
IPC2 - PROYECTO 3

201901055 – Angel Geovany Aragón Pérez

Resumen

El presente proyecto que consiste en implementar una API que pueda brindar servicios utilizando el protocolo HTTP, para esto se utilizó el framework Django para construir el sitio web ya que es uno de los framework más completo y versátil para devolver cualquier tipo de archivo como XML la cual utilizaremos para obtener y guardar datos de la aplicación.

La aplicación consiste en leer un archivo XML donde se encuentra los registros de los reportes con ello se escribirá un nuevo archivo XML para ordenar y guardar los datos y así poder realizar operaciones de dichos datos.

Abstract

The present project that consists of implementing an API that can provide services using the HTTP protocol, for this the Django framework was used to build the website since it is one of the most complete and versatile framework to return any type of file such as XML which we will use to obtain and save application data.

The application consists of reading an XML file where the records of the reports are found, with this a new XML file will be written to order and save the data and thus perform operations on said data.

Palabras clave

Protocolo, Implementó, Archivo, Operaciones, Framework

Keywords

Protocol, Implemented, File, Operations, Framework

Introducción

El presente proyecto está dirigido a comprender y aplicar el desarrollo de una aplicación web con interfaz gráfica que dé a una solución que implemente tipos de datos abstractos en base a archivos de entrada en formato XML.

Desarrollo del tema

Para el desarrollo del proyecto se utilizó de un framework web llamado Django que nos ayudó a realizar un sitio web-

¿Qué es Django?

Django es un framework web de alto nivel que permite el desarrollo rápido de sitios web seguros y mantenibles. Desarrollado por programadores experimentados, Django se encarga de gran parte de las complicaciones del desarrollo web, por lo que puedes concentrarte en escribir tu aplicación sin necesidad de reinventar la rueda. Es gratuito y de código abierto, tiene una comunidad próspera y activa, una gran documentación y muchas opciones de soporte gratuito y de pago.

¿Por qué usar Django?

Los motivos principales para usar Django son:

Es muy rápido: Si tenéis una startup, tenéis prisa por terminar vuestro proyecto o, simplemente, queréis reducir costes, con Django podéis construir una aplicación muy buena en poco tiempo.

Viene bien cargado: Cualquier cosa que necesitéis realizar, ya estará implementada, sólo hay que adaptarla a vuestras necesidades. Ya sea porque hay módulos de la comunidad, por cualquier paquete Python que encontréis o las propias aplicaciones que Django trae, que son muy útiles.

Es bastante seguro: Podemos estar tranquilos con Django, ya que implementa por defecto algunas medidas de seguridad, las más clásicas, para que no haya SQL Injection, no haya Cross site request forgery (CSRF) o no haya Clickjacking por JavaScript. Django se encarga de manejar todo esto de una manera realmente sencilla.

Es muy escalable: Podemos pasar desde muy poco a una aplicación enorme perfectamente, una aplicación que sea modular, que funcione rápido y sea estable.

Es increíblemente versátil: Es cierto que en un principio Django comienza siendo un Framework para almacenar noticias por sitios de prensa, blogs y este estilo de webs, pero con el tiempo ha ganado tanta popularidad que se puede usar para el propósito que queráis.

Otras bondades de Django que no se destacan en la web son:

Su ORM, su interfaz para acceso a la base de datos, ya que hacer consultas con ella es una maravilla, es una herramienta muy buena.

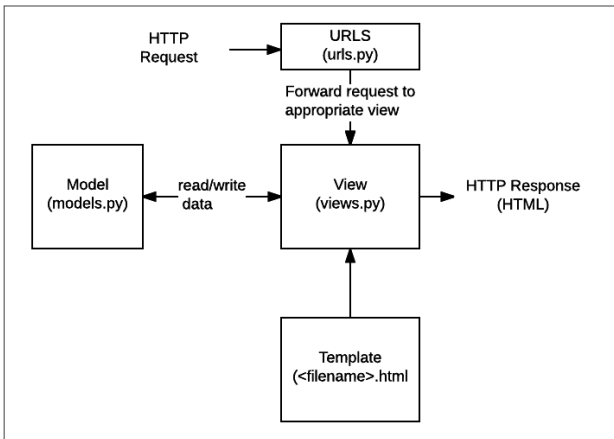
Trae de serie un panel de administración, con el cual podemos dejar a personas sin ningún tipo de conocimiento técnico manejando datos importantes de una forma muy cómoda.

¿Cómo funciona el proyecto?

En un sitio web tradicional basado en datos, una aplicación web espera peticiones HTTP del explorador web (o de otro cliente). Cuando se recibe una petición la aplicación elabora lo que se necesita basándose en la URL y posiblemente en la información incluida en los datos POST o GET. Dependiendo de qué se necesita quizás pueda entonces leer o escribir información desde una base de datos o realizar otras tareas requeridas para satisfacer la petición. La aplicación devolverá a continuación una respuesta al explorador web, con

frecuencia creando dinámicamente una página HTML para que el explorador la presente insertando los datos recuperados en marcadores de posición dentro de una plantilla HTML.

La aplicación esta agrupada de la siguiente manera:



- **Plantillas (Templates):** una plantilla (template) es un fichero de texto que define la estructura o diagrama de otro fichero (tal como una página HTML), con marcadores de posición que se utilizan para representar el contenido real. Una vista puede crear dinámicamente una página usando una plantilla, rellenandola con datos de un modelo. Una plantilla se puede usar para definir la estructura de cualquier tipo de fichero; ¡no tiene porqué ser HTML!
 - **Formularios:** Los formularios HTML se usan para recolectar datos de los usuarios para su procesamiento en el servidor. Django simplifica la creación, validación y procesamiento de los formularios.
-
- **URLs:** Aunque es posible procesar peticiones de cada URL individual vía una función individual, es mucho más sostenible escribir una función de visualización separada para cada recurso. Se usa un mapeador URL para redirigir las peticiones HTTP a la vista apropiada basándose en la URL de la petición.
 - **Vista (View):** Una vista es una función de gestión de peticiones que recibe peticiones HTTP y devuelve respuestas HTTP. Las vistas acceden a los datos que necesitan para satisfacer las peticiones por medio de modelos, y delegan el formateo de la respuesta a las plantillas ("templates")
 - **Modelos (Models):** Los Modelos son objetos de Python que definen la estructura de los datos de una aplicación y proporcionan mecanismos para gestionar (añadir, modificar y borrar) y consultar registros en la base de datos.

Conclusiones

Django permite el desarrollo rápido de sitios web seguros y mantenibles ya que se encarga de gran parte de las complicaciones del desarrollo web, por lo que puedes concentrarte en escribir tu aplicación sin necesidad de reinventar la rueda.