
SOLUCIÓN INTEGRAL IMPLEMENTANDO PROTOCOLO HTTP Y EL CONSUMO DE API

201902888 – Ivan Alessandro Hilario Chacon

Resumen

El presente proyecto tiene como finalidad la implementación de una solución integral que pueda brindar servicios utilizando protocolo HTTP y el consumo de API.

Como simulación de una base de datos se utilizaron archivos de extensión xml. Archivo el cual contenía datos que se analizaban en la parte de servidor.

En el lado del cliente se implementa todo lo relacionado con la interacción del usuario, y la comunicación con el servidor. La utilización de frameworks web para realizar estas operaciones es esencial actualmente y utilizar un framework basado en un lenguaje de programación de alto nivel como lo es Python hace que las funcionalidades puedan ser más accesibles y contar con un nivel más ordenado y sistemático.

Palabras clave

Django, Flask, XML, Interfaz de Programación de Aplicaciones (API), Protocolo HTTP.

Abstract

The purpose of this project is to implement a comprehensive solution that can provide services using HTTP protocol and API consumption.

As simulation of a database, xml extension files were used. File which contained data that was analyzed on the server side.

On the client side, everything related to user interaction and communication with the server is implemented. The use of web frameworks to perform these operations is currently essential and using a framework based on a high-level programming language such as Python makes the functionalities more accessible and have a more orderly and systematic level.

Keywords

Django, Flask, XML, Application Programming Interface (API), HTTP Protocol.

Introducción

La utilización de frameworks web para el desarrollo de aplicaciones es actualmente la manera más idónea para utilizar ya que significa en gran manera la simplificación para el programador de la codificación, arquitectura, etc.

Django es un framework web que utiliza el patrón Modelo Vista Controlador lo cual significa que el software que creemos con dicho framework será más robusto, con un ciclo de vida más adecuado, donde se potencia la facilidad de mantenimiento, reutilización del código y la separación de conceptos.

Por otro lado, Flask es también un framework web, sin embargo, se utiliza mayormente como gestor de API's por su fácil uso de rutas o endpoints.

Django

Django es un framework de desarrollo web de código abierto, escrito en Python, que respeta el patrón de diseño conocido como modelo–vista–controlador (MVC). Fue desarrollado en origen para gestionar varias páginas orientadas a noticias de la World Company de Lawrence, Kansas, y fue liberada al público bajo una licencia BSD en julio de 2005; el framework fue nombrado en alusión al guitarrista de jazz gitano Django Reinhardt.

En junio de 2008 fue anunciado que la recién formada Django Software Foundation se haría cargo de Django en el futuro.

La meta fundamental de Django es facilitar la creación de sitios web complejos. Django pone énfasis en el re-uso, la conectividad y extensibilidad de componentes, el desarrollo rápido y el principio No te repitas (DRY, del inglés Don't Repeat

Yourself). Python es usado en todas las partes del framework, incluso en configuraciones, archivos, y en los modelos de datos.

Los orígenes de Django en la administración de páginas de noticias son evidentes en su diseño, ya que proporciona una serie de características que facilitan el desarrollo rápido de páginas orientadas a contenidos. Por ejemplo, en lugar de requerir que los desarrolladores escriban controladores y vistas para las áreas de administración de la página, Django proporciona una aplicación incorporada para administrar los contenidos, que puede incluirse como parte de cualquier página hecha con Django y que puede administrar varias páginas a partir de una misma instalación; la aplicación administrativa permite la creación, actualización y eliminación de objetos de contenido, llevando un registro de todas las acciones realizadas sobre cada uno, y proporciona una interfaz para administrar los usuarios y los grupos de usuarios (incluyendo una asignación detallada de permisos).

La distribución principal de Django también aglutina aplicaciones que proporcionan un sistema de comentarios, herramientas para syndicar contenido via RSS y/o Atom, "páginas planas" que permiten gestionar páginas de contenido sin necesidad de escribir controladores o vistas para esas páginas, y un sistema de redirección de URLs.

