

Introducción a API

Introducción a API	1
¿Qué aprenderás?	2
Introducción	2
API Rest	3
Ejemplos de APIs existentes	3
¿Cómo se usa una API?	4
Usando los métodos HTTP correctos	5
Postman	5
Nuestra primera consulta a una API	7
Ejercicio Guiado: Primer Request	7
JSON	8



¡Comencemos!

¿Qué aprenderás?

- Comprender el uso de API mediante los conceptos de request y response.
- Comprender la importancia de las API Rest para comunicar programas desde internet.

Introducción

La tecnología avanza día a día en nuestras vidas, nos sorprende con descubrimientos o creaciones nuevas, sin embargo, este avance no sería posible si no existiera un elemento clave como lo es Internet. Este último elemento nos facilita la conexión, la generación de contenido y el acceso a otras personas. Es por ello que en esta unidad veremos cómo adquirir desde Internet aquellos datos tan relevantes para la toma de decisiones en cualquier rubro. Usaremos funciones para obtener respuestas mediante algo denominado API y buscaremos cómo acceder a ella mediante lenguaje Java.

Una API es una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Esta tecnología permite que aplicaciones escritas en distintos lenguajes se integren y se ejecuten en plataformas diferentes. Desarrollar con este estilo de arquitectura otorga una manera sencilla de exponer servicios web y de entender cómo funcionan. La palabra API es un acrónimo de “Application Program Interface”, es decir, una interfaz de acceso que nos permite comunicar programas.

Existen diversos tipos de APIs, pero en esta ocasión hablaremos de API REST y, de ahora en adelante cuando utilicemos el concepto “API”, nos estamos refiriendo al concepto de API REST.

¡Vamos con todo!



API Rest

En términos muy simplificados la palabra REST proviene de “Representational State Transfer”, es decir, es un estilo de arquitectura que busca ordenar mediante reglas la creación de un servicio web. Es un programa que sigue una serie de restricciones para estandarizar la interconexión (interfaz), fácil de ser consultado y muy útil para integrar sistemas o comunicar distintas aplicaciones.

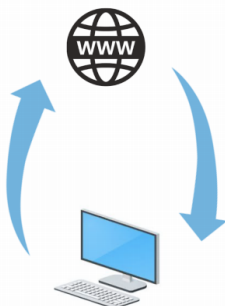


Imagen 1. Diagrama de conexión.
Fuente: Desafío Latam.

Al programa que consulta se le denomina “cliente”. Al programa que entrega la respuesta se le suele llamar “servidor”.

Ejemplos de APIs existentes

Existen miles de APIs para distintos propósitos.

- Clima y temperatura.
- Cambio de monedas.
- Indicadores económicos.
- Servicios para subir archivos.
- Compra y venta de criptomonedas.
- Servicios de geolocalización como Google Maps, etc.

Inclusive, cualquier persona puede construir una API y disponibilizarla a través de Internet. Estas son las denominadas “Fake APIs”.

¿Cómo se usa una API?

Para consumir o utilizar una API mediante un programa (cliente), debemos efectuar una acción llamada "request" o dicho de otra forma una solicitud hacia una dirección web específica (servidor). Esta solicitud nos traerá información de vuelta a nuestro programa en forma de respuesta o "response".

- Si la respuesta es correcta, nos desplegará la información que solicitamos.
- Si la respuesta no es correcta, nos dará un error que se desplegará en la consola.

Para ser más precisos, desde ahora en adelante llamaremos request a una dirección URL (Uniform Resource Locator) y response a la información que nos devuelve este servidor o servicio.

Otro concepto frecuente dentro del consumo de API es el denominado Endpoint. Esto es lo mismo que la parte final de la URL que buscamos como recurso. En otras palabras, son los terminales que expone una API.

