**Axios JS**

**O'Que é Axios?**

Uma biblioteca JavaScript para requisições HTTP;

Axios é Promise based, ou seja, retorna promessas de suas funções;

Torna muito mais simples o trabalho com APIs e requisições assíncronas;

Muito utilizado nas empresas;

Apesar disso, perdeu muita notoriedade para o recurso de fetch do JS;

**Instalando o Axios**

Para instalar o Axios basta **copiar um link de script externo** para o nosso projeto;

O link da documentação é: <https://axios-http.com/>

Em projetos que utilizamos bibliotecas e frameworks, utilizamos o npm para instalar o Axios;



**Nosso primeiro request**

Para fazer uma requisição podemos utilizar o método **get,** isso vai nos trazer dados de algum local;

É recomendado utilizar um **try catch** para identificar possíveis erros;

Como o Axios é baseado em promises, podemos utilizar as **async functions**;



**Exibindo os dados na tela**

Após um request é comum **exibir os dados na tela;**

Podemos fazer isso juntando a resposta da chamada com os nossos conhecimentos em **DOM;**

Criar elementos baseado no que veio na requisição;

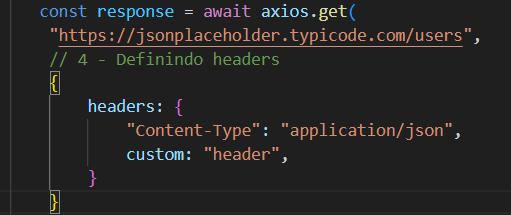


**Configurando os headers**

Os **headers** são configurados no momento da requisição;

Podemos adicionar **parâmetros adicionais;**

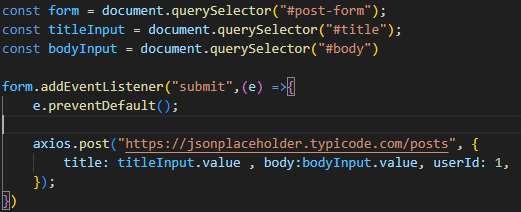
Por exemplo: determinar o tipo de dado que queremos;



**Requisição de POST**

Para enviar dados vamos utilizar o método **post;**

É necessário configurar a **propriedade body** com os dados a serem enviados;



**Global Instance do Axios**

Podemos alterar diretamente as **configurações do Axios**;

Isso nos gera uma facilidade de trabalhar com os **mesmo parâmetros em todas as requisições**;

Ou seja, se configuramos os headers na Global, não é necessário configurar nas requisições;





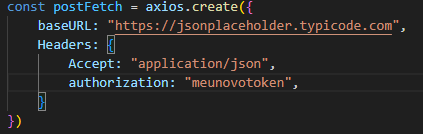
**Custom Instance do Axios**

A **Custom Instance** é semelhante a Global Instance;

Porém aqui temos outras propriedades que são possíveis de configurar, como a **baseUrl**;

Esta estratégia deve ser utilizada para personalização do nosso projeto;

Obs: não é recomendado utilizar as duas instances juntas (manutenção, configuração em dois locais…);



**Interceptors**

Interceptors são como **middlewares**;

Ou seja, podemos **interceptar a requisição e a resposta**;

Inserindo algum código entre estas duas ações;