Pre-coursework: Hello Django

Hello Django

Let us see what a basic Django application looks like. We will create a Hello Django project that will display the Django welcome page.

Step 1. Create the Hello Django Folder

Create a new folder *hello_django*. Inside it create and activate a virtual environment. Like our previous Python applications, we will create a virtual environment to install Python packages.

Step 2. Install Django

Next, we need to install the latest django release.

```
(virtual)$ pip install django==1.11
```

This will make the `django-admin` object available to us in our project. It will allow us to set up our application.

Step 3. Confirm Django is installed

We then need to confirm Django is installed in our application. We need to start a Python shell in our virtual environment by running `python3.6`

```
>>> import django
>>> django.get_version()
'1.11.5'
```

Step 4. Create a Django project

A Django project is a collection of settings for an instance of Django. It is auto-generated and includes database configurations, Django-specific options, and application-specific settings.

There are two ways of creating a project:

Method 1

```
(virtual)$ django-admin startproject heyapp
```

This will create a Django project that has the following structure.

```
heyapp/
    manage.py
    heyapp/
        __init__.py
        settings.py
```

```
        urls.py
        wsgi.py
```

- The outer *heyapp* folder is a container for our project.
- The *manage.py* is a command line utility that allows us to interact with our
- Django project in various ways.
- The inner *heyapp* folder is a package for our project.
- The *__init__.py* file is an empty file that defines the folder as a package
- The *settings.py* is a file that contains all the settings for our web application.
- *urls.py* is a file where we will be creating our URL declarations.
- *wsgi.py* This will be an entry point to our compatible web servers.

Method 2

```
(virtual) $ django-admin startproject heyapp .
```

This method creates a Django project with a similar folder structure as the first method, the only difference is that it does not have the outer `heyapp` folder. The folder structure looks like this;

```
manage.py
heyapp/
     __init__.py
     settings.py
     urls.py
     wsgi.py
```

This method is very useful when it comes to deploying our application and also makes the app structure more understandable. Therefore, from now on we are going to be using the second method whenever we are creating a Django project.

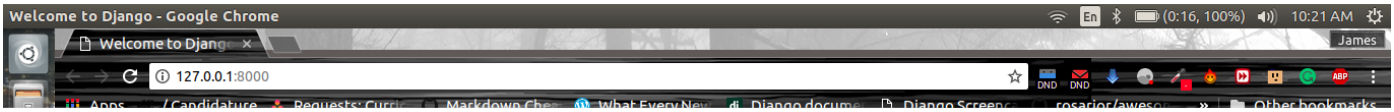
Step 5. Start Django Server

Last but not least we need to start the Django server. Django comes loaded with a fully functional development server that contains everything from debug settings and also automatic reload when we change something in our code.

```
(virtual)$ python3.6 manage.py runserver
Performing system checks...

System check identified no issues (0 silenced).
October 04, 2017 - 01:10:12
Django version 1.11.5, using settings 'heyapp.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

Django checks if there are any errors in your program then start a development server at localhost port `8000`. If everything works you should see the welcome page from Django.


Welcome to Django - Google Chrome
(0:16, 100%)
10:21 AM
Welcome to Djang
James
127.0.0.1:8000
Apps
Other bookmarks