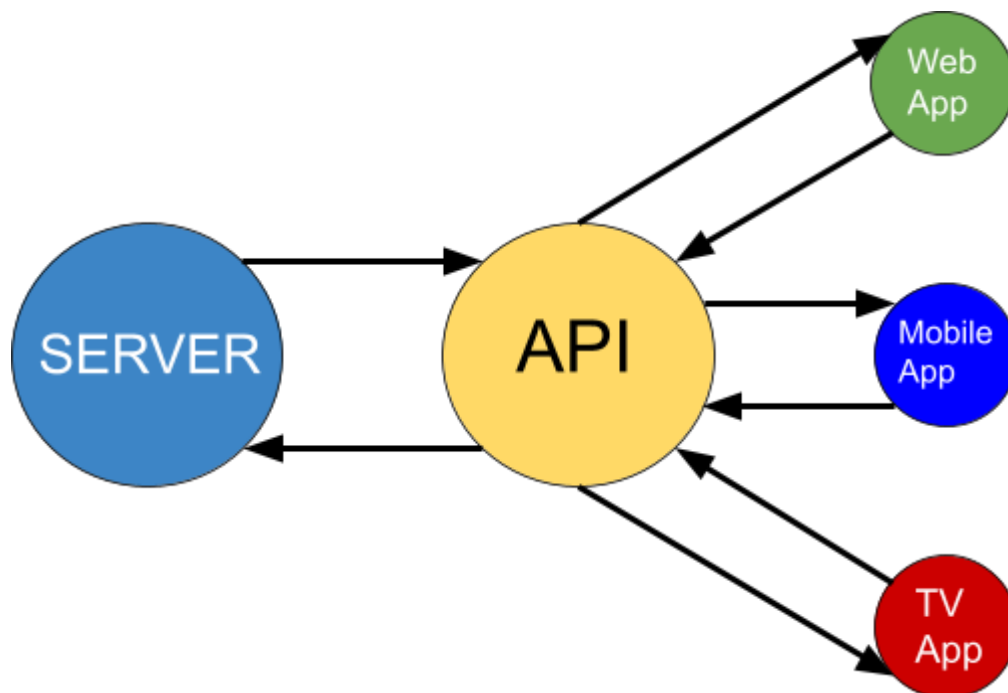


Imagen 2. Diagrama conexión API.
Fuente: Desafío Latam.

Usando los métodos HTTP correctos

Para indicar el tipo de operación que se llevará a cabo, existen diversos métodos HTTP. Estos buscan evitar la utilización de verbos en las rutas. Es importante destacar que el uso de GET es solamente para realizar consultas y no modifica estados ni realiza cambios.

Algunos de los métodos HTTP más usados son los siguientes:

- **GET** : Se utiliza para obtener un recurso o una colección de recursos.
- **POST** : Se utiliza para crear un recurso o una colección de recursos.
- **PUT** : Se utiliza para actualizar un recurso o una colección de recursos
- **DELETE** : Se utiliza para eliminar un recurso o una colección de recursos

Postman

Existe una herramienta muy potente para probar APIs que busca prevenir la necesidad de programar y que nos ayudará a comprender la idea. Esta herramienta se llama Postman.

Podemos descargar Postman desde la página oficial. <https://www.getpostman.com>

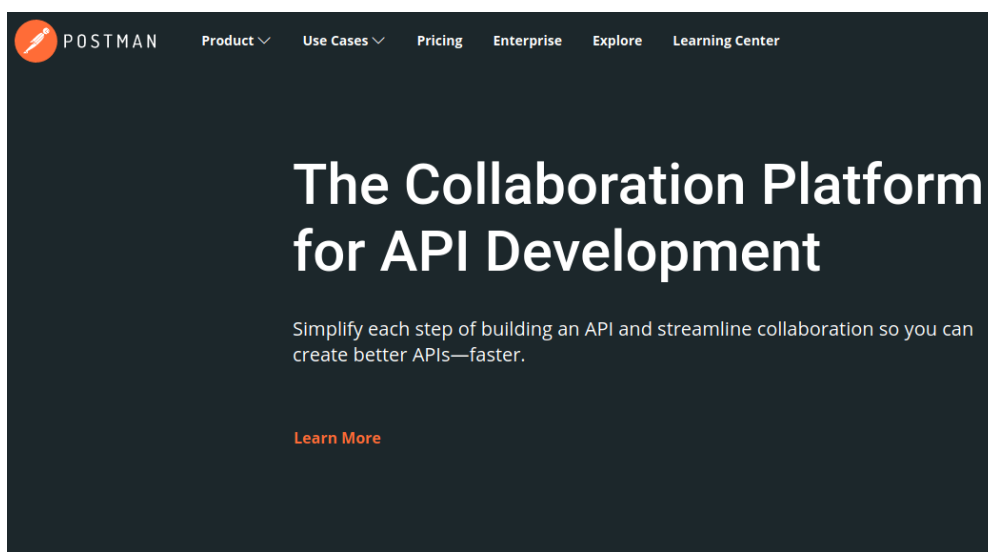


Imagen 3. Página oficial de Postman.
Fuente: Desafío Latam.

