Servicio Web

Que es un servicio web? – I

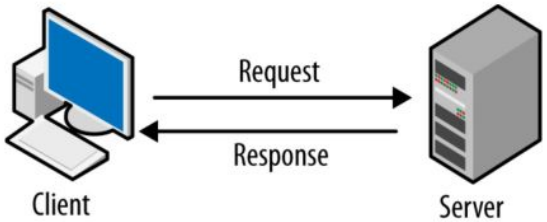
Un servicio web es un programa que provee una funcionalidad determinada y con el cual nos comunicamos a través del protocolo HTTP (el mismo utilizado por los navegadores para interactuar con un sitio web). Esta funcionalidad puede ser desde un simple servicio que recibe dos números y retorna su producto hasta un complejo modelo de inteligencia artificial que recibe una imagen de un jardín y nos retorna los nombres de todas las especies de plantas que se muestran en ella.

Que es un servicio web? II

Los ejemplos más comunes de un simple servicio web incluyen informes meteorológicos, cotizaciones de acciones y divisas, informes empresariales y gubernamentales, comunicación con redes sociales, etc. Cada servicio está identificado por una dirección de URL, podemos conectarnos a ella desde Python y podemos interactuar con la plataforma programáticamente como si lo estuviésemos haciendo manualmente desde alguna de las aplicaciones móviles o web. Otras plataformas como Facebook, twitter, Instagram, WhatsApp y MercadoLibre proveen servicios similares.

Protocolo – HTTP

En programación, siempre que se realiza una conexión entre dos computadoras, al iniciador de dicha conexión se lo conoce como cliente; y a quien recibe y acepta la conexión, servidor. Así, cuando visitamos un sitio web cualquiera, el navegador web es el cliente y el sitio web es el servidor; o cuando nos conectamos a un servicio web desde un programa de Python, éste es el cliente y aquél el servidor. El protocolo HTTP, por su parte, está montado sobre dos conceptos principales: petición (request) y respuesta (response).



Protocolo – HTTP

Luego de haberse establecido una conexión entre dos programas que se comunicarán vía HTTP, el cliente debe hacer una petición exigiendo alguna operación al servidor, y éste le devolverá una respuesta.

Por ejemplo, cuando ingresamos a https://www.educacionit.com/, el navegador (cliente) realiza una petición al sitio web (servidor) indicando que quiere ver la página principal, y éste le responde enviándole contenido correspondiente, que en este caso, por tratarse de un sitio web, será un código HTML.

Cuando interactuamos con un servicio web, los datos que éste envía como respuesta están codificados, por lo general, en formato JSON en lugar de HTML.

Protocolo – HTTP - Resumen

- ***Modelo Cliente-Servidor***: Tu navegador (cliente) envía una solicitud HTTP al servidor web donde está alojada una página. El servidor responde con los datos (HTML) de esa página.
- ***Intercambio de Mensajes***: Se comunican mediante mensajes de solicitud (del cliente) y respuesta (del servidor).
- ***Sin estado (Stateless)***: Tradicionalmente, HTTP no guarda información de sesiones anteriores entre solicitudes, aunque las cookies añaden estado a las interacciones.
- ***Capa de Aplicación:*** Es un protocolo de la capa de aplicación que se ejecuta sobre otros protocolos de red como TCP/IP para la transmisión de datos.
- ***HTTPS (HTTP Secure)*** añade una capa de seguridad (SSL/TLS) que cifra los datos, volviéndose ilegibles para interceptores, utilizando el puerto 443 en lugar del puerto 80 de HTTP.
- ***Métodos de petición HTTP:*** POST – GET – PUT – PATH – DELETE
- ***Códigos de estado:*** 200 OK - 201 CREATED - 204 NO CONTENT - 400 BAD REQUEST - 404 NOT FOUND - 500 INTERNAL SERVER ERROR

Arquitectura REST - I

Dijimos que la comunicación entre un servicio web y un consumidor de ese servicio (en nuestro caso, una aplicación de Python) se realiza vía el protocolo HTTP. Pero para que las cosas sean más simples, muchos servicios web tienen una estructura similar llamada REST.

REST, REpresentational State Transfer, se trata de una arquitectura estándar para comunicaciones web entre sistemas, logrando que se entiendan mucho mejor entre ellos. A los servicios que cumplen con este diseño se les llama RESTful API.

La arquitectura REST se basa en que el cliente envía peticiones para recuperar o modificar recursos, y el servidor responde con el resultado, que puede ser con los datos que hemos pedido o el estado de la petición.

Arquitectura REST II

Una petición está formada por:

- Un verbo HTTP que define la operación a realizar.
- Una cabecera o header que incluye información sobre la petición.
- Una ruta o path hacia un recurso.
- El cuerpo del mensaje o body con los datos de la petición.

Veamos un ejemplo. Consideremos un servicio web ficcional a partir del cual se gestionan los alumnos del instituto, cuya dirección es http://api.educacionit.com/.

Arquitectura REST III

Utilizando este servicio podemos obtener la lista de alumnos, la información de alguno en particular e incluso agregar nuevos y modificar o eliminar alumnos existentes. Pero también podremos realizar operaciones similares con instructores y personal administrativo del instituto. Así, para poder distinguir qué tipo de información queremos obtener, crear, modificar o eliminar, el servicio proveerá direcciones de URL más específicas, tales como:

- http://api.educacionit.com/alumnos
- http://api.educacionit.com/instructores
- http://api.educacionit.com/administrativos

Arquitectura REST IV

Cada una de estas “secciones” de un servicio web, en la terminología de la arquitectura REST, se denomina recurso. Así, quitando la primera parte del dominio para simplificar, tenemos tres recursos: /alumnos, /instructores y /administrativos . No obstante, por lo general, los recursos tienen nombre en singular (más adelante se verá más claramente por qué), de modo que serían, más bien, los siguientes:

- http://api.educacionit.com/alumno
- http://api.educacionit.com/instructor
- http://api.educacionit.com/administrativo

Ahora bien, el protocolo HTTP define un conjunto de verbos (con el nombre de métodos) para identificar qué tipo de operación se quiere ejecutar sobre un recurso determinado, a saber:

- GET, para leer información;
- POST, para agregar nueva;
- PUT, para modificar información preexistente;
- DELETE, para eliminar.
- PATCH, para modificar información preexistente.