

AJAX Fetch

Fetch

O método **fetch** recebe como primeiro parâmetro a URL do **endpoint** que iremos executar a requisição assíncrona.

Como não é possível saber quando a solicitação estará concluída, o servidor retorna uma **promessa**.

```
{}
```

```
fetch("https://restcountries.eu/rest/v2/")
```

1º then

O primeiro bloco **then** irá receber a resposta da requisição **assíncrona** e modificar o seu formato para **JSON**.

```
{}
```

```
fetch("https://restcountries.eu/rest/v2/")  
  .then(function(response){  
    return response.json();  
  })
```

2º then

O segundo **then** recebe a resposta em formato **JSON** e pode manipulá-la como desejarmos.

```
{}  
fetch("https://restcountries.eu/rest/v2/")  
  .then(function(response){  
    return response.json();  
  })  
  .then(function(dataDecode){  
    console.log(dataDecode);  
  })
```

E se algo não funcionar?

Caso ocorra algum erro, o método **catch()** irá se encarregar de capturar e exibir o erro através do **console.log**.

```
{}  
fetch("https://restcountries.eu/rest/v2/")  
  .then(function(response){  
    return response.json();  
  })  
  .then(function(dataDecode){  
    console.log(dataDecode);  
  })  
  .catch(function(error){  
    console.log(error);  
  });
```

Fetch por POST

Como enviaremos dados por **POST**, devemos configurar um objeto literal com os dados necessários para que a API entenda nossa requisição. O método **fetch** possui um parâmetro opcional para essa configuração.

```
{  
  fetch(url, {  
    method: 'POST',  
    body: JSON.stringify(data),  
    headers: {  
      'Content-Type': 'application/json'  
    }  
  })  
}
```

DigitalHouse>
Coding School