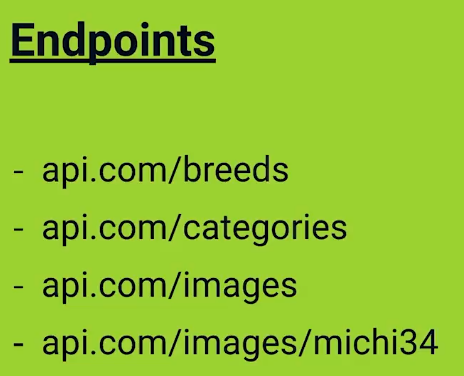
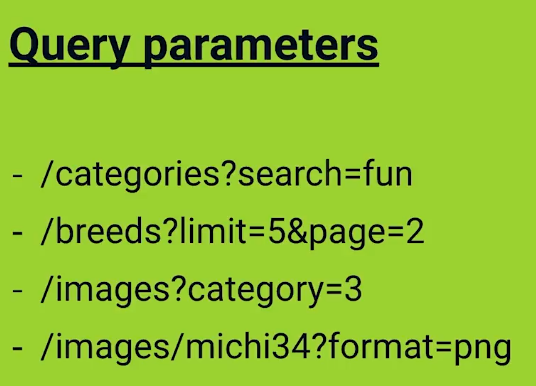
**Endpoints y Query Parameters**

Son formas de modificar los resultados que da el API REST

Endpoints son rutas (ver docu de la API para entender)



Query parameters, es info extra para especificar o limitar cual es el contenido que queremos pedirle al API, al backend . Es algo que agregamos al endPoint para precisar



EJEMPLO de los MICHIS



Solicitud a la API con sintaxis de promesa Fetch

Se puede hacer lo mismo con Async/ wait….



**¿Qué son los HTTP Status Codes?**

O como reaccionar ante los errores.

Los HTTP status code van del 100 al 500 y pucha

Del 100 al 200 🡪 Todo Ok, tanto front como back

3XX 🡪 REDIRECT: la solicitud no es la ruta final. 307 redirect temporal. 308 permanent.

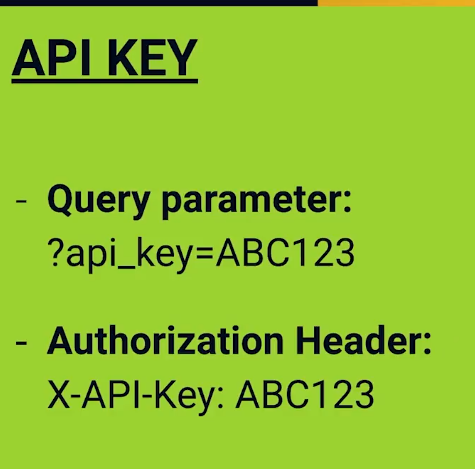
4XX 🡪 Front hizo solicitud incorrecta. 401 Endpoint necesita método de autorización. 402 necesita pagar para entrar a esa ruta. 404 Not found, no existe.

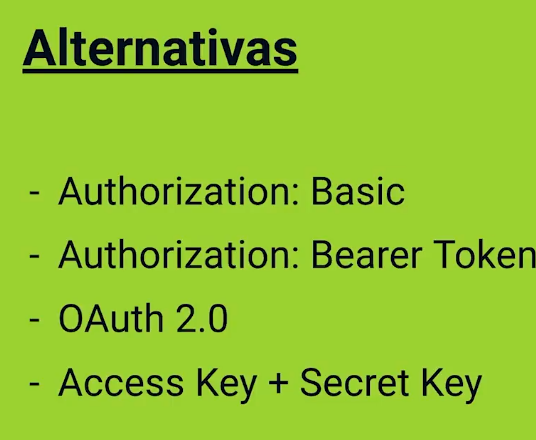
5XX 🡪 BackEnd falló. Tantas solicitudes que no alcanza a responder.

Autenticación-> quien es quien

Autorización-> permisos

Con las API key el backEnd sabe quien es quien, y pone limite a las autorizaciones:





# Qué son los Métodos HTTP?

El front le hace spoiler al backend del tipo de solicitud que vamos a hacer



GET: normales, leer info

POST: crear algo nuevo. Crear nuevo elementos

PUT y PATCH: editar la info que ya esta. PUT editar todo. PATCH editar un pedacito

DELETE: borar info Da!



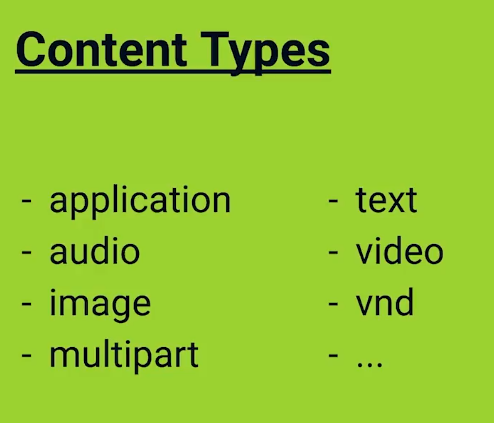
Cuando el Fetch es tipo POST, lo especificamos como un segundo argumento, y le incluimos unos valores por manualmente (JuanDC va a a explicar luego que son esos valores)

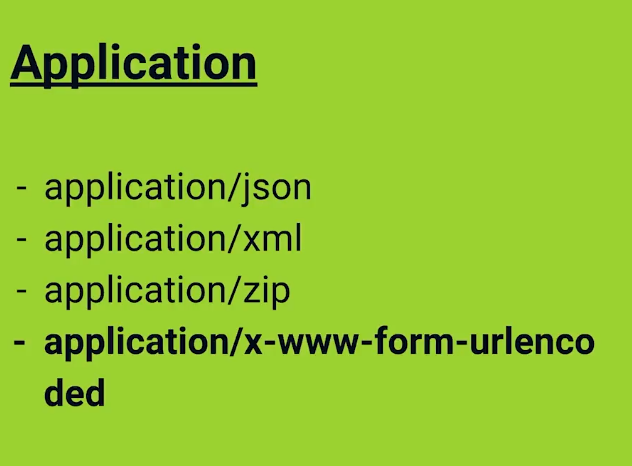
.appenChild método-🡪 inserta

API Headers:

Meta información sobre….

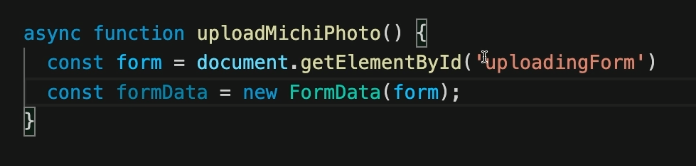






Metodo .FormData

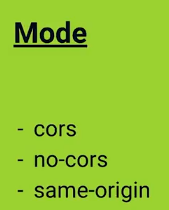
Sirve para formularios

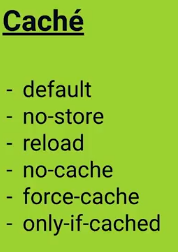


Al new FormData le paso como argumento form ósea la info que estoy poniendo en el Input HTML

# Axios: librerías de JavaScript para consumir APIs

Otras prodiedades de FETCH









**REST = protocolo de transferencia de datos por HTTP**

GraphQL = que no hayan muchos endpoints