Introducción a las API REST

Definición de API:

Una API es un conjunto de métodos o funciones que ofrece o expone un software para que otros softwares puedan interactuar con él.

La API REST son un conjunto de especificaciones que nos permiten crear aplicaciones web que se conecten a un servidor, siguiendo la arquitectura cliente/servidor. En línea con el mismo concepto de API, las API REST exponen funciones que permitirán a un cliente (navegador o móvil) se conecten con nuestra aplicación de backend mediante las API REST

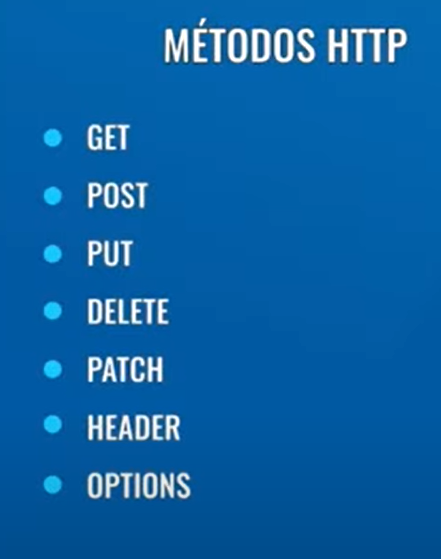
REST se basa en el protocolo HTTP (no es el único, pero el más utilizado en la actualidad) para exponer su arquitectura incluso las comunicaciones entre aplicaciones web utiliza la arquitectura REST bajo el protocolo HTTP.

El protocolo HTTP nos permite exponer reglas para cumplir con objetivos especifico (no confundir con un algoritmo) es un estándar de comunicación.

HTTP se basa en una comunicación cliente / servidor, donde el cliente realiza una petición y el servidor le da una respuesta (esto es solo de forma general)

Las peticiones para cumplir con el estándar deben tener una estructura determinada





Los métodos o verbos HTTP estándar identifica que tipo de petición esta haciendo el cliente al servidor

GET realiza una petición al servidor ej. Envíame un listado de usuarios

POST se utiliza para enviar informacion del cliente al servidor. Ej crear usuarios

PUT se utiliza para actualizar un recurso

DELETE se utiliza para eliminar recursos

PATCH actualización parcial del recurso (no se utiliza con frecuencia)

HEADER consulta la existencia de un recurso (no se utiliza con frecuencia)

OPTION permita validad dominios (lo hacen lo navegadores por seguridad)

Códigos de respuestas del servidor, significan como el servidor proceso la peticion



100 – 199 informativos No se utilizan en las API REST

¿Qué es REST?

Es la transferencia de estado representacional #!$&???? Vamos a explicarlo de forma simple.

En general la web se basa en la arquitectura REST, cuando entramos en un navegador y realizamos una petición, esta se basa en REST bajo el protocolo HTTP.

Las propiedades de la arquitectura REST (estas las tenemos que saber de memoria, es una pregunta recurrente que les hacen a los desarrolladores backend)



TODO EN LA WEB ES UN RECURSO: esto significa que cuando ingresas a una web, lo que estamos haciendo es pedir recursos al servidor

Los recursos tienen las siguientes características

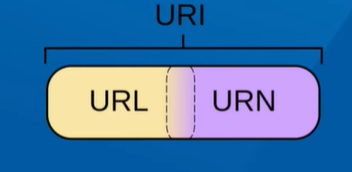
Deben tener un ID único

El recurso devuelto tiene que estar representado por un formato. El mas usados es JSON anteriormente se usaba XML

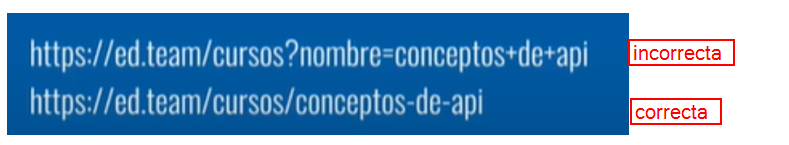
Las comunicaciones no tienen estados, las peticiones son independientes una de otras. Una petición no conoce nada de otra petición.

