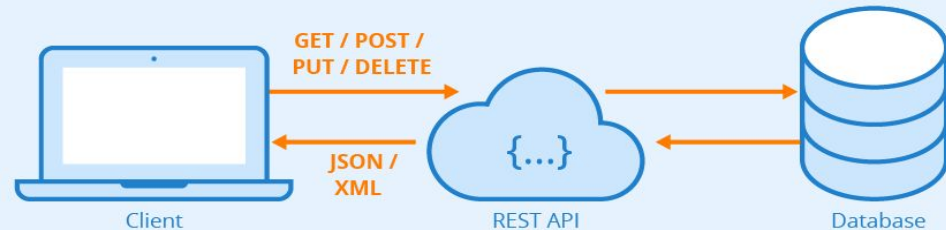


¿Qué es una API?

Una **API(interfaz de programación de aplicaciones)** es un conjunto de definiciones y protocolos que se utiliza para desarrollar e integrar el software de las aplicaciones.

Las **API** permiten que sus productos y servicios se comuniquen con otros, sin necesidad de saber cómo están implementados. Esto simplifica el desarrollo de las aplicaciones y permite ahorrar tiempo y dinero.



¿Qué es Postman?

Postman es una herramienta que se utiliza, sobre todo, para el testing de API REST, aunque también admite otras funcionalidades que se salen de lo que engloba el testing de este tipo de sistemas.

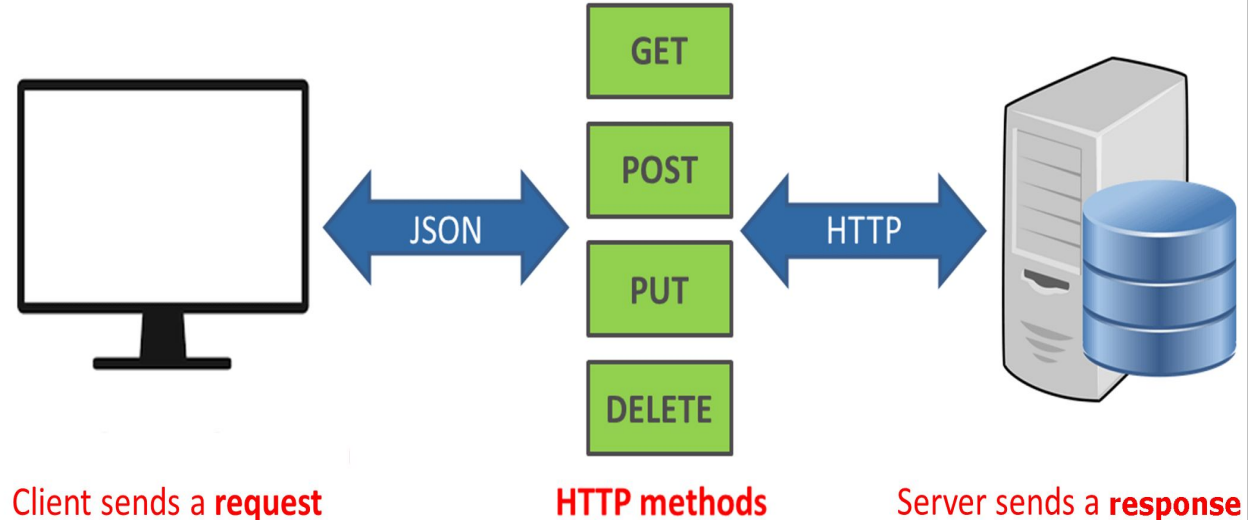
Gracias a esta herramienta, además de testear, consumir y depurar API REST, podremos monitorizar las, escribir pruebas automatizadas para ellas, documentarlas, mockearlas, simularlas.



Métodos de petición HTTP

HTTP define un conjunto de métodos de petición para indicar la acción que se desea realizar para un recurso determinado.

Recurso	GET	POST	PUT(PATCH)	DELETE
/api/v1/users	Obtener todos los registros	Crear nuevo registro	Actualizar registros	Eliminar registros
/api/v1/users/1	Obtener registro con id=1	N/A	Actualizar registro con id=1	Eliminar registro con id=1



GET: El método HTTP **GET** solicita una representación del recurso especificado. Las solicitudes que usan **GET** solo deben recuperar datos.

POST: El método **POST** se utiliza para enviar una entidad a un recurso en específico, causando a menudo un cambio en el estado o efectos secundarios en el servidor.

PUT: El modo **PUT** reemplaza todas las representaciones actuales del recurso de destino con la carga útil de la petición.

DELETE: El método **DELETE** borra un recurso en específico.

¿Cómo funcionan en Postman?

Authorization: Permite definir la autorización que utilizan todos los servicios de tu colección, una forma de autorización común en servicios es JWT que se explica en la sección [Seguridad](#).

Pre-request Scripts: Puedes agregar código javascript que se ejecute antes de la llamada a cada servicio rest. Por ejemplo puedes guardar un log para guardar la fecha actual en la que se ejecuta cada servicio.

Test: Aquí agregas código javascript para ejecutarse después de cada servicio, por ejemplo si todos los servicios que vas a probar regresan un estatus 200 lo puedes poner aquí en lugar de agregar el código para revisar que cada servicio regrese status 200.

Variables: Puedes cambiar o definir valores de variables que necesiten todos tus servicios.