
PROGRAMACION DE APLICACIÓN WEB PARA LA SUPERINTENDENCIA DE ADMINISTRACION TRIBUTARIA

201807220 – Héctor Josué Ponsoy Ayala

Resumen

El elemento principal en el que se basan estos servicios son las URLs. En líneas generales podemos decir que estos servicios consisten en URLs a las que podemos acceder, por ejemplo, mediante protocolo HTTP, para obtener información o realizar alguna operación. El formato de la información que se intercambie con estas URLs ya se decidió como se iban a llamar. Este tipo de servicios acercan los Servicios Web al tipo de arquitectura de la web, con ella se pudo tener o tomar en cuenta que existen patrones de arquitectura para el área del Frontend, para este caso es el MVC. Aplicando siempre los estándares del JSON y XML para el manejo y la persistencia de los datos.

Palabras clave

Protocolo HTTP
Servidor
Peticiones
Cliente
XML

Abstract

The main element on which these services are based are the URLs. In general terms, we can say that these services consist of URLs that we can access, for example, through the HTTP protocol, to obtain information or perform an operation. The format of the information that is exchanged with these URLs has already been decided as they were going to be called. This type of service brings Web Services closer to the type of architecture of the web, with it it could be taken into account that there are architecture patterns for the Frontend area, for this case it is the MVC. Always applying the json and xml standards for the handling and persistence of the data.

Keywords

*HTTP Protocole
Server
Requests
Client
XML*

Introducción

El desarrollar servicios web Rest es algo que se parece al desarrollo de apps web. Aunque una de las diferencias mas notables es acerca de nuestra manera de pensar sobre las acciones a realizar sobre nuestras recopilaciones de datos. De manera mas especifica, el desarrollo del día de hoy esta centrado en el concepto de nombres, el desarrollo tradicional está centrado en el concepto de verbos. Con la primera forma, estamos desarrollando un servicio web RESTful. Con el segundo se trata de un servicio similar a una llamada a un procedimiento remoto (RPC). Y el servicio RESTful modifica el estado de los datos a través de la representación de los recursos, haciendo uso del modelo vista controlador, haciendo uso del framework Django el cual es de desarrollo web y de código abierto, y el mismo está escrito en Python, por lo que se respeta dicho modelo de vista.

Desarrollo del tema

En la actualidad podemos usar APIs para hacer multitud de sitios web. Por ejemplo, podríamos usar el API REST de Twitter para crear un servicio basada en esa red o el API de YouTube para crear un sitio web que muestra vídeos de distintas maneras. Pero para aplicar estos conceptos, inicialmente se tomó en cuenta otras herramientas para la persistencia de los datos. Debido a que el XML proporcionado había que analizarlo detenidamente para obtener la información correctamente y necesaria. XML es el nuevo estándar universal para intercambio electrónico de datos. Este estándar es un metalenguaje que puede ser utilizado para describir la estructura lógica y el contenido de una gran variedad documentos, y puede ser adaptado para satisfacer una gran cantidad de aplicaciones

Estos atributos de ser universal y extensible abre un rango ilimitado de usos para el XML, desde procesadores de texto, páginas web, el comercio electrónico, hasta las más complejas soluciones de almacenamiento en bases de datos.

Para la solución de dicho proyecto se implemento el uso de del microframework Flask, es cual es nativo de Python, por lo que para instalarlo se hace por medio de pip, se hicieron uso de libreritas tales como cors, requests, response, etc.

Así mismo se crearon las distintas rutas en la api de flask, y se hicieron pruebas con Postman antes de hacer la conexión con el frontend, para así saber que la funcionalidad de la aplicación en flask estaba funcionando acorde a lo esperado.

Así mismo se hizo uso de clases y objetos para trabajar con memoria, también se hizo uso de listas para así poder implementar el conteo de los datos.

En la parte del frontend, se uso el framework Django, el cual es también de Python, y se instala de igual forma con pip mediante consola, en el mismo se hicieron las configuraciones en urls, en views, en donde se hacia la conexión con los endpoints de la api hecha en flask.



Figura 1. Conexión Django Flask

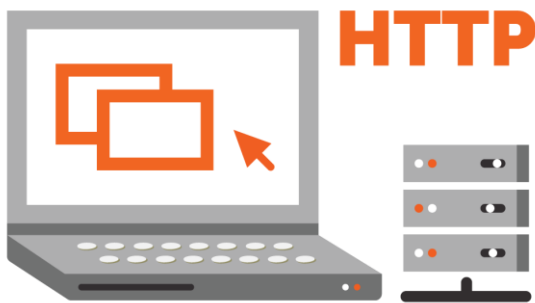
Fuente: Django en español

¿Qué es una Api?

Se trata de un conjunto de subrutinas, funciones y procedimientos que ofrece cierta biblioteca para ser utilizada por otro software con una capa de abstracción.

a. ¿Qué es un protocolo HTTP?

El Protocolo de transferencia de hipertexto es el protocolo de comunicación que permite las transferencias de información a través de archivos en la World Wide Web.



Forma gráfica del protocolo HTTP

Fuente: Developer Mozilla Foundation

b. ¿Qué es un servidor?

Es un conjunto de computadoras capaz de atender las peticiones de un cliente y devolverle una respuesta en concordancia. Los servidores se pueden ejecutar en cualquier tipo de computadora



Figura 2. Forma gráfica de un servidor

Fuente: elaboración propia

c. ¿Qué son peticiones?

Indican la acción que se desea realizar para un recurso determinado. Aunque estos también pueden ser sustantivos, estos métodos de solicitud a veces son llamados HTTP verbs. Cada uno de ellos implementan una semántica diferente, pero algunas características similares son compartidas por un grupo de ellos.

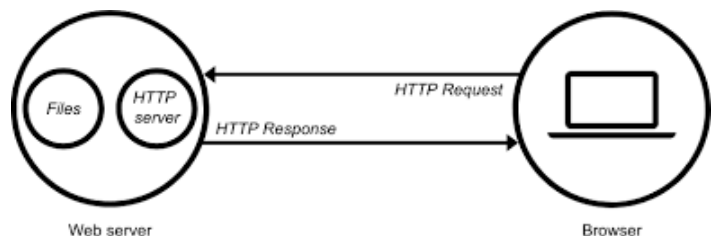


Figura 3. Forma gráfica de las peticiones

Fuente: Developer Mozilla Foundation

d. ¿Qué es cliente en un servidor?

Es un modelo de diseño de software en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Un cliente realiza peticiones a otro programa, el servidor, quien le da

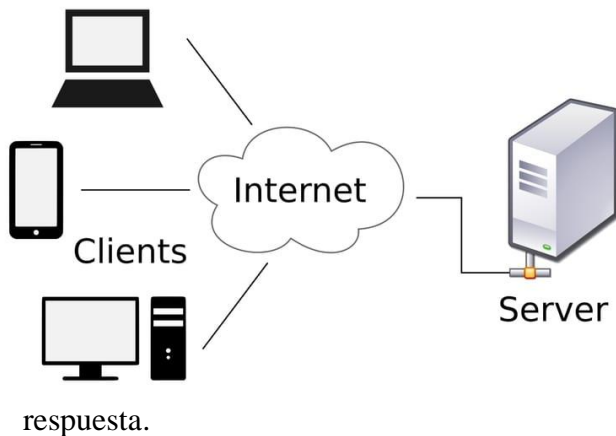


Figura 4. Forma gráfica del cliente servidor

Fuente: Developer Mozilla Foundation

a. ¿Qué es XML?

Es XML, siglas en inglés de eXtensible Markup Language, traducido como 'Lenguaje de Marcado Extensible' o 'Lenguaje de Marcas Extensible', es un metalenguaje que permite definir lenguajes de marcas desarrollado por el World Wide Web Consortium (W3C) utilizado para almacenar datos en forma legible. Proviene del lenguaje SGML y permite definir la gramática de lenguajes específicos (de la misma manera que HTML es a su vez un lenguaje definido por SGML) para estructurar documentos grandes. A diferencia de otros lenguajes, XML da soporte a bases

de datos, siendo útil cuando varias aplicaciones deben comunicarse entre sí o integrar información.¹

XML no ha nacido únicamente para su aplicación en Internet, sino que se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas. Se puede usar en bases de datos, editores de texto, hojas de cálculo y casi cualquier cosa imaginable.

