

UNIVERSIDAD BOLIVIANA DE INFORMATICA
INGENIERÍA DE SISTEMAS



INVESTIGACIÓN REST

Asignatura:

Taller de Sistemas

Docente:

Ing. Daniel Alejandro Coronel Berrios

Estudiante:

Manuel Alejandro Choque Sanjines

La Paz - Bolivia

2022

REST

"Representational State Transfer" o traducido a "Transferencia de presentación de estado" es lo que se domina a REST. ¿Y eso es?, una técnica de arquitectura de software usada para construir APIs que permitan comunicar a nuestro servidor con sus clientes usando el protocolo HTTP mediante URIs lo suficientemente inteligentes para poder satisfacer la necesidad del cliente.

- REST es STATELESS, es decir que cada petición que reciba nuestra API debe perecer. Por ejemplo, no podemos RECORDAR un usuario logeado en el API usando una sesión, esto es un PECADO ya que agotaría la memoria RAM de nuestro servidor (10 mil usuarios conectados a nuestra API). Lo que correcto es pasar un TOKEN para cada petición realizada al API, y el API deberá validar si esta es correcta o no (por ahora no vamos hablar de técnicas para generar el TOKEN, pero lo más común es usar una COOKIE).
- Se implementan RECURSOS para generar comunicación, es decir crea URIs únicas que permiten al cliente entender y utilizar lo que está exponiendo. Por ejemplo:
 - `api.anexsoft.com/users`
 - `api.anexsoft.com/users/1405`
- Cada petición realizada a nuestra API responde a un verbo, y dicho verbo a una operación en común. Mediante los métodos HTTP hacemos las peticiones, lo común es GET y POST, PUT y DELETE.

POST (create): cuando mandamos información para insertar por ejemplo un registro en la base de datos. La información es enviado en el cuerpo de la petición, es decir que los datos no son visibles al usuario.

`api.anexsoft.com/users`

- GET (read): es usado para modo lectura, por ejemplo: cuando queremos listar a todos los usuarios de nuestra base de datos. Los parámetros son enviados por la URL.

`api.anexsoft.com/users`

- PUT (update): cuando queremos actualizar un registro. Actualizar la información de un usuario X.

`api.anexsoft.com/users`

- DELETE (delete): cuando queremos eliminar un registro. Borrar un usuario X de nuestra base de datos.

`api.anexsoft.com/users`

Con esto hemos mencionado algunas características básicas de lo que es REST, la cual podríamos decir que es un estándar para crear una REST Api o RESTFul.

¿Qué significa REST?

REST es una interfaz para conectar varios sistemas basados en el protocolo HTTP (uno de los protocolos más antiguos) y nos sirve para obtener y generar datos y operaciones, devolviendo esos datos en formatos muy específicos, como XML y JSON.

El formato más usado en la actualidad es el formato JSON, ya que es más ligero y legible en comparación al formato XML. Elegir uno será cuestión de la lógica y necesidades de cada proyecto.

REST se apoya en HTTP, los verbos que utiliza son exactamente los mismos, con ellos se puede hacer GET, POST, PUT y DELETE. De aquí surge una alternativa a SOAP.

Cuando hablamos de SOAP hablamos de una arquitectura divididas por niveles que se utilizaba para hacer un servicio, es más complejo de montar como de gestionar y solo trabajaba con XML.

Ahora bien, REST llega a solucionar esa complejidad que añadía SOAP, haciendo mucho más fácil el desarrollo de una API REST, en este caso de un servicio en el cual nosotros vamos a almacenar nuestra lógica de negocio y vamos servir los datos con una serie de recursos URL y una serie de datos que nosotros los limitaremos, es decir, será nuestro BACKEND nuestra lógica pura de negocios que nosotros vamos a utilizar.



¿Por qué debemos utilizar REST?

REST no es solo una moda, y es por las siguientes razones que esta interfaz está teniendo tanto protagonismo en los últimos años:

- Crea una petición HTTP que contiene toda la información necesaria, es decir, un REQUEST a un servidor tiene toda la información necesaria y solo espera una RESPONSE, ósea una respuesta en concreto.
- Se apoya sobre un protocolo que es el que se utiliza para las páginas web, que es HTTP, es un protocolo que existe hace muchos años y que ya está consolidado, no se tiene que inventar ni realizar cosas nuevas.

- Se apoya en los métodos básicos de HTTP, como son:
 - Post: Para crear recursos nuevos.
 - Get: Para obtener un recurso o un recurso en concreto.
 - Put: Para modificar.
 - Patch: Para modificar un recurso que no es un recurso de un dato, por ejemplo.
 - Delete: Para borrar un recurso, un dato por ejemplo de nuestra base de datos.
- Todos los objetos se manipulan mediante URI, por ejemplo, si tenemos un recurso usuario y queremos acceder a un usuario en concreto nuestra URI seria /user/identificadordeobjeto, con eso ya tendríamos un servicio USER preparado para obtener la información de un usuario, dado un ID.

Ventajas de REST

- Nos permite separar el cliente del servidor. Esto quiere decir que nuestro servidor se puede desarrollar en Node y Express, y nuestra API REST con Vue por ejemplo, no tiene por qué estar todos dentro de un mismo.
- En la actualidad tiene una gran comunidad como proyecto en Github.
- Podemos crear un diseño de un microservicio orientado a un dominio (DDD)
- Es totalmente independiente de la plataforma, así que podemos hacer uso de REST tanto en Windows, Linux, Mac o el sistema operativo que nosotros queramos.
- Podemos hacer nuestra API pública, permitiendo darnos visibilidad si la hacemos pública.
- Nos da escalabilidad, porque tenemos la separación de conceptos de CLIENTE y SERVIDOR, por tanto, podemos dedicarnos exclusivamente a la parte del servidor.



¿Quién usa REST?

Muchas empresas como Twitter, Facebook, Google, Netflix, LinkedIn y miles de startups y empresas usan REST. Todas estas empresas y servicios tienen su API REST por un lado con su

lógica de negocio y por otro lado su parte frontend, con lo cual nos permite centrarnos también un poco más en lo que es nuestra lógica de negocio haciendo una API REST potente.

Estas API pueden ser públicas y lo pueden consumir otros usuarios, con lo cual tenemos una forma de dar visibilidad a nuestra API y de testearla, no podemos olvidar que la parte de TEST es una de la parte más importantes, pues ¿por qué no nos serviría como TEST que otros usuarios prueben nuestra API REST y nos den Feedback?, pues aquí lo tenemos solo hay que securizarla y es lo que ocurre con estas empresas.

Bibliografía:

¿Qué es REST? Conoce su potencia

<https://openwebinars.net/blog/que-es-rest-conoce-su-potencia/>

¿QUÉ ES REST?

<https://anexsoft.com/que-es-rest>