

TO
THE
NEW™



Axios

Getting Data from Server

Agenda

Introduction to Axios

API Methods

Error Handling

Interceptors

Setting up Default Config

Introduction

Axios is a promise-based HTTP client that works both in the browser and in a node.js environment

Installation:

Npm install axios --save

Bower install axios

Running Axios

Node

```
var axios = require('axios')  
axios.get('https://api.github.com/users/codeheaven-io')
```

Browser

```
<script src="./bower_components/axios/dist/axios.js"></script>  
<script>  
  axios.get('https://api.github.com/users/codeheaven-io');  
</script>
```

Using Axios

```
import <any alias name> from 'axios';  
<alias name>.get(`http://www.reddit.com/r/${this.props.subreddit}.json`)  
  .then(res => {  
    const posts = res.data.data.children.map(obj => obj.data);  
    this.setState({ posts });  
  });  
`http://www.reddit.com/r/\${this.props.subreddit}.json` : template string part of ES6  
syntax.
```

Axios API

```
// Performing a GET request
axios.get('https://api.github.com/users/' + username)
  .then(function(response){
    console.log(response.data); // ex.: { user: 'Your User'}
    console.log(response.status); // ex.: 200
  });
```

```
// Performing a POST request
axios.post('/save', { firstName: 'Marlon', lastName: 'Bernardes' })
  .then(function(response){
    console.log('saved successfully')
  });
```

Axios API

To execute multiple requests in parallel, simply provide an array argument to **axios.all**. When all requests are complete, you'll receive an array containing the response objects in the same order they were sent. Alternatively you can use **axios.spread** to spread the array into multiple arguments. Spread is preferred since dealing with array indexes could be misleading

Axios All

```
axios.all([
  axios.get(<URL>);
  axios.get(<URL>)
])
.then(axios.spread(function (res1, res2) {
  console.log('User', res1.data, res2.data);
})));
```

Custom Headers

```
var config = {  
  headers: {'X-My-Custom-Header': 'Header-Value'}  
};  
axios.get(<URL>, config);  
axios.post(<URL>, <DATA>, config);
```

Error Handling

```
axios.get(<URL>)  
  .catch(function(error){  
    if(error.response) {  
      Console.log(error.response)  
    }  
    Else{  
      console.log(error.message)  
    }  
  })
```

Interceptors

You can intercept **requests** or **responses** before they are handled by then or catch.

```
Axios.interceptors.request.use(function(config){  
  //TODO: before request is sent  
  return config  
})  
Axios.interceptors.response.use(function(response){  
  //TODO: modify response Data  
  return response;  
})
```

Add or Remove Interceptors

```
Var interceptor = axios.interceptors.request.use(function(){...})  
axios.interceptos.request.eject(interceptor)
```

Setting Up Default Config

- `axios.defaults.baseURL = 'https://api.example.com';`
- `axios.defaults.headers.common['Authorization'] = AUTH_TOKEN;`
- `axios.defaults.headers.post['Content-Type'] = 'application/x-www-form-urlencoded';`

QUESTIONS????