XML es una tecnología sencilla que tiene a su alrededor otras que la complementan y la hacen mucho más grande, con unas posibilidades mucho mayores. Tiene un papel muy importante en la actualidad ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil.

```
import xml.etree.ElementTree as ET

root = ET.fromstring(countrydata)

# Top-Level elements
root.findall(".")

# All 'neighbor' grand-children of 'country' children of the top-level
# elements
root.findall("./country/neighbor")

# Nodes with name='Singapore' that have a 'year' child
root.findall("./year/..[@name='Singapore']")

# 'year' nodes that are children of nodes with name='Singapore'
root.findall("./*[@name='Singapore']/year")

# ALL 'neighbor' nodes that are the second child of their parent
root.findall("./neighbor[2]")
```

Figura 5. Estructura básica de eTree.
Fuente: Python Software Foundation

Conclusiones

- El uso de archivos XML, simplifica y separa datos de manera que al compartir información e intercambiar datos, hace que los datos estén disponibles más fácilmente.
- La visualización de datos a través de graficas hace que se facilite ver lo que obtienen las listas de objetos hechas por la app desde la parte del backend hacia el frontend.
- El uso de listas de Python en el proyecto hace que el uso de memoria se adapte dinámicamente al número de datos almacenados en la lista en cada momento.

Referencias bibliográficas

- Python. Stack Abuse. Recuperado 7 de marzo de 2021, de <https://stackabuse.com/reading-and-writing-xmlfilesin-python/>

Anexos

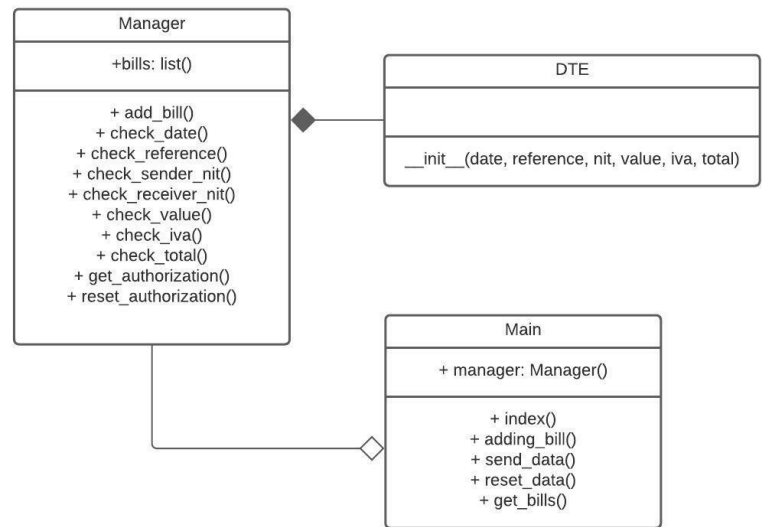


Figura 6. Diagrama de clases

Fuente: elaboración propia.

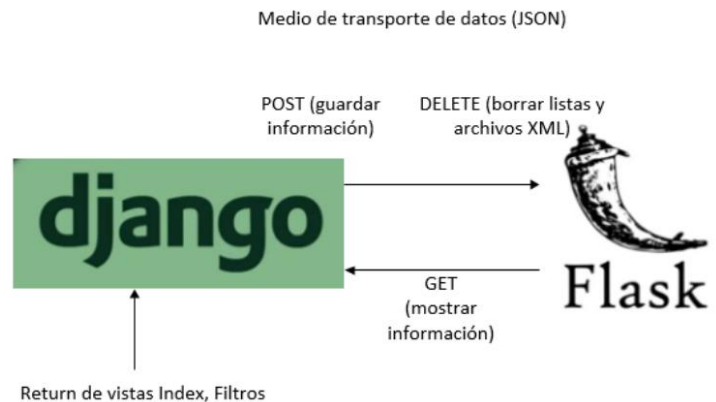


Figura 7. Diagrama cliente servidor.

Fuente: Django en español