

Sistemas Web Desconectados

Defossé Nahuel, van Haaster Diego Marcos

18 de abril de 2009

Índice general

| | |
|--|----------|
| 1. Introducción | 1 |
| 2. Aplicación del lado del servidor | 2 |
| 2.1. El lenguaje de programación Python | 2 |
| 2.2. Frameworks web basados en Python | 2 |
| 2.3. Django | 2 |
| 2.4. Aplicación del lado del cliente | 3 |
| 2.5. Modelo de ejecución en el cliente | 3 |
| 2.6. Librerías de Javascript | 3 |
| 2.6.1. Elementos de una librería de javascript | 3 |
| 2.7. Ejemplos de librerías | 4 |
| 2.7.1. jQuery | 4 |
| 2.7.2. Prototype | 4 |
| 2.7.3. YUI (Yahoo User Interface) | 4 |
| 2.8. Javascript 1.7 | 4 |
| 2.8.1. Protopy | 4 |
| 3. Aplicación web desconectada | 5 |
| 3.0.2. Problemas de la traducción del framework a Javascript | 5 |
| 3.0.3. Soporte en Django | 5 |
| A. Referencia Protopy | 6 |
| A.1. Módulos | 6 |
| A.1.1. event | 6 |
| A.2. Plataforma Mozilla | 6 |

Índice de figuras

Índice de cuadros

Agradecimientos

Resumen

Capítulo 1

Introducción

Capítulo 2

Aplicación del lado del servidor

2.1. El lenguaje de programación Python

2.2. Frameworks web basados en Python

Acá tenemos que justificar por que django Quizás explicar un poquito de WSGI, FastCGI, mod_python para apache

2.3. Django

Django es un framework web escrito en Python¹ el cual sigue vagamente el concepto de Modelo Vista Controlador. Ideado inicialmente como un administrador de contenido para varios sitios de noticias, los desarrolladores encontraron que su CMS era lo suficientemente genérico como para cubrir un ámbito más amplio de aplicaciones. Fue liberado² bajo la licencia BSD en Julio del 2005 como Django Web Framework en honor a Django Reinhart. En junio del 2008 fue anunciada la creación de la Django Software Foundation, la cual se hace cargo hasta la fecha del desarrollo y mantenimiento.

Estructura de Django

Django consiste en un conjunto de utilidades de consola que permiten crear y manipular proyectos y aplicaciones. Dentro de estas utilidades encontramos un servidor standalone de desarrollo para probar los proyectos, creación de tablas a partir de los modelos del mapeador objeto-relacional, realizar volcado y carga de datos (*fixtures*), ejecución de pruebas basadas en test de unidad, entre otras.

Un proyecto Django consiste en 3 módulos³ básicos:

¹Lenguaje de programación orientado a objetos multiparadigma

²En el ámbito del software libre, la liberación es la fecha en la cual se pone a disposición de la comunidad del software en cuestión

³Un módulo en Python, es un archivo con extensión .py

settings El módulo *settings* define la configuración transversal a las aplicaciones de usuario. En este módulo no se suelen definir más que constantes. Dentro de estas constantes encontramos la base de datos sobre la cual trabaja el ORM, el(los) directorio(s) de las plantillas, las clases middleware, ubicación de los medios estáticos⁴. En este módulo se definen la lista de aplicaciones instaladas.

urls Este módulo define las asociaciones entre las URI y las funciones (vistas) que las atienden. Para generar código más modular, django permite delegar el tratado de urls a módulos dentro de cada aplicación.

manage Esta es la interfase con el framework.

Modelo Modelo de datos de la aplicación. Consiste en un mapeador Objeto Relacional desarrollado específicamente para Django.

Vista Funciones a ser ejecutadas cuando un cliente accede a la aplicación.

Template La generación de salida (típicamente código HTML)

2.4. Aplicación del lado del cliente

2.5. Modelo de ejecución en el cliente

2.6. Librerías de Javascript

Breve explicación de la existencia de tantas librerías de javascript, necesidad de utilización de estas librerías para consistencia, modularidad y portabilidad del código

2.6.1. Elementos de una librería de javascript

Manejo unificado de DOM

Problemas con las implementaciones de DOM de cada navegador, sacar material de la charla que dio Jhon Reisig (está en Google Techtalks)

⁴ Un medio estático es todo contenido que no se genera dinámicamente, como imágenes, librerías de javascript, contenido para embeber como archivos multimedia u elementos

Enfoque funcinal (encadenamiento)

Tipos de datos

Eventos

Ajax

Widgets

2.7. Ejemplos de librerías

2.7.1. jQuery

2.7.2. Prototype

2.7.3. YUI (Yahoo User Interface)

2.8. Javascript 1.7

Explicación de Javascript 2.0.

2.8.1. Protopy

Javascript 1.7 + modularidad + ejecución en contexto

Capítulo 3

Aplicación web desconectada

Doff, Django implementado sobre protopy.

3.0.2. Problemas de la traducción del framework a Javascript

3.0.3. Soporte en Django

Protopy

Apéndice A

Referencia Protopy

A.1. Módulos

A.1.1. `event`

`event.connect` Conectar

A.2. Plataforma Mozilla

- Porque desarrollaron e implementaron Javascript 1.7
- Porque javascript 1.7 toma semántica (y sintaxis???) de Python
- Porque es código abierto
- Porque es extensible mediante plugins
 - Tiene firebug
 - Gears y firebug = muy compardor para el desarrollador.
-

Protopy persigue acercar la semántica del Javascript 1.7 a la del lenguaje Python. Las funcionalidades principales son las siguientes:

- Ámbito de nombres
- Semántica de objetos, dentro de la cual se hace una adaptación de
 - Iteradores
 - Generadores
- Iteradores
- Generadores
- Tipos básicos de la librería estándar de python, entre los que se encuentran:
bool,

type Type sirve para definir clases.

Bibliografía

[1] Versiones de Javascript, Jhon Reisig