Arquitectura Django

Aunque Django está fuertemente inspirado en la filosofía de desarrollo Modelo Vista Controlador, sus desarrolladores declaran públicamente que no se sienten especialmente atados a observar

estrictamente ningún paradigma particular, y en cambio prefieren hacer "lo que les parece correcto". Como resultado, por ejemplo, lo que se llamaría "controlador" en un "verdadero" framework MVC se llama en Django "vista", y lo que se llamaría "vista" se llama "plantilla".

Gracias al poder de las capas mediator y foundation, Django permite que los desarrolladores se dediquen a construir los objetos Entity y la lógica de presentación y control para ellos.

Modelo Vista Controlador (MVC)

En líneas generales, MVC es una propuesta de arquitectura del software utilizada para separar el código por sus distintas responsabilidades, manteniendo distintas capas que se encargan de hacer una tarea muy concreta, lo que ofrece beneficios diversos.

MVC se usa inicialmente en sistemas donde se requiere el uso de interfaces de usuario, aunque en la práctica el mismo patrón de arquitectura se puede utilizar para distintos tipos de aplicaciones. Surge de la necesidad de crear software más robusto con un ciclo de vida más adecuado, donde se potencie la facilidad de mantenimiento, reutilización del código y la separación de conceptos.

Su fundamento es la separación del código en tres capas diferentes, acotadas por su responsabilidad, en lo que se llaman Modelos, Vistas y Controladores, o lo que es lo mismo, Model, Views & Controllers, si lo prefieres en inglés. En este artículo estudiaremos con detalle estos conceptos, así como las ventajas de ponerlos en marcha cuando desarrollamos.

MVC es un "invento" que ya tiene varias décadas y fue presentado incluso antes de la aparición de la Web. No obstante, en los últimos años ha ganado mucha fuerza y seguidores gracias a la aparición de numerosos frameworks de desarrollo web que utilizan el patrón MVC como modelo para la arquitectura de las aplicaciones web.

¿Por qué MVC?

La rama de la ingeniería del software se preocupa por crear procesos que aseguren calidad en los programas que se realizan y esa calidad atiende a diversos parámetros que son deseables para todo desarrollo, como la estructuración de los programas o reutilización del código, lo que debe influir positivamente en la facilidad de desarrollo y el mantenimiento.

Los ingenieros del software se dedican a estudiar de qué manera se pueden mejorar los procesos de creación de software y una de las soluciones a las que han llegado es la arquitectura basada en capas que separan el código en función de sus responsabilidades o conceptos. Por tanto, cuando estudiamos MVC lo primero que tenemos que saber es que está ahí para ayudarnos a crear aplicaciones con mayor calidad.

Flask

Flask es un framework minimalista escrito en Python que permite crear aplicaciones web rápidamente y con un mínimo número de líneas de código. Está basado en la especificación WSGI de Werkzeug y el motor de templates Jinja2 y tiene una licencia BSD.

Hola Mundo en Flask

```
from flask import Flask
app = Flask(__name__)

@app.route("/")
def hello():
    return "¡Hola Mundo!"

if __name__ == "__main__":
    app.run()
```

¿Qué es XML?

XML es el acrónimo de Extensible Markup Language, es decir, es un lenguaje de marcado que define un conjunto de reglas para la codificación de documentos. ¿Te parece complicado? Entonces, vamos a simplificarlo.

El lenguaje de marcado es un conjunto de códigos que se pueden aplicar en el análisis de datos o la lectura de textos creados por computadoras o personas. El lenguaje XML proporciona una plataforma para definir elementos para crear un formato y generar un lenguaje personalizado.

Un archivo XML se divide en dos partes: prolog y body. La parte prolog consiste en metadatos administrativos, como declaración XML, instrucción de procesamiento opcional, declaración de tipo de documento y comentarios. La parte del body se compone de dos partes: estructural y de contenido (presente en los textos simples).

El diseño XML se centra en la simplicidad, la generalidad y la facilidad de uso y, por lo tanto, se utiliza para varios servicios web. Tanto es así que hay sistemas destinados a ayudar en la definición de lenguajes basados en XML, así como APIs que ayudan en el procesamiento de datos XML – que no deben confundirse con HTML.

```
1 <persona>
2   <nombres>Elsa</nombres>
3   <apellidos>Zambrano</apellidos>
4   <fecha-de-nacimiento>
5     <día>18</día>
6     <mes>6</mes>
7     <año>1996</año>
8   </fecha-de-nacimiento>
9   <ciudad>Pamplona</ciudad>
10</persona>
```

Ejemplo de un archivo xml.

