

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.

O que são APIs

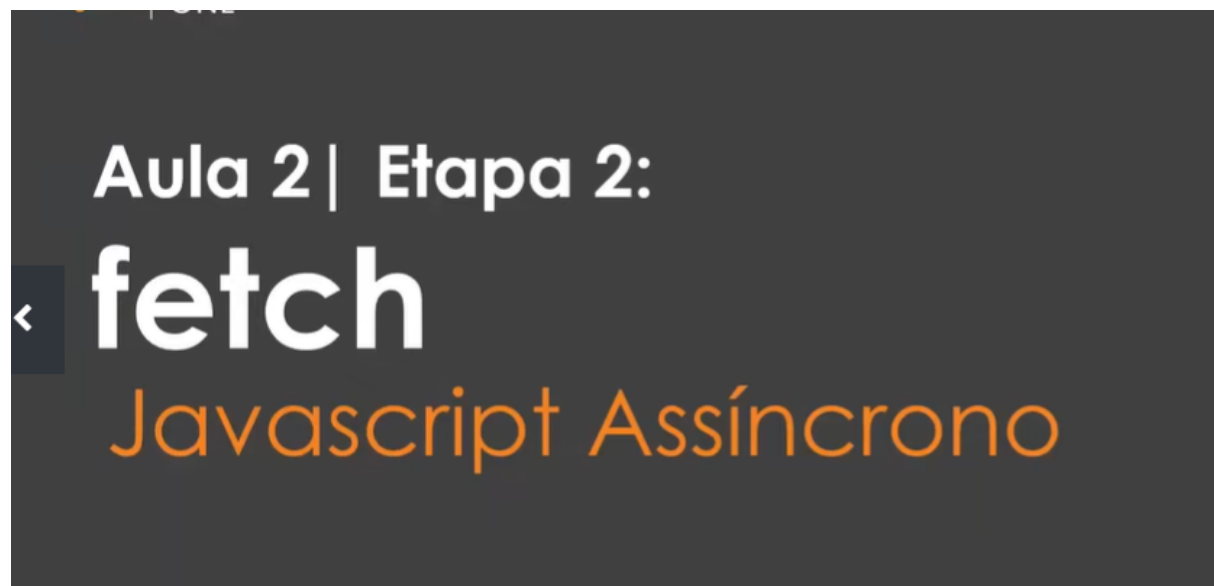
JSON: JavaScript Object Notation

```
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.

JSON: Anotação de Objeto em JavaScript

O resultado de APIs muitas vezes vem como JSON ou você precisar mandar como JSON também se você for fazer um post. É um formato de objeto mesmo com colchete chave e os valores.

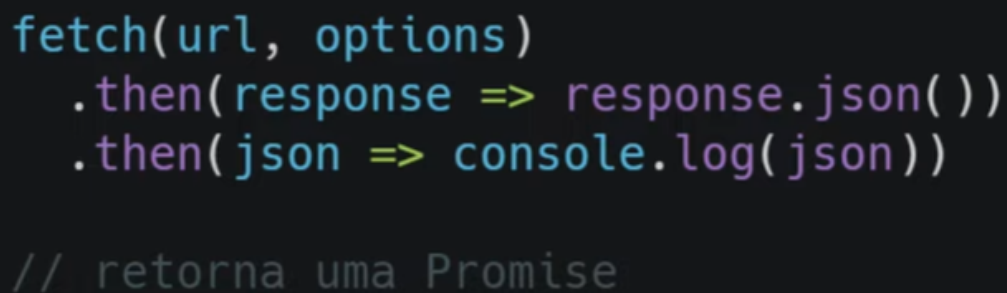


Ele é a grande estrela do show quando a gente fala sobre consumo de APIs.

Então você vai utilizar esse método com a seguinte sintaxe:

fetch

Consumindo APIs



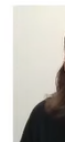
```
fetch(url, options)
  .then(response => response.json())
  .then(json => console.log(json))

// retorna uma Promise
```

A URL é padrão e podemos passar algumas opções, então podemos pesquisar na documentação quais são os tipos de opções que gente pode passar para uma requisição. O Fetch vai retornar uma Promise então você pode utilizar o encadeamento utilizando o then e você sempre tem que converter a sua resposta para JSON caso ela venha como uma string pra que você consiga utilizar dentro do JavaScript.

Ele vai retornar uma Promise então você precisa usar o await quando você for utilizá-lo.

fetch



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
```

Um exemplo usando o GET as opções que a gente usa pode ter um método, pode ter configurações de cache etc. No caso o POST você manda dados no banco de dados e aí você precisa converter de JSON pra uma string. Então você sempre tem que tratar o valor que você recebe de uma API ou quando você manda pra uma API sempre que você usa.

Muito importante: O Fetch retorna uma promise.

APIS e Fetch: A grosso modo eles são as regras que vão trazer dados do banco de dados para nós.