

**Aluno :** Hector Luiz Ortiz

## **API Dolar para Real**

O código começa fazendo uma chamada à função `fetch` e passando a URL `"https://economia.awesomeapi.com.br/last/USD-BRL"` como argumento. Essa URL é a endpoint da API que retorna a taxa de câmbio atual entre o dólar americano (USD) e o real brasileiro (BRL).

Em seguida, usamos o método `then` para lidar com a resposta da API. O método `then` é usado para encadear promessas e executar código quando a promessa (no caso, a resposta da API) é resolvida.

Dentro do primeiro `then`, recebemos a resposta do servidor e chamamos o método `json()` na resposta para convertê-la em formato JSON. A função `json()` retorna uma nova promessa que representa o corpo da resposta convertido em JSON.

Em seguida, usamos outro `then` para lidar com a resposta JSON. Dentro dessa função, temos acesso aos dados retornados pela API, que estão armazenados na variável `data`.

Neste exemplo, o código extrai o valor da taxa de câmbio atual entre o dólar americano e o real brasileiro. A propriedade `data.USDBRL.high` contém o valor mais alto (`high`) da taxa de câmbio USD-BRL.

Em seguida, o código utiliza `console.log()` para imprimir a taxa de câmbio atual no console, interpolando o valor em uma string.

Caso ocorra algum erro durante a solicitação à API, o `catch` é acionado. O código dentro do `catch` é executado se houver qualquer erro na solicitação ou na conversão do JSON. Neste exemplo, o código imprime uma mensagem de erro específica no console.

**Fonte:** <https://docs.awesomeapi.com.br/api-de-moedas>

## API Piada

O código começa declarando uma constante `apiUrl` que armazena a URL da API do Chuck Norris Jokes. Essa API retorna uma piada aleatória sobre Chuck Norris.

Em seguida, usamos a função `fetch` para fazer uma solicitação HTTP GET para a URL da API (`apiUrl`). A função `fetch` retorna uma promessa que representa a resposta da API.

Usamos o método `then` para lidar com a resposta da API. O método `then` é usado para encadear promessas e executar código quando a promessa (no caso, a resposta da API) é resolvida.

Dentro do primeiro `then`, recebemos a resposta do servidor e chamamos o método `json()` na resposta para convertê-la em formato JSON. A função `json()` retorna uma nova promessa que representa o corpo da resposta convertido em JSON.

Em seguida, usamos outro `then` para lidar com a resposta JSON. Dentro dessa função, temos acesso aos dados retornados pela API, que estão armazenados na variável `data`.

Neste exemplo, o código extrai a piada do objeto `data`. A propriedade `data.value` contém a piada sobre Chuck Norris.

Em seguida, o código utiliza `console.log()` para imprimir a piada no console, interpolando o valor em uma string.

Caso ocorra algum erro durante a solicitação à API, o `catch` é acionado. O código dentro do `catch` é executado se houver qualquer erro na solicitação ou na conversão do JSON. Neste exemplo, o código imprime uma mensagem de erro específica no console.

**Fonte :** <https://api.chucknorris.io>

## API Conselho

O código começa chamando a função `fetch` e passando a URL da API `https://api.advicejslip.com/advice` como argumento. Isso inicia uma solicitação HTTP GET para a API para obter um conselho.

Em seguida, usamos o método `then` para lidar com a resposta da API. O método `then` é usado para encadear promessas e executar código quando a promessa (no caso, a resposta da API) é resolvida.

Dentro do primeiro `then`, recebemos a resposta do servidor e chamamos o método `json()` na resposta para convertê-la em formato JSON. A função `json()` retorna uma nova promessa que representa o corpo da resposta convertido em JSON.

Em seguida, usamos outro `then` para lidar com a resposta JSON. Dentro dessa função, temos acesso aos dados retornados pela API, que estão armazenados na variável `data`.

Neste exemplo, o código simplesmente imprime o conselho no console usando `console.log()`. A propriedade `data.slip.advice` contém o conselho retornado pela API.

Caso ocorra algum erro durante a solicitação à API, o `catch` é acionado. O código dentro do `catch` é executado se houver qualquer erro na solicitação ou na conversão do JSON. Neste exemplo, o código simplesmente imprime uma mensagem de erro e o erro específico no console.

**Fonte :** <https://api.advicejslip.com/#endpoint-random>