



Axios

DEVinHouse

Parcerias para desenvolver a sua carreira

SENAI

<LAB365>

AGENDA DA SEMANA

- **Axios**
- Deploy e firebase hosting

Axios é um cliente HTTP baseado em Promises para fazer requisições. Pode ser utilizado tanto no navegador quanto no Node.js ou qualquer serviço de API



PROMISES

Promises, definem uma ação que vai ser executada no futuro, ou seja, ela pode ser resolvida (com sucesso) ou rejeitada (com erro).

```
axios.get('https://625f0d2e3b039517f10057c8.mockapi.io/api/v1/user')  
  .then((response) => {  
    //... SUCCESS  
  })  
  .catch((err) => {  
    //... ERROR  
  })
```

Métodos de Requisição HTTP

O protocolo HTTP define um conjunto de métodos de requisição responsáveis por indicar a ação a ser executada para um dado recurso. Dentre os métodos, os mais comuns são:

GET: O método GET solicita a representação de um recurso específico. Requisições utilizando o método GET devem retornar apenas dados.

POST: O método POST é utilizado para submeter uma entidade a um recurso específico, frequentemente causando uma mudança no estado do recurso ou efeitos colaterais no servidor.

PUT: O método PUT substitui todas as atuais representações do recurso de destino pela carga de dados da requisição.

PATCH: O método PATCH é utilizado para aplicar modificações parciais em um recurso.

DELETE: O método DELETE remove um recurso específico.

Instalação do Axios

Para utilizarmos o Axios no Vue, faça a instalação da dependência pelo comando:

```
npm install axios
```

Após a instalação, esta biblioteca ficará disponível para uso na aplicação ao importá-lo com:

```
import axios from 'axios';
```

AXIOS - GET

Conforme vimos anteriormente, podemos fazer uma busca de informação em uma API utilizando o método GET. No Axios podemos fazer essa busca da seguinte maneira:

```
async getUsers() {  
  axios.get('https://625f0d2e3b039517f10057c8.mockapi.io/api/v1/user')  
    .then((response) => {  
      //... SUCCESS  
    })  
    .catch((err) => {  
      //... ERROR  
    })  
}
```

AXIOS - GET

Caso necessitamos passar algum parâmetro, no método GET faremos isso via *path param*, onde o parâmetro vai após a "/" da url, ou via *query param*, iniciando a declaração pelo "?" e separando-os por "&":

```
//PATH PARAM
axios.get(`https://625f0d2e3b039517f10057c8.mockapi.io/api/v1/user/${id}`)
```

```
//QUERY PARAM
axios.get('https://625f0d2e3b039517f10057c8.mockapi.io/api/v1/user', {
  params: {
    id: 1
  }
})
```


AXIOS - POST

Também podemos submeter um objeto via POST seguindo o modelo, enviando o objeto a ser inserido no body da requisição:

```
axios.post(`https://625f0d2e3b039517f10057c8.mockapi.io/api/v1/user`, {  
  nome: 'Nome',  
  cpf: '000000000000',  
  cargo: 'Cargo'  
})  
  
  .then((response) => {  
    //... SUCCESS  
  })  
  .catch((err) => {  
    //... ERROR  
  })
```

AXIOS - PATCH e PUT

Os métodos PATCH e PUT são semelhantes, o objetivo é atualizar um objeto com a diferença que pelo protocolo o PATCH é destinado para alterações parciais do objeto enquanto o PUT para substituição completa do objeto em questão. Em uma simples operação de atualização de registro, normalmente é utilizado o PATCH.

```
axios.put(`https://625f0d2e3b039517f10057c8.mockapi.io/api/v1/user/${3}`, {
  nome: 'Nome Alterado',
  cpf: '000000000000',
  cargo: 'Cargo 3'
})
  .then((response) => {
    //... SUCCESS
  })
  .catch((err) => {
    //... ERROR
  })
```

AXIOS - DELETE

Para exclusão de um registro, normalmente é utilizado o método DELETE, onde passamos via path param ou query param (Assim como no GET), a identificação do registro a ser excluído. Neste método não é possível utilizar o body.

```
axios.delete(`https://625f0d2e3b039517f10057c8.mockapi.io/api/v1/user/${3}`)  
  .then((response) => {  
    //... SUCCESS  
  })  
  .catch((err) => {  
    //... ERROR  
  })
```



DEVinHouse

Parcerias para desenvolver a sua carreira

OBRIGADO!



<LAB365>