



Taller de Programación Web

Django: Características





DJANGO

Framework de Desarrollo Web

¿Qué es Django?

Django es un **framework de aplicaciones web gratuito** y de **código abierto** (open source) escrito en Python. Un framework web es un conjunto de componentes que te ayudan a desarrollar sitios web más fácil y rápidamente.

Cuando construyes un sitio web, siempre necesitas un conjunto de componentes similares: una manera de manejar la autenticación de usuarios (registrarse, iniciar sesión, cerrar sesión), un panel de administración para tu sitio web, formularios, una forma de subir archivos, etc.

Por suerte para nosotros, hace tiempo que los desarrolladores se dieron cuenta de que siempre se enfrentaban a los mismos problemas cuando construían sitios web, y por eso se unieron y crearon frameworks (Django es uno de ellos) con componentes listos para usarse.

Los frameworks sirven para que no tengamos que reinventar la rueda cada vez y que podamos avanzar más rápido al construir un nuevo sitio.



¿Por qué necesitas un framework?

Para entender para qué sirve realmente Django, necesitamos fijarnos en cómo funcionan los servidores. Lo primero es que el servidor necesita enterarse de que tú quieres que te sirva una página web.

Imagina un buzón (puerto) en el que alguien está constantemente mirando si hay cartas entrantes (peticiones). Esto es lo que hace un servidor web. El servidor web lee la carta, y envía una respuesta con la página web. Pero para enviar algo, tenemos que tener algún contenido. Y Django nos ayuda a crear ese contenido.

Más Características de Django

Django te ayuda a escribir software que es:





- **Completo:**

Django sigue la filosofía "Baterías incluidas" y provee casi todo lo que los desarrolladores quisieran que tenga "de fábrica". Porque todo lo que necesitas es parte de un único "producto", todo funciona a la perfección, sigue principios de diseño consistentes y tiene una amplia y actualizada documentación.

- **Versátil:**

Django puede ser usado para construir casi cualquier tipo de sitio web — desde sistemas manejadores de contenidos y wikis, hasta redes sociales y sitios de noticias. Puede funcionar con cualquier framework en el lado del cliente, y puede devolver contenido en casi cualquier formato (incluyendo HTML, RSS feeds, JSON, XML, etc). ¡El sitio que estás leyendo actualmente está basado en Django! Internamente, mientras ofrece opciones para casi cualquier funcionalidad que desees (distintos motores de base de datos, motores de plantillas, etc.), también puede ser extendido para usar otros componentes si es necesario.

- **Seguro:**

Django ayuda a los desarrolladores a evitar varios errores comunes de seguridad al proveer un framework que ha sido diseñado para "hacer lo correcto" para proteger el sitio web automáticamente. Por ejemplo, Django, proporciona una manera segura de administrar cuentas de usuario y contraseñas, evitando así errores comunes como colocar informaciones de sesión en cookies donde es vulnerable (en lugar de eso las cookies sólo contienen una clave y los datos se almacenan en la base de datos) o se almacenan directamente las contraseñas en un hash de contraseñas.

Un hash de contraseña es un valor de longitud fija creado al enviar la contraseña a una cryptographic hash function. Django puede validar si la contraseña ingresada es correcta enviándola a través de una función hash y comparando la salida con el valor hash almacenado. Sin embargo debido a la naturaleza "unidireccional" de la función, incluso si un valor hash almacenado se ve comprometido es difícil para un atacante resolver la contraseña original.

Django permite protección contra algunas vulnerabilidades de forma predeterminada, incluida la inyección SQL, scripts entre sitios, falsificación de solicitudes entre sitios y clickjacking (consulte Seguridad de sitios web para obtener más detalles sobre dichos ataques).

- **Escalable:**

Django usa un componente basado en la arquitectura "shared-nothing" (cada parte de la arquitectura es independiente de las otras, y por lo tanto puede ser reemplazado o cambiado si es necesario). Teniendo en cuenta una clara separación





entre las diferentes partes significa que puede escalar para aumentar el tráfico al agregar hardware en cualquier nivel: servidores de caché, servidores de bases de datos o servidores de aplicación. Algunos de los sitios más concurridos han escalado a Django para satisfacer sus demandas (por ejemplo, Instagram y Disqus, por nombrar solo dos).

- **Mantenible:**

El código de Django está escrito usando principios y patrones de diseño para fomentar la creación de código mantenible y reutilizable. En particular, utiliza el principio No te repitas "Don't Repeat Yourself" (DRY) para que no exista una duplicación innecesaria, reduciendo la cantidad de código. Django también promueve la agrupación de la funcionalidad relacionada en "aplicaciones" reutilizables y en un nivel más bajo, agrupa código relacionado en módulos, siguiendo el patrón MVC (Model View Controller).

- **Portable:**

Django está escrito en Python, el cual se ejecuta en muchas plataformas. Lo que significa que no está sujeto a ninguna plataforma en particular, y puede ejecutar sus aplicaciones en muchas distribuciones de Linux, Windows y Mac OS X. Además, Django cuenta con el respaldo de muchos proveedores de alojamiento web, y que a menudo proporcionan una infraestructura específica y documentación para el alojamiento de sitios de Django.