¿Qué es HTML?

HTML (Hypertext Markup Language) es el lenguaje de marcado de documentos para construir páginas web. Por lo tanto, los comandos de formato utilizados en los contenidos para web se refieren a la estructura del mismo y al diseño que se mostrará en el navegador.

Es decir, los navegadores leen el documento con el formato HTML y lo procesan en la pantalla mediante el examen de los elementos HTML insertados en el documento, que se considera un archivo de texto con la información que se debe publicar.

Por eso, podemos generar un archivo HTML utilizando el Bloc de notas de nuestra computadora, por ejemplo.

Las instrucciones incorporadas se conocen como elementos que muestran la estructura y la presentación del documento en el navegador. Estos elementos se componen de las tags que definen el formato de un texto. Las tags suelen estar dos veces: tag inicial y tag final. Por ejemplo, para poner el texto en negrita, las etiquetas se usan al principio y al final.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-
Compatible" content="//E=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>

</body>
</html>
```

Ejemplo de plantilla base de HTML 5

Diferencias entre XML y HTML

El XML es un lenguaje de marcado basado en texto que tiene una estructura de auto-descripción y puede definir efectivamente otro lenguaje de marcado. Por otro lado, el HTML es un lenguaje de marcado predefinido y tiene una capacidad limitada;

El XML proporciona la estructura lógica del documento, mientras que la estructura del HTML está predefinida, utilizando tags heads e body;

Cuando se trata de lenguaje, el HTML es insensible a mayúsculas y minúsculas. En cambio, el XML distingue entre mayúsculas y minúsculas;

El HTML fue diseñado con un énfasis en las características de presentación de los datos. En contraste, el XML es específico de datos;

Interfaz de Programación de Aplicaciones

Una API es una interfaz de programación de aplicaciones (del inglés API: Application Programming Interface). Es un conjunto de rutinas que provee acceso a funciones de un determinado software.

Son publicadas por los constructores de software para permitir acceso a características de bajo nivel o propietarias, detallando solamente la forma en que cada rutina debe ser llevada a cabo y la funcionalidad que brinda, sin otorgar información acerca de cómo se lleva a cabo la tarea. Son utilizadas por los programadores para construir sus aplicaciones sin necesidad de volver a programar funciones ya hechas por otros, reutilizando código que se sabe que está probado y que funciona correctamente.

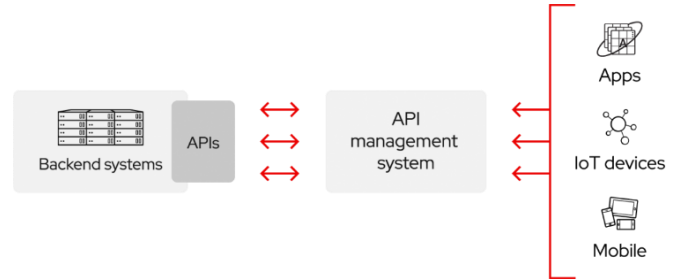
Las API permiten que sus productos y servicios se comuniquen con otros, sin necesidad de saber cómo están implementados. Esto simplifica el desarrollo de las aplicaciones y permite ahorrar tiempo y dinero. Las API le otorgan flexibilidad; simplifican el diseño, la administración y el uso de las aplicaciones, y proporcionan oportunidades de innovación, lo cual es

ideal al momento de diseñar herramientas y productos nuevos (o de gestionar los actuales).

A veces, las API se consideran como contratos, con documentación que representa un acuerdo entre las partes: si una de las partes envía una solicitud remota con cierta estructura en particular, esa misma estructura determinará cómo responderá el software de la otra parte.

