GOMYCODE®

What's Axios?

Axios is a promise based HTTP client for the browser and Node.js.

By using Axios it's easy to send asynchronous HTTP request to REST endpoints and perform CRUD operations. The Axios library can be used in your plain JavaScript application or can be used together with more advanced frameworks like React.js

There are some difference between axios the fetch web API:

Syntax: the http request using axios is more simple than using fetch

Support: the fetch web API is not supported by all the browser, on the other hand axios is supported even by the old browser version (like IE)

Response timeout: the timeout set up is much easier using axois than using the fetch API.

Install Axios

To install axios we simply run the following command:

```
npm i axios
// or
npm install axios
```

Use Axios

Axios syntax is mostly like the fetch API.

To Perform a get request:

```
// Make a request for a user with a given ID
axios.get('/user?ID=12345')
  .then(function (response) {
    // handle success
    console.log(response);
  })
  .catch(function (error) {
    // handle error
```

```
    console.log(error);
  })
```

Posting Data using Axios

The post request syntax is the following:

```
axios.post(url, data,config)
```

- Url: the url of the api

- Data: the data to be sent (optional field)

- Config: the header configuration (optional field)

```
axios
  .post("/user", {
    firstName: "Fred",
    lastName: "Flintstone"
  })
  .then(function(response) {
    console.log(response);
  })
  .catch(function(error) {
    console.log(error);
  });
```

Chaining request

Axios comes with build in feature which gives us the ability to send multiple requests.

```
const requestOne = axios.get("https://api.storyblok.com/v1/cdn/stories/health?version
const requestTwo = axios.get(https://api.storyblok.com/v1/cdn/datasources/?token=wANp
const requestThree = axios.get("https://api.storyblok.com/v1/cdn/stories/vue?version=

axios
  .all([requestOne, requestTwo, requestThree])
  .then(axios.spread((firstResponse, secondResponse, thirdResponse) => {
    console.log(firstResponse.data,secondResponse.data, thirdResponse.data);
  })
```

```
)
  .catch(errors => {
    console.error(errors);
  });
```