Python Programing

Lesson 07: Web Programming and Web Dev Framework

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

People matter, results count.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

Lesson Objectives

- What is Python Web Framework.
- Different frameworks in Python
- Django – Architecture
  - Urls
  - Views
  - Models
- Django Sample Implementation
- Django Benefits

Web Framework

- A Web framework is a collection of packages or modules which allow developers to write Web applications.
- It provides support for various activities such as getting requests, interpreting it, producing responses, storing data persistently, and so on.
- They are the de facto way to build web-enabled applications.
- The world of Python web frameworks is full of choices.

Popular Frameworks

Following are few popularly used Python web frameworks.

- Django
- Pyramid
- Flask
- Bottle
- Other Frameworks
  - Web2py
  - Pylons
  - TurboGears and many..

Different frameworks

Django vs Flask vs Pyramid

- Flask, the youngest of the three frameworks, started in mid-2010.
- The Pyramid framework began life in the Pylons project and got the name Pyramid in late 2010, though the first release was in 2005.
- Django had its first release in 2006, shortly after the Pylons (eventually Pyramid) project began.
- Pyramid and Django are extremely mature frameworks, and have accumulated plugins and extensions to meet an incredibly large range of needs.

Different frameworks

Django vs Flask vs Pyramid

- Flask is a "micro framework" primarily aimed at small applications with simpler requirements.
- Pyramid and Django are both aimed at larger applications, but take different approaches to extensibility and flexibility.
- Pyramid targets for flexibility. Developers choosing according to the requirement.
- Django aims to include everything, a web application needs.

