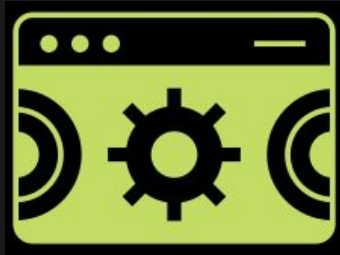


Programación Backend NodeJS

# Introducción a API REST



# Definición de API



**API / Application Programming Interface** / Interfaz para Programación de Aplicaciones:

Se trata de una **interfaz que se ejecuta como servicio en un servidor** (backend en términos de desarrollo web), **capaz de recibir solicitudes de clientes** (frontend) en un formato predefinido, y **devolver respuestas** con datos organizados también bajo un formato estándar.

# Definición de API REST



## REST / Representational State Transfer:

Una arquitectura muy utilizada actualmente en la creación de APIs. **Aporta un conjunto de reglas** para mantener un estándar en la organización del servicio.

**Una API REST no es más que una API estructurada bajo arquitectura REST**, que utiliza el protocolo http / https para comunicación.

# Conceptos de una API REST



Una **API REST** posibilita que un cliente realice una solicitud mediante protocolo web **http / https**, y reciba datos “**crudos**” organizados en un formato normalizado (JSON por ejemplo). El cliente se encargará de procesar y mostrar estos datos con la distribución espacial y estética que necesite.

Es importante recordar:

- La **API** se concentra en servir los datos específicos, no en su presentación.
- **Es independiente del lenguaje:** cliente y servidor no requieren utilizar el mismo lenguaje y arquitectura. Es totalmente válido por ejemplo, un frontend escrito en VueJS (Javascript) solicitando datos a una API elaborada en Python. Otro cliente (escrito en C++ o Go), utilizará el mismo formato de solicitud al conectarse con esa misma API Python.

# APIs locales y remotas



Si bien el concepto de API se encuentra íntimamente ligado con Internet y las conexiones **cliente / servidor**, también puede operar de forma **local**.

Como ejemplo podemos citar APIs locales en teléfonos móviles, que permiten un intercambio de datos entre diversas apps instaladas, realizando en esos casos solicitudes locales. Esas mismas apps podrán también conectarse con APIs remotas para otros requerimientos.

# Métodos de solicitud



Ya que una API REST se basa en solicitudes `http / https`, al momento de solicitar un contenido, el cliente deberá conectarse a una **URL específica** y utilizar un **método web aceptado**, por ejemplo:

- **GET:** para recuperar datos. Si se debe pasar algún parámetro a la API, la única forma de hacerlo es incluirlo en la propia URL.
- **POST:** para enviar datos nuevos. La API recibirá esta información para procesarla y almacenarla por ejemplo en una base de datos.
- **PUT:** para actualizar datos, por ejemplo, enviar un nuevo número de teléfono para cambiar en una cuenta de usuario existente.
- **DELETE:** para borrar recursos existentes.

# Métodos de solicitud



**Una API no soportará necesariamente todos los métodos**, dependerá de las funcionalidades que los desarrolladores hayan decidido habilitar y su organización interna.

Puede que ciertas funcionalidades no se encuentren activas bajo la buena práctica habitual. Por ejemplo, al momento de enviar una solicitud **para borrar un recurso, lo correcto sería realizarla empleando el método DELETE**, pero esta funcionalidad **podría estar habilitada mediante POST** en una URL diferente (endpoint) y ejecutar el mismo proceso en el servidor.

# Métodos de solicitud. Tokens



Un **token** esencialmente es una credencial digital encriptada, un mecanismo que permite enviar datos de identificación para validarse al momento de conectarse a un determinado sistema.

En las API REST, el tipo más utilizado es el **JWT**. Si la API requiere autenticación, permitirá enviar este token como parámetro de URL (GET), o token de autorización (POST).

Encoded

PASTE A TOKEN HERE

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoxNTE2MjM5MDIyfQ.SflKxwRJSMeKKF2QT4fwpMeJf36P0k6yJV_adQssw5c
```

Decoded

EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

PAYLOAD: DATA

```
{
  "sub": "1234567890",
  "name": "John Doe",
  "iat": 1516239022
}
```

VERIFY SIGNATURE

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  your-256-bit-secret
) ☐ secret base64 encoded
```





# Métodos de solicitud. URI

## URI / Universal Resource Identifier:

Se trata de un **identificador** universal que permite indicar dónde encontrar un recurso **específico**, como puede ser una imagen o un documento web. Lo utilizamos constantemente dentro del navegador, solo que de forma transparente sin percatarnos.

