1. Introducción
2. API REST
3. Crear una API

1.

Introducción

El termino API

APIs: aplication programming interface (interfaz de programación de aplicaciones). Conjunto de entripoints

Las APIs son herramientas que podemos construir y consumir, las cuales nos proveen de información de manera sencilla y accesible en un formato que para esta altura no debe ser desconocido para nosotros: el formato JSON.

Puntualmente, los objetivos que queremos abarcar en este módulo son:

* Descomponer los distintos componentes que arman un servicio de API.
* Construir la API en sí.
* Testear el funcionamiento de la API.
* Usar un software que nos permita testear los distintos tipos de endpoints de nuestras API.

Básicamente es un URL que no nos dará una vista, si no que funciona para enviar información para que la consuma otro sistema.

Ejemplo de ver netflix desde la tele y continuar viendo donde lo dejaste desde el celu

Son creadas para que dos sistemas se puedan comunicar.

Hay APIs públicas (gratuitas, solo nos registramos y listo. Ejemplo RESTCountries(nos da todos los países del mundo)), APIs semi privadas (Twitter, nos registramos y tenemos limites, pero si pagamos podemos tener más info) y privadas (netflix usan apis internas para comunicar distintas aplicaciones y que los endpoint no están disponibles para el usuario)

**ENDPOINT**: puntos de conexión donde necesitamos apuntar para obtener la info que queremos

Son las URL que debemos utilizar para obtener info de un servidos usando una API.

2.

API REST:

Es un sistema a través del cual un front-end se puede comunicar con un back-end de una manera mucho más organizada y funcional. Siempre pensando en la optimización y velocidad del sistema, así como también en la facilidad de uso del mismo.

Sistema rest es el protocolo o esquema que le indicará al sistema que quieran comunicarse con el de qué forma lo tiene que hacer.

HTTP es el protocolo para que el CLIENTE y el SERVIDOR se comuniquen como ya sabemos. HTTP forma el esquema de una llamada de teléfono (por ejemplo), pero no explica cómo vas a hablar en la llamada.

¿Cómo es el dialogo entre el cliente y el servidor? Se buscó lo que se conoce como la **arquitectura de servicio**. El estándar más conocido le llama **REST.**

4 características de REST:

1. SEPARA EN DOS CAPAS LA APLICACIÓN. Lo que es la interfaz del usuario y por otro todo lo que la aplicación provee como servicio que la interfaz consume. Una capa es la interfaz web para pedir productos a domicilio, mientras que la otra realiza la lógica, registros notifica a los proveedores, hace los pagos. Entonces, hay una URL que se especializa en una capa y otra URL en otra, ambas conectadas por APIs.
2. UBICACIÓN DE LOS RECUROS solo hay una ruta única para las info en el servidor. Ejemplo de spotify y la ruta /canciones o /damecanciones. Solo una debe funcionar.
3. STATELESS O “SIN ESTADO” no existe algo como sesión de usuario. Cada pedido es nuevo y único, donde debemos pasarle tooooda la info necesaria entre ellas permisos de accesos etc.
4. CACHEABLE (similar al 3) si el miso pedido se hace muchas veces. Para que el servido no se cance, debe renviar un encaberzado llamado max-age y su valor es la cantidad de segundo que tiene valides ese recurso. Cuando se explira el cliente debe volver a pedirlo

Recursos son todos los datos relevantes para el cliente.

Formatos de APIs más usados:

Documentación: <https://restfulapi.net/>

3.

Crear una API: