Django Tutorial
Part II

Greg (mastergreg) Liras, John (nemo) Giannelos

foss.ntua

January 7, 2012

Outline

What is Django?

- ▶ Django is a web-framework written in Python

What is Django?

- ▶ Django is a web-framework written in Python
- ▶ It encourages rapid development and clean pragmatic design

What is Django?

- ▶ Django is a web-framework written in Python
- ▶ It encourages rapid development and clean pragmatic design
- ▶ Follows the MVC (Model-View-Controller) architecture

What is Django?

- ▶ Django is a web-framework written in Python
- ▶ It encourages rapid development and clean pragmatic design
- ▶ Follows the MVC (Model-View-Controller) architecture
- ▶ Focuses on automating common actions (DRY principle)

What is Django?

- ▶ Django is a web-framework written in Python
- ▶ It encourages rapid development and clean pragmatic design
- ▶ Follows the MVC (Model-View-Controller) architecture
- ▶ Focuses on automating common actions (DRY principle)
- ▶ Allows the use of pluggable applications

But first of all...

What is a web-framework?

What is a web-framework?

As quoted from wikipedia:

> *A web application framework is a software framework that is designed to support the development of dynamic websites, web applications and web services. The framework aims to alleviate the overhead associated with common activities performed in Web development.*

▶ At first, WWW was only a bunch of hardcoded HTML pages

- At first, WWW was only a bunch of hardcoded HTML pages
- Rise in need for more and more dynamic content and user interaction

- ▶ At first, WWW was only a bunch of hardcoded HTML pages
- ▶ Rise in need for more and more dynamic content and user interaction
- ▶ New languages designed specifically for web use

- ▶ At first, WWW was only a bunch of hardcoded HTML pages
- ▶ Rise in need for more and more dynamic content and user interaction
- ▶ New languages designed specifically for web use
- ▶ Common tasks showed up in web development

- ▶ At first, WWW was only a bunch of hardcoded HTML pages
- ▶ Rise in need for more and more dynamic content and user interaction
- ▶ New languages designed specifically for web use
- ▶ Common tasks showed up in web development

As a result, complete software-stacks appeared to support the web development process. Think of it is a collection of libraries that do common tasks in web development and are designed to work together.

The MVC architecture

- ▶ MVC stands for Model-View-Controller

The MVC architecture

- ▶ MVC stands for Model-View-Controller
- ▶ Seperates the modeling of the domain, the presentation and the logic/actions of the project, permitting independent development of each.

The MVC architecture

- ▶ MVC stands for Model-View-Controller
- ▶ Seperates the modeling of the domain, the presentation and the logic/actions of the project, permitting independent development of each.
    - ▶ Model: Database and data logic/behaviour

The MVC architecture

- ▶ MVC stands for Model-View-Controller
- ▶ Seperates the modeling of the domain, the presentation and the logic/actions of the project, permitting independent development of each.
    - ▶ Model: Database and data logic/behaviour
    - ▶ View: (What it says), manages what user views

The MVC architecture

- ▶ MVC stands for Model-View-Controller
- ▶ Seperates the modeling of the domain, the presentation and the logic/actions of the project, permitting independent development of each.
    - ▶ Model: Database and data logic/behaviour
    - ▶ View: (What it says), manages what user views
    - ▶ Controller: Interpretation of user's actions

The MVC architecture

- ▶ MVC stands for Model-View-Controller
- ▶ Seperates the modeling of the domain, the presentation and the logic/actions of the project, permitting independent development of each.
  - ▶ Model: Database and data logic/behaviour
  - ▶ View: (What it says), manages what user views
  - ▶ Controller: Interpretation of user's actions
- ▶ Django implements MVC slightly different

Django's MVC

As mentioned before, in the Django framework things are a little bit different.

Welcome to "MTV" (model-template-view).

Django's MVC

▶ Model: Pretty much the same as MVC. Describes our
database using ORM.

Django's MVC

- ▶ Model: Pretty much the same as MVC. Describes our database using ORM.
- ▶ Template: Seperates our website content from its presentation "style".

Django's MVC

- ▶ Model: Pretty much the same as MVC. Describes our database using ORM.

- ▶ Template: Seperates our website content from its presentation "style".

- ▶ View: Describes which data should be presented (decided using appropriate functions) and sends them to our template. It also takes care of the url handling.

Django's MVC

- ▶ Model: Pretty much the same as MVC. Describes our database using ORM.
- ▶ Template: Seperates our website content from its presentation "style".
- ▶ View: Describes which data should be presented (decided using appropriate functions) and sends them to our template. It also takes care of the url handling.
    - ▶ According to Django's FAQ page:

Django's MVC

- ▶ Model: Pretty much the same as MVC. Describes our database using ORM.
- ▶ Template: Seperates our website content from its presentation "style".
- ▶ View: Describes which data should be presented (decided using appropriate functions) and sends them to our template. It also takes care of the url handling.
    - ▶ According to Django's FAQ page:

        ...a view is the Python callback function for a particular URL, because that callback function describes which data is presented.

Django admin-site

Django has a built-in website for various (common) administrative
jobs. It allows you:

- ▶ To manage site users
- ▶ To change form display and handling
- ▶ To make various validations (add,remove,publish e.g
  "articles")