Dentro de las API REST, la URI hace referencia a la **dirección completa que se utiliza en la llamada**, incluyendo los eventuales parámetros dentro de la misma URL, para solicitar el paquete específico de datos que nos interesa, por ejemplo, mediante método GET:

GET <http://idux.net:3030/api/rnd>

# Métodos de solicitud. URI



GET <http://idux.net:3030/api/rnd>

- En este ejemplo, realizamos una **solicitud mediante método GET y conexión normal** (http).
- La solicitud se efectúa al **puerto 3030** del servidor que administra lo relacionado al **dominio idux.net**, es decir, el servidor en el cual está corriendo la API a la que deseamos comunicarnos.
- El **endpoint** específico que consultamos es **rnd**, y nos retornará una cadena de 8 caracteres generada al azar, mezclando letras y números.

# Códigos de estado



Toda solicitud http/s retornará un código (mensaje) de estado standard que indicará el resultado:

- **2xx: proceso exitoso**. Si la solicitud retorna un código 200, significa que todo funcionó OK.
- **3xx: redirección**. El contenido que intentamos acceder ha sido movido de forma temporal o permanente a otra dirección.
- **4xx: solicitud no válida**. Recibiremos un código 404 si intentamos conectarnos a un recurso inexistente, puede ser un error de tipeo, tal vez el recurso se ha trasladado y no se activó la redirección necesaria, o estamos intentando acceder a una ubicación sin autorización (403).
- **5xx: error interno de servidor**. La solicitud provocó algún tipo de conflicto en la ejecución de código del lado del servidor.

# Formato de datos



**Las APIs pueden retornar datos en formatos muy variados, incluso texto plano,** pero se destacan JSON, XML, CSV.

**En la actualidad el más utilizado para API REST es JSON (Javascript Object Notation** / Notación de Objeto de Javascript).

El / los formatos disponibles dependen enteramente de lo que soporta la API en cuestión y no pueden ser modificados del lado del cliente. No obstante, es habitual en muchos casos, que la API permita el envío de un parámetro adicional en las solicitudes, para que el cliente indique en qué formato desea recibir los datos.



## ACTIVIDAD EXTRA CLASE

# API JsonPlaceholder y OpenWeatherMap

Códigos de referencia: [repo de Github](#):

- Escribir script para realizar solicitudes GET y POST de práctica hacia endpoints en **JsonPlaceHolder**. Utilizar **axios** o **node-fetch** para ello.
- Crear cuenta gratuita en **OpenWeatherMap**, generar un token propio (app\_id) y luego elaborar script para obtener **datos de clima actuales en tu ciudad**.
- Generar un tercer **script de tipo servidor** con una API simple basada en **Express**. Debe realizar una **solicitud remota** a cualquiera de las dos APIs de prueba, y **servir ese mismo resultado en un endpoint** propio. Probarla localmente.

## {JSON} Placeholder

Free fake API for testing and prototyping.

Powered by [JSON Server](#) + [LowDB](#). Tested with [XV](#).

As of Oct 2022, **serving ~1.7 billion requests each month**.

for in your city

Guide API Dashboard Marketplace Pricing Maps Our Initiatives Partners Blog For

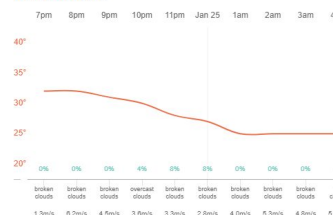
Jan 24, 07:52pm  
Rafaela, AR

32°C

Feels like 31°C. Broken clouds. Moderate breeze  
A 5.2m/s S. 1003hPa  
Humidity: 35% Dew point: 14°C  
Visibility: 10.0km



Hourly forecast



8-day forecast

Tue, Jan 24	40 / 22°C	clear sky
Wed, Jan 25	34 / 24°C	light rain
Thu, Jan 26	38 / 22°C	light rain
Fri, Jan 27	29 / 21°C	moderate rain
Sat, Jan 28	28 / 21°C	light rain
Sun, Jan 29	34 / 21°C	clear sky
Mon, Jan 30	36 / 20°C	scattered clouds
Tue, Jan 31	37 / 24°C	light rain