Que es URI y URL

Como una de las propiedades de REST nos dice que todo recurso debe estar identificado de manera única en la web para eso se utilizan las URI que pueden estar representadas por una URL o URN. Las URL son localizadores de recursos y las URN son los nombres de los recursos

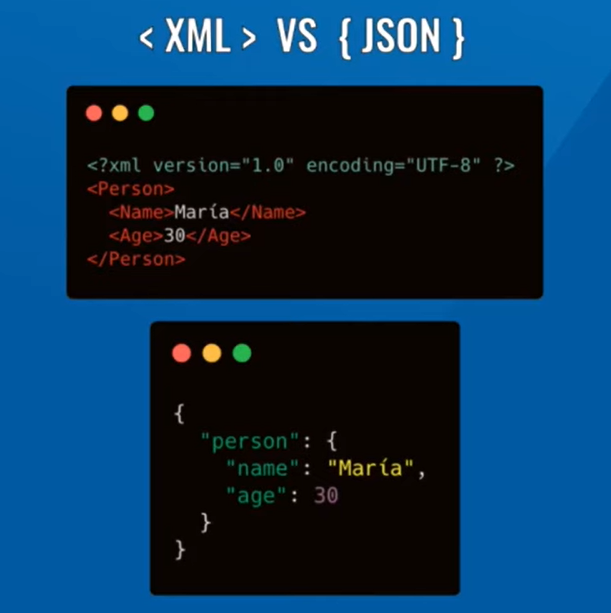


Las URN no se utiliza, generalmente se utilizan URL



Formatos de recursos





Comunicación sin estado (solución al login)

Cookies y Tokens son cadenas de caracteres que se generan a la hora de logearnos el servidor valida los datos y entrega una cookie o un token al cliente y en adelante cada petición que realiza el cliente las envía con la cookies o token que recibió del servidor.

Los tokens son mas usados en seguridad, el servidor firma el tokens y de este modo detecta cualquier modificación que se haya echo en el archivo tokens. Las cookies solo se utiliza en navegadores, mientras que los tokens se pueden usar además de los navegadores en lenguajes backend.

Que es la idempotencia

Es la característica que hace que aunque una petición se realice varias veces las respuesta es siempre la misma. La idempotencia aplica a los métodos HTTP. Algunos métodos son idempotentes (seguros) y otros métodos no lo son.

Los idempotentes son



¿Por qué los métodos idempotentes son seguros? Por ejemplo, si realizamos una petición GET al mismo usuario siempre nos devolverá el mismo resultado, no hay riesgo de que se modifique la informacion. En cambio, POST puede devolver resultados diferentes por cada registro enviado por lo tanto no es idempotente.

Hateoas (investiga sobre esta restricción en la arquitectura REST)

1. ¿Qué significa el acrónimo REST?
   1. Transformación de estado representacional
   2. Transferencia de estado representacional
   3. Representar transformación de estado
   4. Transferencia de estado representacional
2. ¿Cuál es el protocolo utilizado en REST?
   1. HTTP
   2. FTP
   3. XML
   4. URL
3. Los efectos secundarios de N > 0 solicitudes idénticas son los mismos que para una sola solicitud. Esto es..
   1. Equivalence
   2. Idempotency
   3. RESTFullness
   4. APIance
4. ¿Qué código de estado HTTP relacionado con la página no encontrada?
   1. 402
   2. 404
   3. 403
   4. 401
5. ¿Qué son los servicios web RESTful?
   1. Son servicios que deben cumplir con la arquitectura REST
   2. Deben devolver datos en formato JSON
   3. Están escritos en un solo lenguaje de programación
   4. Frontend y Backend residen en el mismo servidor
6. ¿Qué quiere decir que la tecnología REST es sin estado?
7. Explicar mediante un gráfico el flujo de una API RESTful
8. ¿Qué quiere decir que un método es idempotente?
9. ¿Cuál de las siguientes opciones sobre los servicios web RESTful es correcta?
   1. Los servicios web basados en la arquitectura REST se conocen como servicios web RESTful.
   2. B - Estos servicios web utilizan métodos HTTP para implementar el concepto de arquitectura REST.
   3. C : un servicio web RESTful generalmente define un URI, un identificador uniforme de recursos de un servicio, proporciona una representación de recursos como JSON y un conjunto de métodos HTTP.
   4. Todas las anteriores son correctas
10. ¿Cuál de los siguientes métodos HTTP debería ser de naturaleza idempotente?
    1. OPTIONS
    2. DELETE
    3. POST
    4. HEAD