

Django: Framework MVC en python

El Framework para perfeccionistas con fechas limite

Jesús Espino García

GUL UC3M

Jornadas del GUL / 2008

Índice

1

Introducción

- Las cinco preguntas básicas
- Ventajas y Desventajas

2

El Framework

- MVC RoR vs MVC Django
- Las piezas
- Extras

3

Ejemplos

- Ejemplos

Índice

1

Introducción

- Las cinco preguntas básicas
- Ventajas y Desventajas

2

El Framework

- MVC RoR vs MVC Django
- Las piezas
- Extras

3

Ejemplos

- Ejemplos

Las cinco preguntas básicas

- ¿Qué?
 - Framework Web.
 - Escrito en Python.
 - Extraído no diseñado.
- ¿Quién?
 - Adrian Holovaty.
 - Simon Willison.
- ¿Dónde?
 - Lawrence, Kansas.
 - En el periódico Lawrence Journal-World.
- ¿Cuándo?
 - Otoño del 2003.
- ¿Por qué?
 - Necesidad de cumplir unas fechas límite muy ajustadas.

Las cinco preguntas básicas

- ¿Qué?
 - Framework Web.
 - Escrito en Python.
 - Extraído no diseñado.
- ¿Quién?
 - Adrian Holovaty.
 - Simon Willison.
- ¿Dónde?
 - Lawrence, Kansas.
 - En el periódico Lawrence Journal-World.
- ¿Cuándo?
 - Otoño del 2003.
- ¿Por qué?
 - Necesidad de cumplir unas fechas límite muy ajustadas.

Las cinco preguntas básicas

- ¿Qué?
 - Framework Web.
 - Escrito en Python.
 - Extraído no diseñado.
- ¿Quién?
 - Adrian Holovaty.
 - Simon Willison.
- ¿Dónde?
 - Lawrence, Kansas.
 - En el periódico Lawrence Journal-World.
- ¿Cuándo?
 - Otoño del 2003.
- ¿Por qué?
 - Necesidad de cumplir unas fechas límite muy ajustadas.

Las cinco preguntas básicas

- ¿Qué?
 - Framework Web.
 - Escrito en Python.
 - Extraído no diseñado.
- ¿Quién?
 - Adrian Holovaty.
 - Simon Willison.
- ¿Dónde?
 - Lawrence, Kansas.
 - En el periódico Lawrence Journal-World.
- ¿Cuándo?
 - Otoño del 2003.
- ¿Por qué?
 - Necesidad de cumplir unas fechas límite muy ajustadas.

Las cinco preguntas básicas

- ¿Qué?
 - Framework Web.
 - Escrito en Python.
 - Extraído no diseñado.
- ¿Quién?
 - Adrian Holovaty.
 - Simon Willison.
- ¿Dónde?
 - Lawrence, Kansas.
 - En el periódico Lawrence Journal-World.
- ¿Cuándo?
 - Otoño del 2003.
- ¿Por qué?
 - Necesidad de cumplir unas fechas límite muy ajustadas.

Índice

1

Introducción

- Las cinco preguntas básicas
- **Ventajas y Desventajas**

2

El Framework

- MVC RoR vs MVC Django
- Las piezas
- Extras

3

Ejemplos

- Ejemplos

Ventajas

- Es Python.
- Es rápido de desarrollar.
- Esta pensado para la eficiencia.
- Es modular.
- Tiene muy bajo acoplamiento.
- Genera automáticamente un panel de administración.
- Sus bibliotecas hacen gran parte del trabajo.
- Soporta varias bases de datos (MySQL, SQLite, Postgres, MS-SQL)
- Es MVC.

Desventajas

- Es Python.
- No es tan simple de implantar.
- Es mas lento que un framework en un lenguaje compilado.
- No incluye AJAX de serie (todavía).

Índice

1

Introducción

- Las cinco preguntas básicas
- Ventajas y Desventajas

2

El Framework

- MVC RoR vs MVC Django
- Las piezas
- Extras

3

Ejemplos

- Ejemplos

MVC RoR

- Modelo: Abstracción del acceso a datos.
- Vista: Presentación de los datos al usuario.
- Controlador: Lógica de programa.
- Rutas: Redirige las consultas al controlador apropiado.

