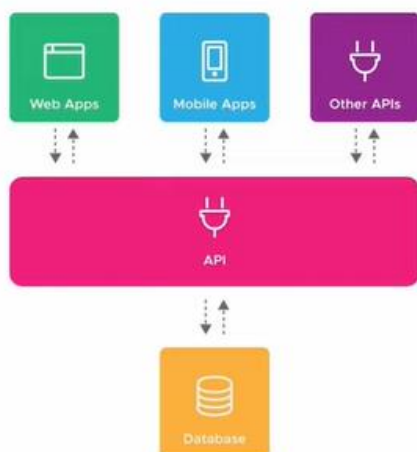


Objetivos

- 1.** Explicar o que são APIs
- 2.** Ensinar como lidar com requisições dessas APIs
- 3.** Apresentar o método `fetch()`

O que são APIs

Application Programming Interface



Uma API é uma forma de intermediar os resultados do back-end com o que é apresentado no front-end.

Você consegue acessá-la por meio de URLs.

JSON: JavaScript Object Notation

```
bank.json x
1
2
3 {
4   "description": "schema POST bank",
5   "type": "object",
6   "properties": {
7     "id": {
8       "type": "number",
9       "minimum": 0
10    },
11    "code": {
12      "type": "string"
13    },
14    "name": {
15      "type": "string"
16    }
17  }
18 }
19
```

É muito comum que APIs retornem seus dados no formato .json, portanto precisamos tratar esses dados quando os recebermos.

fetch

Consumindo APIs

```
fetch(url, options)
  .then(response => response.json())
  .then(json => console.log(json))

// retorna uma Promise
```

Operações no banco (POST, GET, PUT, DELETE, etc)

```
fetch('https://endereco-api.com/', {
  method: 'GET',
  cache: 'no-cache',
})
.then(response => response.json())
.then(json => console.log(json))

// retorna uma Promise
```

```
fetch('https://endereco-api.com/', {
  method: 'POST',
  cache: 'no-cache',
  body: JSON.stringify(data)
})
.then(response => response.json())
.then(json => console.log(json))

// retorna uma Promise
```