**AXIOS**

É um HTTP Cliente que pode s para node.js e browser, ou seja, pode ser executado tanto no browser(usa o XMLHttpRequest) e no node.js(usa o módulo http)

**Instalação**

* **Basta importar usando CDN:**

**Fazendo um requisição**

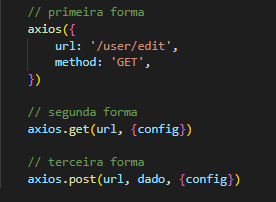
Pode ser feita de três formas:

* **Axios({config}):** passando as configurações
* **Axios(url, {config}):** passando a URL e as configurações
* **Axios.method(url, dado, {config}):** onde method é qualquer método HTTP e dado caso o método envie dados no body

Opções disponíveis:

* **url:** url requisitada
* **method:** método hTTP. Padrão é ‘GET’
* **baseURL:** origin da URL(método, domínio e porta). Se a proprieedade ‘url’ for absoluta, esta propriedade será ignorada.
* **transformRequest [function(data, headers)]:** um array de funções para transformar os dados antes de enviá-los para o servidor.
  + Apenas aplicável para métodos POST, DELETE, HEADERS e PATCH
  + A última função do array deve retonar uma **String, Buffer, ArrayBuffer, DataForm ou Stream**
* **transformResponse[function (data, headers)]:** transforma os dados antes de enviar para o then ou catch, da mesma forma que a propriedade transformRequest
* **headers:** objeto definindo ehaders customizados para serem enviados
* **params:** parâmetros para serem enviados na URL
  + Pode ser um objeto ou URLSearchParams
* **Data:** dado sendo enviado no corpo da requisição
  + Apenas para os métodos **POST, DELETE, PUT, PATH**
  + Pode ser um objeto, string, ArrayBuffer, Buffer, ArrayBufferView, URLSearchParams, FormData, File ou Blob
* **responseType:** define o tipo de dado que o servidor
  + Tipos possíveis: 'arraybuffer', 'document', 'json', 'text', 'stream' browser only: 'blob'. Padrão é ‘json’

Veja todas as opções em [Axios Request Config](https://axios-http.com/docs/req_config)

Usando o axios especificando o método:

* **axios.get(url, {config})**
* **axios.delete(url, {config})**
* **axios.head(url, {config})**
* **axios.options(url, {config})**
* **axios.post(url, data, {config})**
* **axios.put(url, data, {config})**
* **axios.patch(url, data, {config})**

**Ex.:** usando os métodos GET e POST

Depois que fizer uma requisição, axios retornará uma **Promise**, onde:

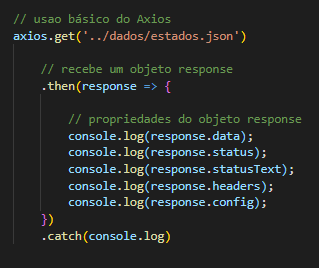
* **resolve** retorna um **objeto response** (pode se acessada pelo método **then**)
* **reject** retorna um **erro** (pode se acessada pelo método **catch)**

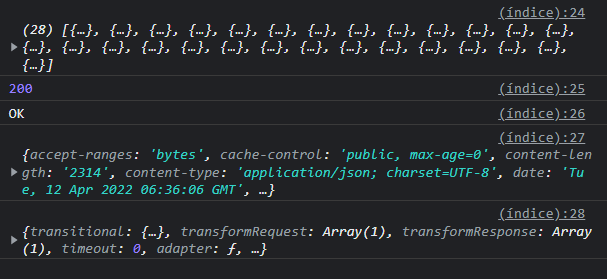
**Objeto Response**

Será um objeto que representa a resposta enviada pelo server. Propriedades:

* **response.data:** dado enviado pelo server
* **response.status:** código de status HTTP
* **response.statusText:** descrição do código de status
* **response.headers:** objeto contendo todos os ehaders de resposta
* **response.request:** objeto XMLHTTPRequest
* **response.config**: objeto contendo todas as configurações feitas pelo Axios no request

**Ex.:** todas as propriedades do objeto response



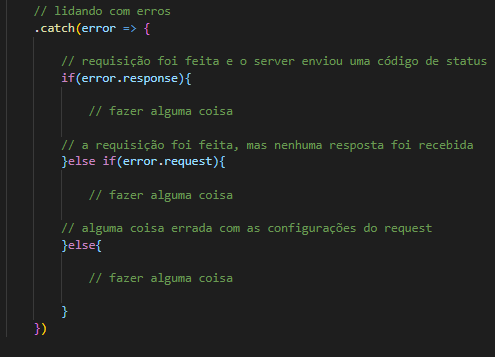


**Lidando com Erros**

Reject da promise recebe um objeto erro. Propriedades:

* **error.response:** caso o server envie uma resposta com código de status de erro
* **erro.request:** o request foi enviado, mas não recebeu nenhuma resposta do server
* **error:** caso haja algo errado nas configurações do request

**Ex.1:** capturando e tratando erros



**Ex.2:**