



Technical Design - REST/HTTP

28-11-2022

—

Leandro Baier

Metas

Temas a revisar:

1. ¿Qué es REST?
2. ¿Por qué utilizarlo? Beneficios de una API orientada a verbos.
3. Mejores Prácticas de diseño: Definición de recursos, errores, paginación,etc.

1. ¿Qué es REST?

REST deriva de "REpresentational State Transfer".

La clave de REST es que un servicio REST no tiene estado (es stateless), lo que quiere decir que, entre dos llamadas cualesquiera, el servicio pierde todos sus datos. Esto es, que no se puede llamar a un servicio REST y pasarle unos datos (p. ej. un usuario y una contraseña) y esperar que "nos recuerde" en la siguiente petición.

2. ¿Por qué utilizarlo? Beneficios de una API orientada a verbos.

El protocolo REST en comparación con SOAP es simple, no quiere dar soluciones para todo y por lo tanto no pagamos con demasiada complejidad una potencia que quizá no vamos a necesitar.

La utilización de los verbos HTTP nos permite una mayor compresión del servicio que consumimos.

3. Mejores Prácticas de diseño: Definición de recursos, errores, paginación,etc.

- Usar sustantivos en las URIs en lugar de verbos, la acción está en el verbo HTTP.
- Métodos GET y los parámetros de consulta no deben alterar el estado
- Usar nombres en plural, por ejemplo:
 - /auto -> es incorrecto
 - /autos -> es correcto
- Usar subrecursos para establecer relaciones
 - /autos/409/conductores
 - /autos/409/conductores/34
- Usar cabeceras HTTP para la serialización de formatos

- Content-type: Application/json
- Usar HATEOAS
- Proveer filtrado, ordenación y paginación para colecciones
 - /autos?color=rojo
 - /autos?ordenarPor=marca&orden=asc
 - /autos?pagina=2&limite=20
- Versionar la API
 - /api/v1/autos
- Manejar errores con código de estado HTTP
- Usar el payload para los errores
- Permitir la sustitución del método HTTP
 - X-HTTP-Method-Override: DELETE