MVC Django

- Modelo: Abstracción del acceso a datos.
- Vista: Generación y presentación del documento presentado al usuario.
- Controlador: Redirige las consultas a la vista apropiada.
- Plantilla: Generación del HTML a partir de unos datos.

MVC RoR vs MVC Django

RoR		Django	
Modelo	=>	Modelo	
Vista	=>	Plantilla	
Controlador	=>	Vista	
Rutas	=>	Controlador	

Índice

1

Introducción

- Las cinco preguntas básicas
- Ventajas y Desventajas

2

El Framework

- MVC RoR vs MVC Django
- Las piezas
- Extras

3

Ejemplos

- Ejemplos

El diagrama ilustra el flujo de datos en una aplicación Django MVC. El **Usuario** interactúa con el **Controlador** (Consulta 1) y recibe una **Respuesta** (7). El **Controlador** envía una **Consulta** (2) al **Vista**. El **Vista** interactúa con el **Modelo** (Consulta 3, Respuesta 4) y la **Plantilla** (Consulta 5, Respuesta 6). El **Modelo** también envía una **Respuesta** (4) al **Vista**.

```
graph LR
    Usuario[Usuario] -- 1 Consulta --> Controlador[Controlador]
    Controlador -- 2 Consulta --> Vista[Vista]
    Vista -- 3 Consulta --> Modelo[Modelo]
    Modelo -- 4 Respuesta --> Vista
    Vista -- 5 Consulta --> Plantilla[Plantilla]
    Plantilla -- 6 Respuesta --> Vista
    Vista -- 7 Respuesta --> Usuario
```

Legenda:

- Consulta (Flecha azul)
- Respuesta (Flecha verde)

El controlador

- Asocia consultas (urls) a vistas.
- Usa expresiones regulares para determinar las equivalencias.
- Es capaz de capturar parámetros desde la url.
- El uso de expresiones regulares nos permite un primer nivel de validación.

Vista

- Ejecuta las operaciones necesarias para devolver el html al cliente.
- Es el encargado de la lógica de programa.
- Hace uso del modelo para extraer los datos de la base de datos.
- Hace uso de las plantillas para generar el código HTML.

Modelo

- Define los objetos de nuestra base de datos.
- Permite establecer relaciones entre los objetos.
- Una vez definidos nos permite crearlos en nuestra base de datos.
- Una vez creados nuestros objetos nos permite acceder a ellos cómodamente.

Plantillas

- Nos permiten crear la presentación de los datos.
- Genera html a partir de un código html y una serie de variables y estructuras simples.
- Las plantillas que incluye django soportan herencia, bucles, bifurcaciones, variables y filtros.

Middleware

- Procesa los datos entre varios puntos del sistema.
- Nos permite que todas las consultas y respuestas pasen una serie de filtros.
- Django integra algunos como el manejo de sesiones o la autenticación.

Índice

1

Introducción

- Las cinco preguntas básicas
- Ventajas y Desventajas

2

El Framework

- MVC RoR vs MVC Django
- Las piezas
- Extras

3

Ejemplos

- Ejemplos

Interfaz de administración

- Es una aplicación que viene integrada con django.
- Nos permite administrar nuestros objetos de la base de datos.
- Facilita la inserción de datos durante el proceso de desarrollo.
- Se puede personalizar, tanto el aspecto como el funcionamiento.

Vistas genéricas

- Django incorpora algunas vistas genéricas.
- Nos permiten hacer las operaciones mas habituales (listar, borrar, crear...).
- Agilizan el proceso de desarrollo inicial.
- Sirven de base para crear nuestras propias vistas (a partir de ellas).

Índice

1

Introducción

- Las cinco preguntas básicas
- Ventajas y Desventajas

2

El Framework

- MVC RoR vs MVC Django
- Las piezas
- Extras

3

Ejemplos

- Ejemplos

Ejemplos

- Crear un proyecto.
- Crear una aplicación holamundo.
- Hacer nuestra aplicación holamundo en muchos sabores.
- Crear una aplicación miniblog.
- Crear un middleware de ejemplo.

Resumen

- MVC.
- Muy modular.
- Muy rápido de programar.
- Muy potente.