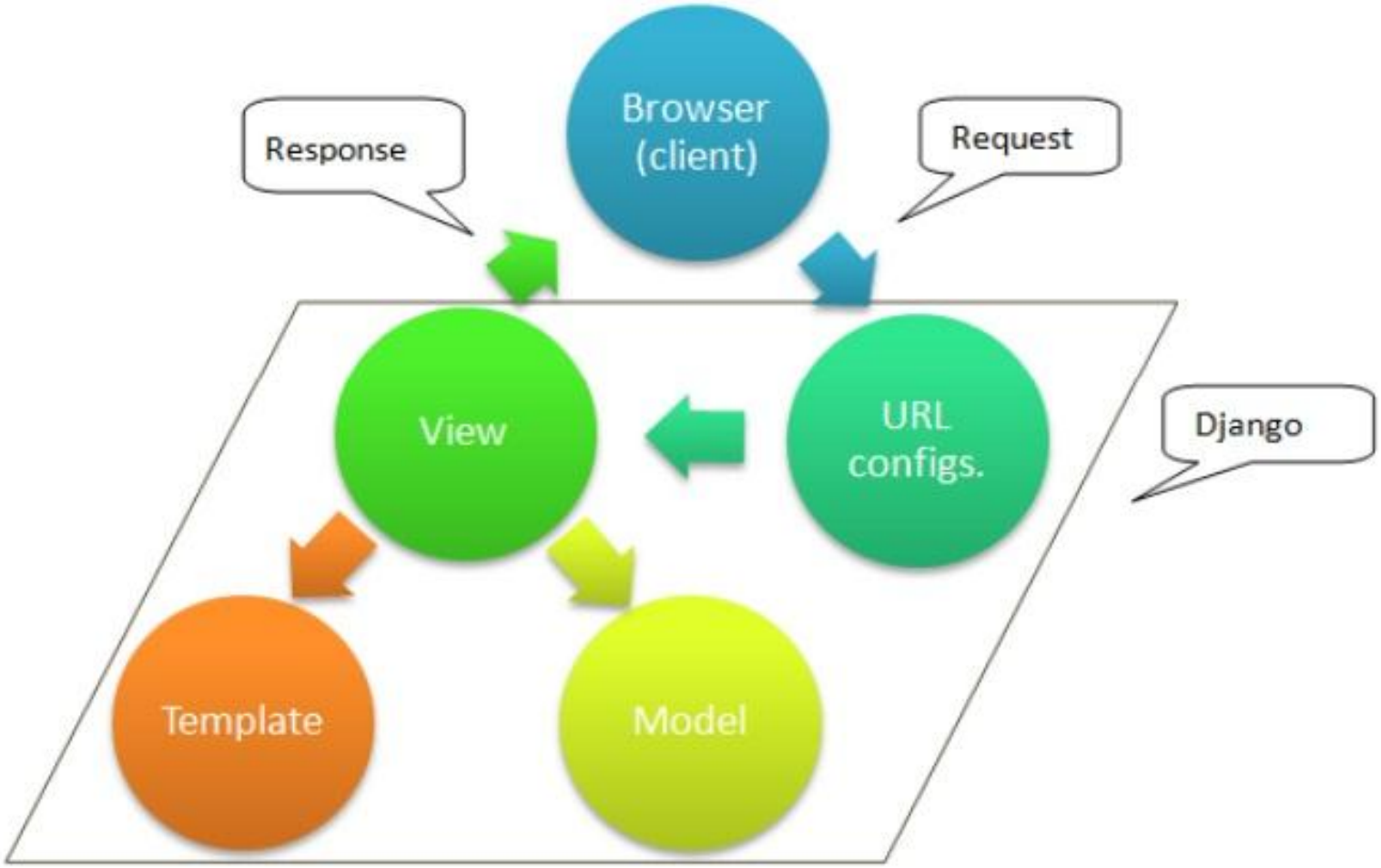
Django

- Named after famous Guitarist “Django Reinhardt”.
- Developed by Adrian Holovaty and Jacob Kaplan-moss at World Online News for efficient development.
- Open sourced in 2005.
- First Version released September 3, 2008

Django as an MVC Design Pattern

- MVC Architecture:
  - Models
    - Describes your data structure/database schema
  - Views
    - Controls what a user sees
  - Templates
    - How a user sees it
  - Controller
    - The Django Framework
    - URL parsing

Django Architecture



What Django generates…

- MySite/
  - __init__.py
  - Manage.py // Script to interact with Django
  - Settings.py // Config
  - URLs.py   // My Site URL mapping
  - MyProject /
    - __init__.py
    - URLs.py // Project specific URL mapping
    - Models.py // Data Models
    - Views.py // Contains the call back functions
    - Admin.py
    - Templates

What Django generates…

- The Django root directory will be named according to the project name you specified in
  - django-admin startproject [projectname].

- This directory is the project's connection with Django.
  - **[projectname]/settings/:** Instead of a plain *settings*-file, the configuration is split into several files in this Python module.

  - **[projectname]/urls.py:** The root URL configuration of the project. The only configured set of urls is the admin-application. Whenever user requests something from website then django will move to urls.py file.

  - **[projectname]/wsgi.py:** Deploying Django makes use of WSGI, the Pythonic way of deploying web applications.

What Django generates…

- manage.py
  - manage.py is automatically created in each Django project. It does the following,
    - It puts your project's package on sys.path.
    - It sets the DJANGO_SETTINGS_MODULE environment variable so that it points to your project's settings.py file.
- mysite/__init__.py
  - An empty file that tells Python that this directory should be considered a Python package.

Django commands

- django-admin startproject [projname]
  - django-admin is Django's command-line utility for administrative tasks.
  - The django-admin script should be on your system path if you installed Django via its setup.py utility
- python manage.py runserver
  - To start the server, change into your project container directory (cd projname) and run the above command
- python manage.py startapp [appname]
  - This command will help to create an app. Once the app is created, its directory structure Is as follows,

Django commands

- appname/
  - __init__.py
  - admin.py
  - apps.py
  - migrations/
    - __init__.py
  - models.py
  - tests.py
  - views.py

Django commands

- python manage.py migrate.
  - When the above command is executed, It will move to website->settings and scroll for APPS.
  - It will then move for individual directory and check what all tables are required to run that app.

- python manage.py shell
  - Above command will initiate a shell which will allow to interact with data base.

Django Benefits

- Object-Relational Mapping (ORM) Support
- Framework Support
- Administration GUI
- Development Environment
- Fast
- Exceedingly scalable.
- Better database handling

Demo

- Demo 1
- Demo 2
- Demo 3
- Demo 4

Summary

- Python Web framework
- Different Frameworks and differences
  - Django
  - Pyramid
  - Flask
- Django Architecture
- Django sample Implementation
- Django Benefits