

PRÁCTICA 2

UAM-Proyecto de Sistemas Informáticos-G1311

Pareja 2

Autores:

Marcos las Heras Bru y Alejandro Pérez Sánchez

Django es un framework web que funciona basándose en el lenguaje Python, y mezclando contenido de distintos lenguajes; además de caracterizarse por tener una velocidad de cálculo para operaciones complejas relativamente alta.

La base de una aplicación django(Model-Template-View) la podemos dividir en :

Views: Controla la algoritmia y cálculos que se realizan en una determinada vista; es el módulo que termina de dar forma a los templates. En este módulo se controlan tanto los inputs realizados por el usuario como ciertos outputs. Los métodos que pertenecen a este módulo van ligados a los templates y cada método suele controlar un solo template. Estos métodos reciben siempre un tipo de *request* y los argumentos necesarios para realizar cálculos (tal como podríamos señalar en nuestra app como los slugs- `category_name_slug` ...). Views estructura el template haciendo uso de la función *render_template* con la que se genera dicha vista. A esta función se le pueden pasar los datos necesarios utilizados en el template(variables, diccionarios listas ...) y estas variables serán completamente accesibles desde el template previamente mencionado; una utilidad simple de esta función sería mostrar una lista filtrada por Views en el template asociado. Además , los formularios dentro de la aplicación son *parseados* por este módulo, que, por ejemplo en el caso de añadir una página o categoría la crea asignándole los valores adecuados.

Templates: Archivos de código HTML que dan cuerpo a las vistas y que constituyen el esqueleto de la aplicación, se comunican con el resto de módulos mediante formularios procesados en Views y las funciones en este módulo. Podemos estructurar los Templates de un modo simple, y muy común en el desarrollo de páginas web; utilizamos una plantilla de código que será común para todas las vistas(base.html) y se utiliza la herencia para reciclar código y hacer los módulos adicionales mucho más legibles. Un ejemplo de uso de esto en nuestra aplicación sería por ejemplo los links a Sign In, Sign Up ... Que efectivamente parten de base.html. Además mostrará toda la información en el navegador.

Models: Se trata del módulo que controla la parte de la base de datos; es decir, define el cuerpo y tablas de la base de datos (Category, Pages ...) “qué información se va a mostrar y en qué template se va a mostrar”.

CÓMO SE APLICA/CONECTA/FUNCIONA DJANGO

Partiremos del navegador, y cada vez que nosotros hagamos click en cualquier enlace de este se activará un archivo de enlace 'URL' . Este archivo enviará una petición(*request*) a una vista, y aquí podemos encontrar dos caminos:

- 1) Si la vista necesita información de la DATABASE, se va al modelo, el cuál interactuará con la DATABASE y recogerá toda la información necesaria o requerida por la vista para devolvérsela a esta. Una vez que la vista tiene esta información se la envía a un template para que este organice la información y la muestre en el navegador
- 2) Si la vista no necesita información/datos de la DATABASE, simplemente enviará la información al template para que este nuevamente lo muestre en el navegador.

La construcción de las *urls* dentro del proyecto, es gestionada desde el archivo `urls.py`; que, de manera dinámica genera todas y cada una de las urls clickables en la app(menos, como es lógico las urls externas).

A su vez, si definimos el funcionamiento desde un nivel más bajo, al igual que en otras aplicaciones *django* tiene asignado (desarrolladores) un puerto predefinido , esto se refiere al puerto que se define como base(con ip 127.0.0.1 y puerto 8000) al que se accede nada más ejecutar la aplicación.

En el caso de los formularios(*forms*) la ruta interna de ejecución que se sigue funciona de la siguiente manera; el usuario rellena unos campos(formulario gestionado desde `forms.py`); estos se comprueban y parsean(desde `views.py`); y si todo es correcto se le envía al usuario de vuelta el template correspondiente a la siguiente página.