



# Introduction to Database Systems: CS312

## Django: The “Hello World” App!

Oliver Bonham-Carter

14 Oct 2020

# Online Tool: "Python Anywhere"

Creating an  
App

mysite/urls

hello/urls

hello/views

Create Super  
User



- If you were to create an online database site, there are hosting sites to help
- *Host, run, and code Python in the cloud!*
- <https://www.pythonanywhere.com/>
- An account to code (develop Django) from a browser window
- Free accounts (Although, they choose the name of your domain... )

# Today's Task

Creating an  
App

`mysite/urls`

`hello/urls`

`hello/views`

Create Super  
User



- Yesterday: Introduction to what Django does, and for whom
- Today: We create a simple webserver to run a “Hello World” page
- Tomorrow: We add a simple Sqlite3 database to our site to contain website data

# Steps to set up a virtual environment

## MacOS and Linux Commands

Creating an App

`mysite/urls`

`hello/urls`

`hello/views`

Create Super User

### Some commands may require `sudo` for *superuser*

- Install the virtual environment software. Done once. (See next slide.)
- `pip install virtualenv`
- or, `pip3 install virtualenv`
- Setup: Create an environment `myenv` for use with `python3`
- `virtualenv myenv --p python3`
- Activate the environment
- `source myenv/bin/activate`
- Install the Django software packages. Done once for each env.
- `pip install django`
- or, `pip3 install django`

# More on Python virtual environments

Another way to set up a virtual environment (called, "myenv")

Creating an  
App

mysite/urls

hello/urls

hello/views

Create Super  
User

Some versions of Python already have *virtualizing* software already installed

- `python -m venv myenv`
- or, `python3 -m venv myenv`

# Steps to set up a virtual environment

## Windows Commands

Creating an  
App

[mysite/urls](#)

[hello/urls](#)

[hello/views](#)

Create Super  
User

- Install the virtual environment software. Done once. (See prev. slide.)

- `pip install virtualenv`

- Setup: Create an environment `myenv` for use with `python3`

- `virtualenv myenv --p python3`

- Activate the environment

- `cd myenv/Scripts/`

- Execute: `activate`

- Install the Django software packages. Done once for each env.

- `pip install django`

# Setting-up Django

Your terminal should now say, (myenv)

Creating an  
App

mysite/urls

hello/urls

hello/views

Create Super  
User

## Install Django in your virtualenv

```
python -m django --version # check version  
#or, python3 -m django --version # check version
```

## Create your first Django project!

```
django-admin startproject mysite
```

## Use manage.py to run the webserver to see your project!

```
cd mysite/  
# we are now in: djangoWorking/myenv/mysite  
python manage.py runserver  
# or, python3 manage.py runserver
```

## Use your browser to check your work

```
http://127.0.0.1:8000/  
Control-c to exit
```

# Some of New Files Created

Creating an  
App

mysite/urls

hello/urls

hello/views

Create Super  
User

```
find . -not -path '*/\.*'
```

```
-----  
./mysite  
./mysite/__init__.py  
./mysite/__pycache__  
./mysite/__pycache__/settings.cpython-36.pyc  
./mysite/__pycache__/wsgi.cpython-36.pyc  
./mysite/__pycache__/__init__.cpython-36.pyc  
./mysite/__pycache__/urls.cpython-36.pyc  
./mysite/settings.py  
./mysite/urls.py  
./mysite/wsgi.py  
./db.sqlite3  
./manage.py
```



# Add Some Functionality to *mysite*

Creating an  
App

`mysite/urls`

`hello/urls`

`hello/views`

Create Super  
User



- We will now add an App to make the web site do something useful
- We will add a database component later
- Our app will print, "Hello World"
- Isn't that useful?!
- When you add apps, the procedure is often very similar for every project

# Create The Hello App

Creating an  
App

[mysite/urls](#)

[hello/urls](#)

[hello/views](#)

Create Super  
User



- Change into the *mysite* directory to locate the file `manage.py`, if you are not already there.

```
python manage.py startapp hello
```

# Some of New Files Created

Creating an  
App

mysite/urls

hello/urls

hello/views

Create Super  
User

```
find . -not -path '*/\.*'
```

```
-----  
./hello  
./hello/migrations  
./hello/migrations/__init__.py  
./hello/models.py  
./hello/__init__.py  
./hello/__pycache__  
./hello/apps.py  
./hello/admin.py  
./hello/tests.py  
./hello/urls.py  
./hello/views.py
```

## Notable Files

- **apps.py**: The main file for the *hello* App
- **models.py**: A blueprint for how data will be used in the site
- **tests.py**: For adding tests for bug checking the *hello* part of the project
- **views.py**: A request-handler for connecting the URL to the displayed website
- **mysite/mysite/urls.py**: Requests for apps are all directed using this file.
- **mysite/hello/urls.py**: Requests for the *hello* apps are all directed using this file.



## `mysite/mysite/urls.py` and `mysite/hello/urls.py`

- When the user enters a URL address, the website needs to know what pages to display.
- The `hello/urls.py` file and the `mysite/mysite/urls.py` files are used to hold this URL-to-webpage connection information.
- We have to create the `mysite/hello/urls.py` file for to connect the `hello` app's urls to those of the entire website.



## mysite/mysite/urls.py

```
from django.conf.urls import url, include
from django.contrib import admin

urlpatterns = [
    url(r'', admin.site.urls),
    url(r'^admin/', admin.site.urls),
    url(r'^hello/', include('hello.urls'))
]
```

- Be sure to use the correct quotation marks and add include to the top import statement.



## mysite/hello/urls.py

This file needs to be created!

```
from django.conf.urls import include, url
## import local views.py file
from . import views

urlpatterns = [
    url(r'^$', views.index, name = 'index'),
]
```

- Be sure to use the correct quotation marks and add include to the top import statement.



```
from django.shortcuts import render
from django.http import HttpResponse
```

```
def index(request):
    my_str = "<h1> The Hello App's homepage </h1>"
    return HttpResponse(my_str)
```

- The file, *hello/views.py* pushes a an html page to the browser
- The string (the homepage content) is contained by the variable *my\_str*.



# A Working Website?

Creating an  
App

mysite/urls

hello/urls

hello/views

Create Super  
User

- Restart the server: `python manage.py runserver`
- Enter the local URL in your browser:  
`http://127.0.0.1:8000`

## Try these URLs

- `http://127.0.0.1:8000/hello`
- `http://127.0.0.1:8000/admin`



# Create a Super User to View Database

- Need to make a user for the site.

```
python3 manage.py migrate  
python3 manage.py createsuperuser
```

- Migrate is to connect a database to the project to hold user (admin) data
- Username (leave blank to use 'user'): admin
- Email address: studentID@allegheny.edu
- Password: *"pass1234"*
- Password (again): *"pass1234"*
- Superuser created successfully.
- Now, look around the admin page,  
<http://127.0.0.1:8000/admin>