home function just redirects to base.html template, the main template
43 def home(request):
44     """View function for home page of site."""
45
46     # Create an empty context and render the template with it
47     context = {}
48     return render(request, 'base.html', context=context)
```

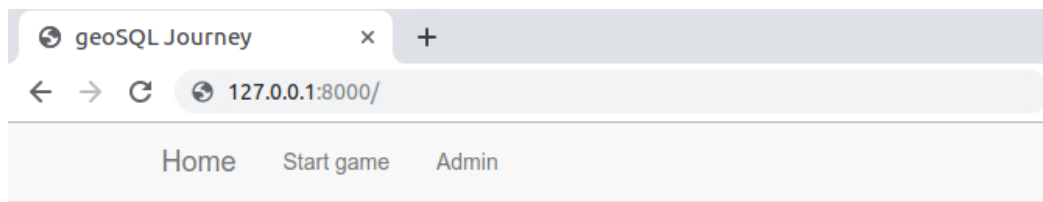
So let's see how is done the base.html template:

```
base.html x
geodjango > templates > base.html > ...

1 {% load static %}
2 <!DOCTYPE html>
3 <html>
4
5 <head>
6     <title>geoSQL Journey</title>
7     <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"
8         integrity="sha384-BVYiiSIfeKldGmJRAKycuHAHRg320mUcw7on3RYdg4Va+PmSTsz/K68vbdEjh4u" crossorigin="anonymous">
9 </head>
10
11 <body>
12     <center>
13         <nav class="navbar navbar-default navbar-static-top">
14             <div class="container">
15                 <ul class="nav navbar-nav">
16                     <li><a class="navbar-brand" href="{% url 'home' %}">Home</a></li>
17                     <li><a class="navbar-link" href="{% url 'new schema' %}">Start game</a></li>
18                     <li><a class="navbar-link" href="{% url 'admin:index' %}">Admin</a></li>
19                 </ul>
20             </div>
21         </nav>
22         <div class="container">
23             <h2>Welcome to geoSQL Journey </h2>
24             <h5>Developed by <em>Alejandro Martín</em></h5>
25         </div>
26     </center>
27     {% block content %}{% endblock %}
28 </body>
29 </html>
```

Analysing the template code, we can see that once the system context is in that point there are several options: home, start game and admin. And depending on what option in the browser we choose, the system will redirect to the corresponding url, and the url to the corresponding view function, and the view to the rendered template.

So once the system context is here, in the browser will be like this:



And this is how Django works! 😊