Es importante destacar que para instalar la aplicación debemos registrarnos en la página y luego buscar la opción de descarga, tal como se muestra en la imagen a continuación.

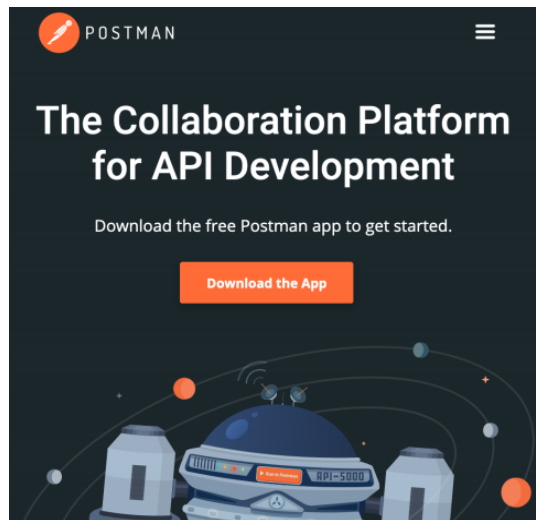


Imagen 4. Pantalla de descarga Postman.
Fuente: Desafío Latam.

Una vez instalada la aplicación, la abrimos y nos encontraremos con una interfaz semejante a la que se muestra en imagen.

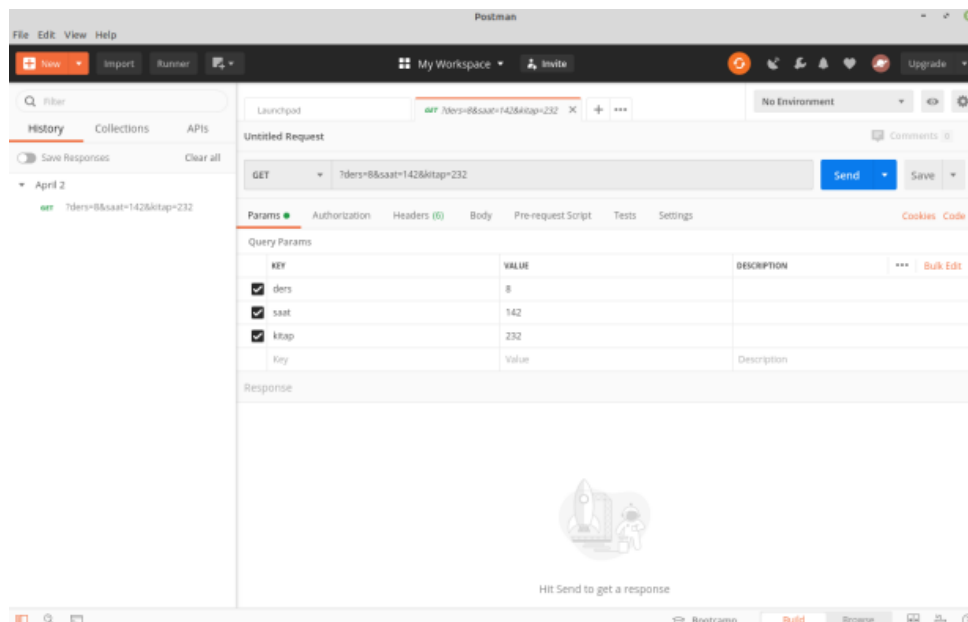


Imagen 5. Interfaz de Postman.
Fuente: Desafío Latam.

Nuestra primera consulta a una API

Para realizar nuestra primera consulta usaremos una API llamada jsonplaceholder:

<https://jsonplaceholder.typicode.com/>

Esta API contiene datos falsos y no tiene información real, pero nos sirve para darnos el primer acercamiento a este mundo. Funciona igual que el resto de las APIs y veremos un ejemplo a continuación.

Ejercicio Guiado: Primer Request

- **Paso 1:** Para comenzar a utilizar Postman, debemos colocar la siguiente URL en la casilla Request URL:

<https://jsonplaceholder.typicode.com/posts>

- **Paso 2:** Lo que haremos será realizar una petición GET para obtener un recurso, por lo tanto, se debe definir como tipo el método GET y luego hacer clic donde dice “send”. Obtendremos nuestra respuesta la cual será una lista de publicaciones.

