

Ejercicio 1:

Investigar y dar una definición con sus palabras los siguientes términos:

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.

REST se apoya en HTTP, creando peticiones que contienen toda la información necesaria, es decir, un REQUEST a un servidor, y solo espera una RESPONSE, osea una respuesta en concreto.

Las peticiones/funciones que nos permite hacer son:

- GET - obtiene información del servidor; no requiere un cuerpo de petición;
 - POST - crea un nuevo objeto en la base de datos, requiere un cuerpo de petición con los campos de este objeto;
 - PUT - restablece completamente un objeto específico en la base de datos;
 - PATCH - similar a PUT, restablece un objeto específico en la base de datos, pero sólo actualizará los campos que se mencionan en el cuerpo de la petición;
- Es una vía de acceso en el servidor.
- DELETE - elimina el objeto de la base de datos.

Una de sus ventajas es que nos permite separar el cliente del servidor. En resumen, permite que los clientes puedan realizar funciones para acceder a los datos del servidor. (Los clientes y los servidores intercambian datos mediante HTTP).

Api

Son mecanismos que permiten a dos componentes de software comunicarse entre sí mediante un conjunto de definiciones y protocolos.

API (Application Programming Interface) significa “interfaz de programación de aplicaciones” y permite habilitar el acceso a nuestros recursos y, al mismo tiempo, mantener la seguridad y el control.

No son la parte visible de una página, sino los circuitos internos que sólo los desarrolladores ven y conectan para hacer funcionar una herramienta y facilitar su interacción.

Ej: cuando cargamos los datos de una tarjeta la web utiliza una API para enviar esa información de forma remota a otro programa que verifica si los datos son correctos o es una tarjeta inventada. Una vez se verifica, este programa remoto le dice a la web que todo está en orden, y de esa forma podemos continuar.

Endpoint

Son las URLs de un API o un backend que responden a una petición. Es una pasarela que conecta los procesos del servidor de la aplicación con una interfaz externa. En otras palabras, es la dirección a la que se envían las peticiones.

Los endpoints funcionan junto con los métodos de la API (juegan un papel vital en su funcionamiento). Cada tipo de petición es un endpoint distinto.

Es un lugar de la API donde se produce el intercambio. Los puntos finales son URI (índices de recursos uniformes) en una API a los que una aplicación puede acceder.

Ej: Existe una API de Cat Facts. Esta API proporciona datos aleatorios sobre los gatos. Sin embargo, enumera varios **endpoints** con los que se puede solicitar información categorizada. Hay tres puntos finales disponibles:

/fact: Devuelve un único dato aleatorio sobre el gato.

/facts: Devuelve una lista de datos aleatorios sobre gatos.

/breeds: Devuelve una lista de razas de gatos.

Para hacer una solicitud a esta API y recuperar un dato sobre gatos, se tiene que añadir el punto final correcto (que es /fact) a la URL base de la API (que es <https://catfact.ninja/>). Esto dará la siguiente URL del punto final: <https://catfact.ninja/fact>

Encabezados (headers)

Son elementos que se utilizan dentro del establecimiento de las constantes de HTTP, las peticiones y respuestas, así como el uso de las API.

Forman parte del proceso de establecimiento de una petición para obtener una respuesta a un servidor o plataforma web, dado que el uso de este objeto permitirá añadir información adicional sobre lo que necesitamos integrar junto a la petición o respuesta.

Los tipos de cabeceras que usualmente se emplean en las llamadas a las API son:

- Cabecera general: no tienen ninguna relación con los datos de entrada ni de salida.
- Cabecera de consulta: están relacionadas con los datos del contenido que se obtiene.
- Cabecera de respuesta: están relacionadas con el contenido y el origen del servidor.
- Cabecera de entidad: se relacionan con la información sobre el cuerpo del contenido de una petición y la obtención de una respuesta.

Algunos ejemplos de los parámetros de cabecera HTTP que se pueden utilizar son:

- Host: Es la IP o la URL del servidor al que se hace la petición.
- User-agent: Este parámetro contiene información sobre el cliente y su Sistema Operativo (SO). Normalmente, indica el navegador que está utilizando y su versión.
- Accept-language: Esta línea indica al servidor qué idioma prefiere el cliente, en caso de que haya varias versiones del archivo que estás solicitando.
- Accept-encoding: Esta línea indica el tipo de codificación o compresión que el cliente puede procesar.
- Connection: Indica al servidor si debe mantener viva la conexión o establecer un tiempo de espera para la misma. Si la conexión se agota antes de completar la solicitud, devolverá un error.