

An Introduction to Django

Patrick Turley & 'Wole Solana

April 13, 2015

ThoughtWorks®

What is Django?

What is Django?

- **Django** is an open source web application (web app) framework written in Python, and follows the **Model-View-Controller** (MVC) design pattern.



What is Django?

- **Django** is an open source web application (web app) framework written in Python, and follows the **Model-View-Controller** (MVC) design pattern.
- A **web framework** can be loosely defined as a set of components that assist in building web apps faster and easier.



What is Django?

- **Django** is an open source web application (web app) framework written in Python, and follows the **Model-View-Controller** (MVC) design pattern.
- A **web framework** can be loosely defined as a set of components that assist in building web apps faster and easier.
- Other frameworks include Rails (Ruby), Drupal (PHP) and Spring (Java)



What is Django?

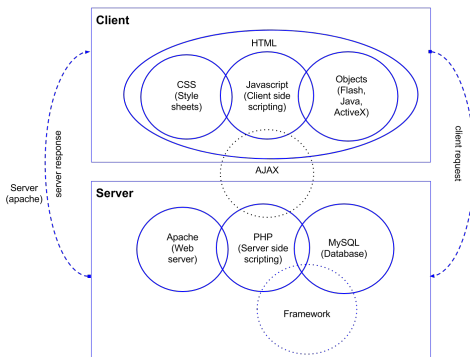
- **Django** is an open source web application (web app) framework written in Python, and follows the **Model-View-Controller** (MVC) design pattern.
- A **web framework** can be loosely defined as a set of components that assist in building web apps faster and easier.
- Other frameworks include Rails (Ruby), Drupal (PHP) and Spring (Java)
- Popular sites built with Django include NASA, Instagram, The Guardian.



Why use a web framework?

Why use a web framework?

Client (Browser - Chrome, Opera, Firefox, IE, etc.)

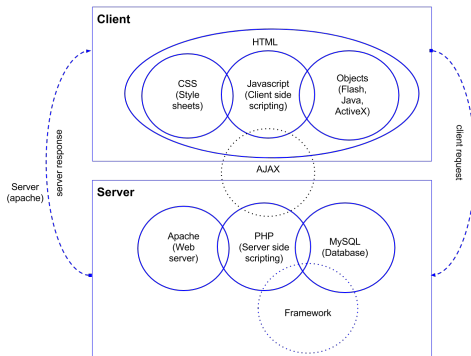


- It enables us to build web apps faster by focusing on the unique functionality of our web app rather than the infrastructure

Figure : Schematic diagram of a web app

Why use a web framework?

Client (Browser - Chrome, Opera, Firefox, IE, etc.)



- It enables us to build web apps faster by focusing on the unique functionality of our web app rather than the infrastructure
- Most provide us with libraries and templates which we can reuse, thus enhancing our productivity.

Figure : Schematic diagram of a web app

Model-View-Controller

Model-View-Controller

- A design pattern (which extends OOP) that helps us better organise our web app by **separating concerns**

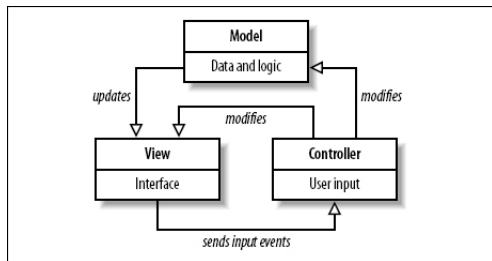


Figure : The MVC pattern

Model-View-Controller

- A design pattern (which extends OOP) that helps us better organise our web app by **separating concerns**
- The **model** contains **data and logic**, the **view** forms the **user interface** (what the user sees and the **controller** manages the **interaction** between the model and view.

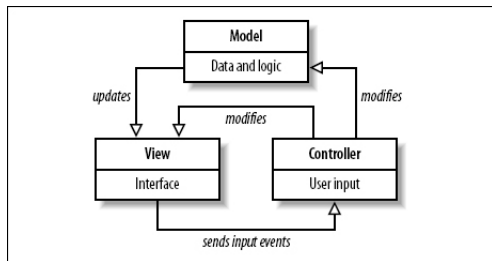


Figure : The MVC pattern

Installation and set up

■ Setup your virtual environment

```
1 virtualenv django_env
```

Installation and set up

■ Setup your virtual environment

```
1 virtualenv django_env
```

■ Activate your virtual environment

```
1 source django_env/bin/activate
```

Installation and set up

■ Setup your virtual environment

```
1 virtualenv django_env
```

■ Activate your virtual environment

```
1 source django_env/bin/activate
```

■ Install django

```
1 pip install django
```

Installation and set up

■ Setup your virtual environment

```
1 virtualenv django_env
```

■ Activate your virtual environment

```
1 source django_env/bin/activate
```

■ Install django

```
1 pip install django
```

■ Start your project

```
1 django-admin startproject firstsite
```


Let's take a look at our app

- Setup the database. Go into the `firstsite` folder and run

```
1 python manage.py migrate
```

Let's take a look at our app

- Setup the database. Go into the `firstsite` folder and run

```
1 python manage.py migrate
```

- Run the server

```
1 python manage.py runserver &
```

Let's take a look at our app

- Setup the database. Go into the `firstsite` folder and run

```
1 python manage.py migrate
```

- Run the server

```
1 python manage.py runserver &
```

- In your browser, navigate to `http://127.0.0.1:8000/`, and *voila!*

Let's add some content

- Still in the `firstsite` folder, run

```
1 python manage.py startapp my_library
```

Let's add some content

- Still in the firstsite folder, run

```
1 python manage.py startapp my_library
```

- Open the settings.py file and in the file, add 'my_library' to the INSTALLED_APPS list so you have something like this:

```
INSTALLED_APPS = (  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'my_library',  
)
```

Creating the book model

- Open `my_library/models.py`

Creating the book model

- Open `my_library/models.py`
- Edit the file (we'll do this together) with the model attributes

Creating the book model

- Open `my_library/models.py`
- Edit the file (we'll do this together) with the model attributes
- Update the database

```
python manage.py makemigrations my_library  
2 python manage.py migrate my_library
```


Creating the book model

- Open `my_library/models.py`
- Edit the file (we'll do this together) with the model attributes
- Update the database

```
python manage.py makemigrations my_library  
2 python manage.py migrate my_library
```

- Open the `my_library/admin.py` file. Update it with the following content:

```
from django.contrib import admin  
from my_library.models import Book  
admin.site.register(Book)
```

Creating the book model

- Open `my_library/models.py`
- Edit the file (we'll do this together) with the model attributes
- Update the database

```
python manage.py makemigrations my_library  
2 python manage.py migrate my_library
```

- Open the `my_library/admin.py` file. Update it with the following content:

```
from django.contrib import admin  
from my_library.models import Book  
admin.site.register(Book)
```

- In your browser, visit `'http://127.0.0.1:8000/admin'`

Some admin

■ Create a superuser

```
python manage.py createsuperuser  
2 python manage.py migrate my_library
```

Some admin

■ Create a superuser

```
python manage.py createsuperuser  
2 python manage.py migrate my_library
```

- Go back to 'http://127.0.0.1:8000/admin', log in and create some books

So how do we get to see our view?