Debido a que simplifican la forma en que los desarrolladores integran los elementos de las aplicaciones nuevas en una arquitectura actual, las API permiten la colaboración entre el equipo comercial y el de TI. Las necesidades comerciales suelen cambiar rápidamente en respuesta a los mercados digitales en constante cambio, donde la competencia puede modificar un sector entero con una aplicación nueva. Para seguir siendo competitivos, es importante admitir la implementación y el desarrollo rápidos de servicios innovadores. El desarrollo de aplicaciones nativas de la nube es una forma identificable de aumentar la velocidad de desarrollo y se basa en la conexión de una arquitectura de aplicaciones de microservicios a través de las API.

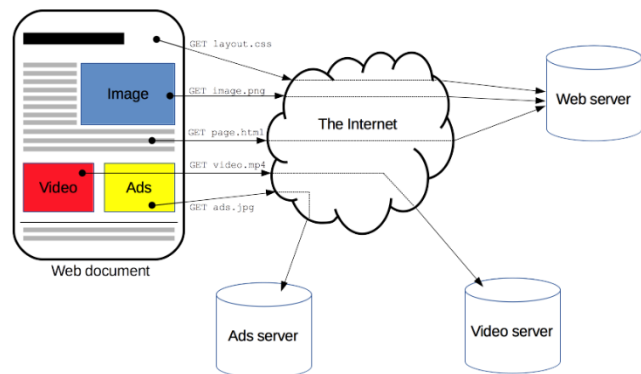
Las API son un medio simplificado para conectar su propia infraestructura a través del desarrollo de aplicaciones nativas de la nube, pero también le permiten compartir sus datos con clientes y otros usuarios externos. Las API públicas representan un valor comercial único porque simplifican y amplían la forma en que se conecta con sus partners y, además, pueden rentabilizar sus datos (un ejemplo conocido es la API de Google Maps).



Ejemplo de funcionalidad de API.

Protocolo HTTP

HTTP, de sus siglas en inglés: "Hypertext Transfer Protocol", es el nombre de un protocolo el cual nos permite realizar una petición de datos y recursos, como pueden ser documentos HTML. Es la base de cualquier intercambio de datos en la Web, y un protocolo de estructura cliente-servidor, esto quiere decir que una petición de datos es iniciada por el elemento que recibirá los datos (el cliente), normalmente un navegador Web. Así, una página web completa resulta de la unión de distintos sub-documentos recibidos, como, por ejemplo: un documento que especifique el estilo de maquetación de la página web (CSS), el texto, las imágenes, vídeos, scripts, etc...



Clientes y servidores se comunican intercambiando mensajes individuales (en contraposición a las comunicaciones que utilizan flujos continuos de datos). Los mensajes que envía el cliente, normalmente un navegador Web, se llaman peticiones, y los mensajes enviados por el servidor se llaman respuestas.

Diseñado a principios de la década de 1990, HTTP es un protocolo ampliable, que ha ido evolucionando con el tiempo. Es lo que se conoce como un protocolo de la capa de aplicación, y se transmite sobre el protocolo TCP, o el protocolo encriptado TLS (en-US), aunque teóricamente podría usarse cualquier otro protocolo fiable. Gracias a que es un protocolo capaz de ampliarse, se usa no solo para transmitir documentos de hipertexto (HTML), si no que además, se usa para transmitir imágenes o vídeos, o enviar datos o contenido a los servidores, como en el caso de los formularios de datos. HTTP puede incluso ser utilizado para transmitir partes de documentos, y actualizar páginas Web en el acto.

Conclusiones

Los frameworks Web facilitan el desarrollo de software, ya que simplifican en gran parte la codificación y trabajos de arquitectura en los proyectos que se realizan utilizando esta tecnología.

El protocolo HTTP es la base de cualquier intercambio de datos en la Web, ya que una petición de datos es iniciada por el elemento que recibirá los datos (el cliente), normalmente un navegador Web hacia un servidor. Por lo que el uso del protocolo es indispensable en cualquier proyecto web.

Referencias bibliográficas

MDN Web Docs (2020). Generalidades del protocolo HTTP. [Generalidades del protocolo HTTP - HTTP | MDN \(mozilla.org\)](https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview)

Wikipedia, (2021). Web API. [Web API - Wikipedia, la enciclopedia libre](https://es.wikipedia.org/wiki/Web_API)

RedHat (2020). Qué es una API. [¿Qué es una API? \(redhat.com\)](https://www.redhat.com/es/topics/api/what-is-an-api)

Wikipedia, (2021). Extensible Markup Language. [Extensible Markup Language - Wikipedia, la enciclopedia libre](https://es.wikipedia.org/wiki/Extensible_Markup_Language)