Lecture 3

John C. Pyun

Spring 2016

1 Before we begin...

If you want to know more about all the commands you can use from your manage.py file, use the following command:

```
#in the shell
$python manage.py
```

This should list all the cool commands you can use from python manage.py. I wouldn't recommend just randomly trying out these commands so do not do random commands unless you know what you are doing... I guess...

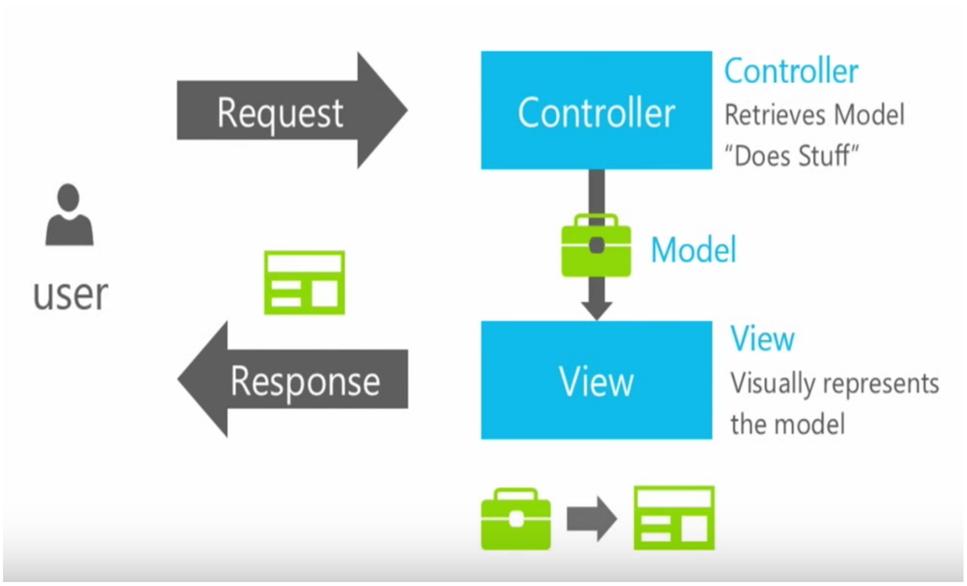
2 Let's migrate our database!

We will now be setting up a bit more intense stuff for our django stuff. So, first things first. Let us migrate our models.

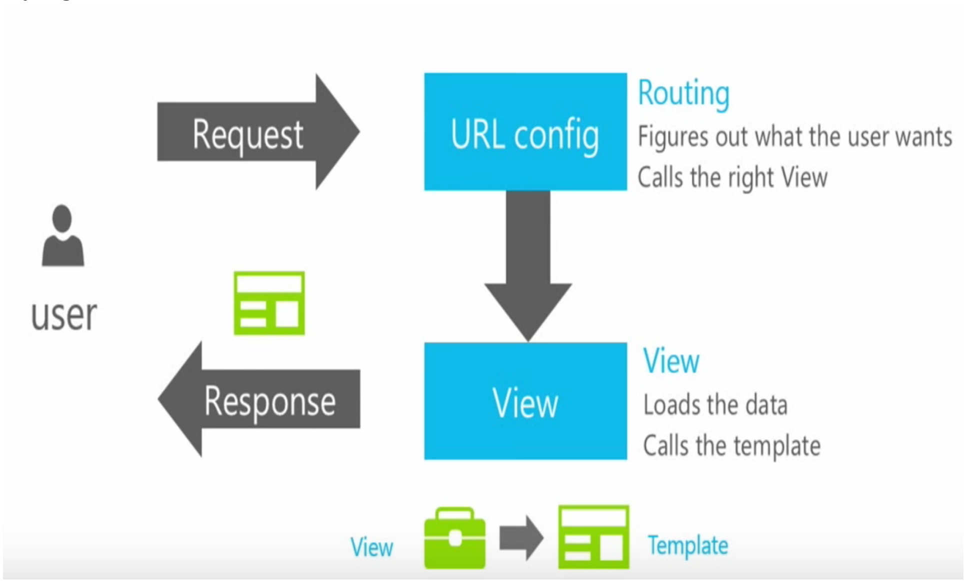
Remember how we talked about MVC format and how model works? If you don't remember, please refer to the following Wikipedia post:

https://en.wikipedia.org/wiki/Model–view–controller

some pictures to refer to:
Classical MVC:



Django MVC:



Now, let's migrate our model to our database. There are two things you must do to migrate your model. They are: "makemigrations" and "migrate" commands. I will explain the difference:

makemigrations is responsible for creating new migrations based on the changes you have made to your models.

migrate is responsible for applying migrations, as well as unapplying and listing their status.

Think of it this way. You first have to use makemigrations command to overwrite all the previous migrations. Then you use migrate command as like ”are you sure you want to migrate?” that finalizes the migration.
Whenever you migrate something you MUST do these two commands accordingly.

```
#in the shell
$python manage.py makemigrations
$python manage.py migrate
```

Now bunch of mumbo jumbo should pop up. If mombo jumbo pops up without any red texts, that means you have successfully migrated all your dank codes.

3 creating superuser

Let’s create a superuser. Superuser is basically the admin account for your django app. creating a super user, it is extremely simple. Just do the following in the terminal and follow the instruction:

```
# in the shell
$python manage.py createsuperuser
# now follow the steps and type enter whenever you finish.
```

Now you have successfully created a superuser. Let’s check it out! start the server by doing the whole python manage.py runserver and then go to the following url:
http://127.0.0.1:8000/admin
(notice how we added ”/admin” at the end. (this can also be seen in urls.py
Now log in, and you should be able to access your admin page!

```
Now, LET’S BUILD A WEBAPP WUBBA LUBBA DUB DUB
```

4 making our first admin site working

First thing’s first. Remember how in our previous chapter, we started a new app called ”homepage” by using the following command?:

```
python manage.py startapp homepage
```

well django now automatically made our ”homepage” app template for us, but it just doesn’t know if we want to use it or not. We must manually tell it that we intend to use our ”homepage” app more intensively.

To do so, open settings.py and go to INSTALLEDAPPS. Then, add ’homepage’, to the INSTALLEDAPPS. After, it should look like the following:

```
#in settings.py
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'homepage',
]
```

Now, let us create our first models. Go to models.py that is situated inside our ”homepage” app. your models.py should look like the following:

```
#models.py
from __future__ import unicode_literals

from django.db import models

# Create your models here.

class blog(models.Model):
    title=models.CharField(max_length=100)
    context=models.TextField()
    after= models.DateTimeField()
    initial= models.DateTimeField()

    def __unicode__(self): ### python 2.7
        return self.title  ### python 2.7
########### for python 3 and above, instead of def __unicode__, do the
     following instead:
    def __str__(self):
        return self.title
```

Notice how we used CharField for title and TextField for context. This is because we want to make our title to be character counter sensitive. (we don’t want a freaking long ass title).

Now, let us go to admins.py and edit the code there. open admin.py and edit to the following:

```
#admin.py
from django.contrib import admin

from .models import blog

admin.site.register(blog)
```

Now, we have set up everything that we need to get our django-backend side working. Let us now migrate django again. Remember howe we first have to do "makemigrations" and then do "migrate"? let's do that again.

```
#in the shell
$ python manage.py makemigrations
$ python manage.py migrate
```

Now let's run our server doing that python manage.py runserver.

Now, go check out your admin site and BAM you have a new section caled "Blog"!

Inside blog, you can add a new post and delete a post and etc.

If you are stuck, please write a piazza post and write an e-mail me at jcp@andrew.cmu.edu