

HTTP API REST

¿Qué es REST?

- REST son las siglas de **REpresentational State Transfer**.
- Se trata de un estilo arquitectónico para el diseño de redes de software.
- El uso de esta arquitectura impone una serie de condiciones de cómo se debe desarrollar una API.
- Establecemos un grupo de protocolos para que todas las comunicaciones que hagamos a través de HTTP se desarrollen de la misma manera.

PUNTOS CLAVE

- **Cliente - Servidor.** Debe existir una separación de responsabilidades entre el cliente y el servidor.
- **Stateless.** Cada petición debe contener toda la información necesaria para ser procesada. De igual manera, la respuesta debe contener los datos requeridos en la petición o marcar de alguna manera el posible error.

PUNTOS CLAVE

- **Cacheable.** Las respuestas deben definir si pueden ser almacenadas en caché o no. Se debe especificar si esa respuesta es reutilizable en el futuro.
- **Interfaz uniforme.** Simplifica y desacopla la arquitectura. La estructura de las peticiones y de las respuestas siempre será la misma, independientemente de las tecnologías utilizadas en cliente o servidor.

MÉTODOS EN HTTP REST

- Dependiendo del método que utilicemos en la petición estaremos indicando una acción u otra:
 - **GET.** Recuperación de recursos.
 - **POST.** Creación de un recurso.
 - **PUT.** Edición de un recurso existente.
 - **DELETE.** Borrado de un recurso.

ESTRUCTURA DE LA PETICIÓN

- Además del método, dentro de una petición HTTP, podemos encontrar los siguientes elementos claves para el procesamiento de dicha petición.
- **URL**
 - Se trata de la dirección de un recurso dentro de la web.
 - Con ella definimos el recurso concreto al que estamos accediendo.
 - Está compuesta por varias partes, incluyendo el protocolo (http o https), el nombre de dominio, el path y posiblemente un fragmento o parámetros de consulta.

ESTRUCTURA DE LA PETICIÓN

- **HEADERS**

- Proporcionan información adicional sobre la petición o la respuesta.
- Las cabeceras son pares de nombre-valor que definen detalles como el tipo de contenido, la codificación, el estado del caché, la autenticación, entre otros.
- En las APIs REST, las cabeceras a menudo se utilizan para manejar la autenticación, utilizando diferentes técnicas como por ejemplo, JWT.

ESTRUCTURA DE LA PETICIÓN

- **BODY**

- El cuerpo de la petición HTTP es la sección de la petición donde se envían datos al servidor.
- Se utiliza principalmente en los métodos HTTP POST, PUT y PATCH, que suelen crear o actualizar recursos en el servidor.
- Los datos en el cuerpo de la petición pueden ser de diversos tipos, incluyendo texto plano, formularios URL codificados, datos en formato JSON y archivos binarios.

ESTRUCTURA DE LA RESPUESTA

- La respuesta enviada desde el servidor también dispone de una serie de elementos clave para su consumo desde el cliente:
- **CÓDIGOS DE ESTADO.**
 - Los códigos de estado HTTP indican el resultado de la petición. Por ejemplo, "200" significa que la petición fue exitosa, mientras que "404" significa que el recurso solicitado no se encontró.
 - Hay cinco clases de códigos de estado: 1xx (informativos), 2xx (exitosos), 3xx (redirecciones), 4xx (errores del cliente), y 5xx (errores del servidor).

ESTRUCTURA DE LA RESPUESTA

- **HEADERS**

- Al igual que las cabeceras de la petición, las cabeceras de la respuesta proporcionan información adicional sobre la respuesta.
- Las más importantes son todas las relacionadas con el tipo de datos incluidos en la respuesta.

ESTRUCTURA DE LA RESPUESTA

- **BODY**

- Esta es la parte de la respuesta donde el servidor envía los datos solicitados al cliente.
- Los datos pueden ser de cualquier tipo, como HTML, JSON, imágenes, etc., y el tipo de datos se especifica generalmente en la cabecera **Content-type**.