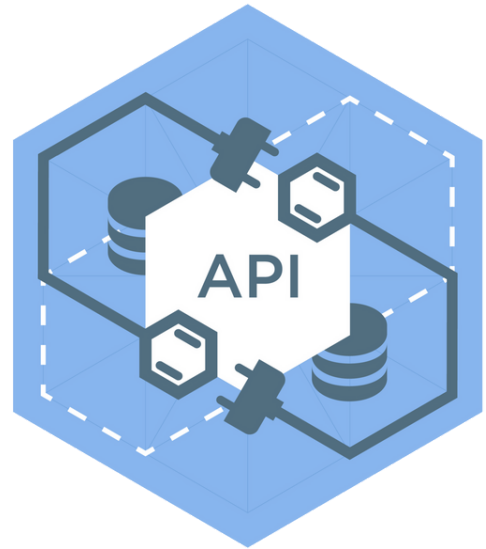


Mejores prácticas de una API REST

¿Qué es una API?

Una API (Interfaz de programación de aplicaciones) es un conjunto de funciones que permiten la comunicación entre dispositivos.



¿Qué es REST?



Rest se creó como una arquitectura a principios del 2000, y utiliza al protocolo HTTP para generar la transferencia de datos

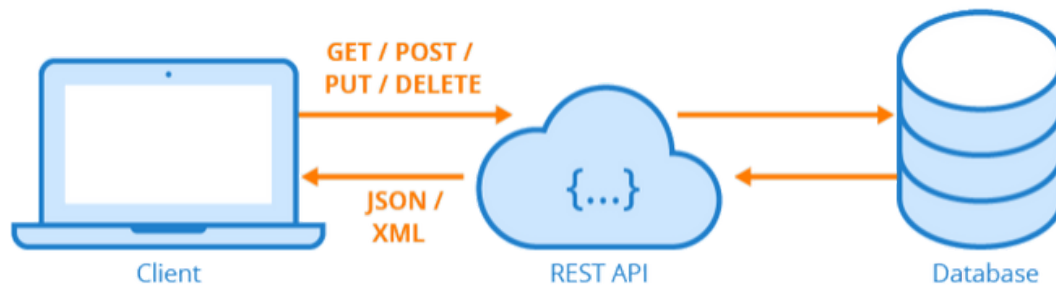
¿Qué es HTTP?

Es el 'idioma' que permite la transferencia de datos que pueden viajar bajo el formato HTML, JSON, XML, *etc.*

Mejores prácticas de un API REST

¿Qué es una API-REST?

API REST es un estilo de arquitectura de Software que permite consumir datos o usar una determinada funcionalidad de un Servicio o Producto alojado en Servidores Informáticos haciendo uso de rutas específicas.



Mejores Prácticas de API REST

1 APIs con rutas Concretas

En API REST, es mejor hacer rutas **Concretas** y **Definidas**, por ejemplo a continuación se tienen 2 versiones de una ruta de API REST para acceder a los datos de un Blog

Versión 1
/entidades

Versión 2
/autores
/blogs
/bloposts

Mejores Prácticas de API REST

URI con Formato Sencillos 2

Se debe de hacer rutas lo más sencillas posible, siempre en minúsculas, en algunos casos se debe evitar lo siguiente:

No hacer

```
/listarPostsBlog  
/actualizarPostBlog/11  
/eliminarPostBlog/11  
/obtenerAutor/4  
/eliminarAutor/4  
/actualizarAutor/4
```

Si hacer

```
/postsblog/listar  
/postsblog/actualizar/11  
/postsblog/eliminar/11  
/obtener/autor/4  
/eliminar/autor/4  
/actualizar/autor/4
```

3 Manejo de Errores

Proporcionar un código de **Estado** con una breve descripción explicando lo que ocurrió y brindar una página con los códigos de respuesta de la API.

```
{  
  "errors": [  
    {  
      "code": 215,  
      "message": "Bad Authentication"  
    }  
  ],  
  "data": {}  
}
```

Mejores Prácticas de API REST

4

Códigos de Estado

Las rutas de API REST utilizan un Código de estado HTTP que describe múltiples respuestas posibles

200 OK

400 Bad Request

500 Internal Server Error

201 Created

204 No Content

401 Unauthorized

403 Forbidden

404 Not Found

5

Seguridad

La seguridad es importante en un API REST, usa mecanismos y protocolos HTTP que pueden ser vulnerables, una forma de proteger el acceso a las rutas, es usando Tokens de Seguridad,

// Leer Postres

//Uso de JWT

<https://undominio.com/api/v1/tiendadepostres/postres?token=AadsdDADQEE345sfsfd34343D>

Mejores Prácticas de API REST

6 Métodos de petición HTTP

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

GET

Solicita una representación de un recurso específico. Las peticiones que usan el método GET sólo deben recuperar datos.

HEAD

Solicita una respuesta idéntica a la de una petición GET, pero sin el cuerpo de la respuesta.

POST

Enviar una entidad a un recurso en específico, causando un cambio en el estado secundario en el servidor.

Mejores Prácticas de API REST

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.

CONNECT

El método CONNECT establece un túnel hacia el servidor identificado por el recurso.

Mejores Prácticas de API REST

OPTIONS

Es utilizado para describir las opciones de comunicación para el recurso de destino.

TRACE

Realiza una prueba de bucle de retorno de mensaje a lo largo de la ruta al recurso de destino.

PATCH

El método PATCH es utilizado para aplicar modificaciones parciales a un recurso.

Fuente:

<https://developer.mozilla.org/es/docs/Web/HTTP/Methods>