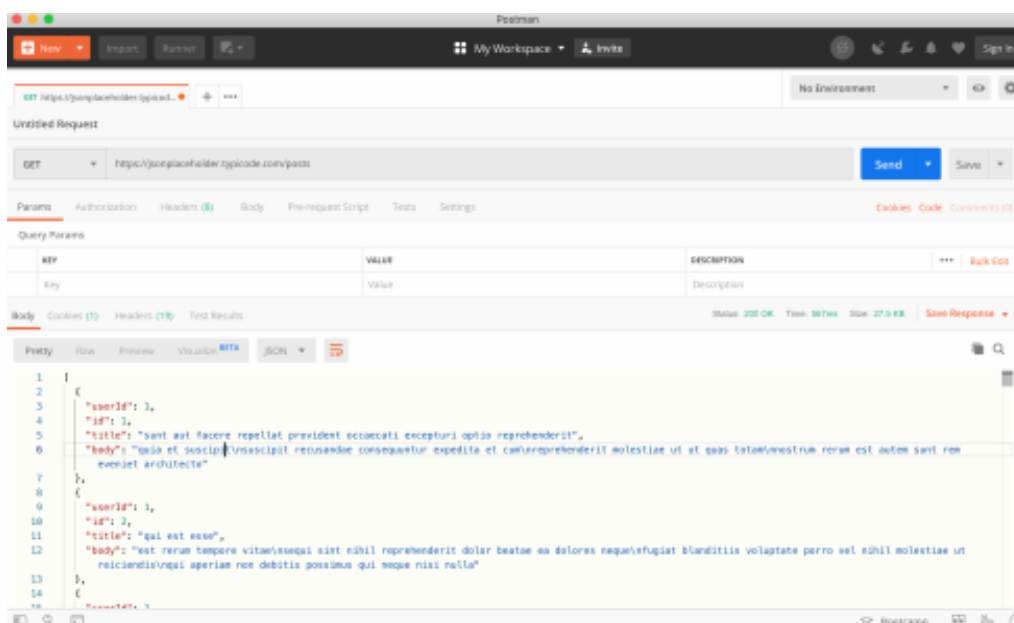


Imagen 6. Obteniendo URL.
Fuente: Desafío Latam.

- **Paso 3:** Luego de algunos milisegundos o segundos, el servidor nos enviará una respuesta en caso de ser exitosa. Aparecerá un texto en la parte inferior tal como se muestra en la imagen número 7.

```
▼ 0:
  userId: 1
  id: 1
  ▼ title: "sunt aut facere repellat provident occaecati excepturi optio reprehenderit"
  ▼ body: "quia et suscipit\nsuscepit recusandae consequuntur expedita et cum\nreprehenderit molestiae ut ut quas totam\nnostrum rerum est autem s
▼ 1:
  userId: 1
  id: 2
  title: "qui est esse"
  ▼ body: "est rerum tempore vitae\nsequi sint nihil reprehenderit dolor beatae ea dolores neque\nfugiat blanditiis voluptate porro vel nihil mole
▼ 2:
  userId: 1
  id: 3
  ▼ title: "ea molestias quasi exercitationem repellat qui ipsa sit aut"
  ▼ body: "et iusto sed quo iure\nvoluptatem occaecati omnis eligendi aut ad\nvoluptatem doloribus vel accusantium quis pariatur\nmolestiae porro
▼ 3:
```

Imagen 7. Formato JSON visto desde Firefox.
Fuente: Desafío Latam.

La respuesta es el contenido que traemos directamente desde internet y se parece mucho a algunas estructuras que ya conocemos.

JSON

La respuesta está en un formato llamado **JSON** y se destaca por separar los objetos con las siguientes llaves { }, sin embargo es importante preguntarnos ¿Qué es JSON?

JSON es un acrónimo de "**Javascript Object Notation**". Es un formato para enviar información en texto plano, fácilmente legible por humanos y fácilmente analizable por lenguajes de programación. Hoy en día es uno de los formatos más utilizado para enviar información entre sistemas.

Que el acrónimo tenga la palabra Javascript no quiere decir que necesitemos saber del lenguaje Javascript o utilizar Javascript. El formato JSON es de tipo plano y lo podemos utilizar desde cualquier lenguaje.