

REACTJS – MODULE 7

INTRODUCTION TO AXIOS (HTTP CLIENT)



AGENDA

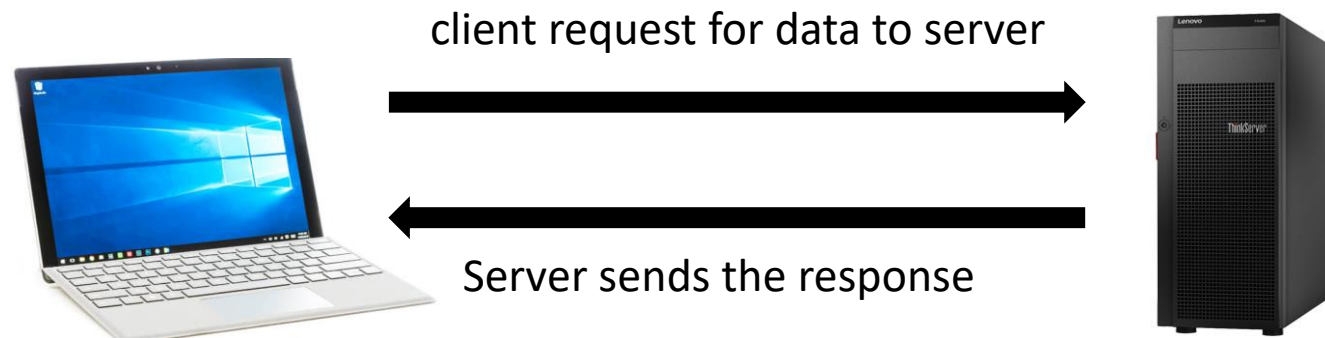
- What is http client
- Introduction to axios
- Demo
- Activity



Introduction to HTTP

HTTP is a connection oriented data transfer protocol, using this protocol data will be transmitted over the network

Whenever the client require the data it will make a request to server and server will provide the data



Introduction to HTTP Client - AXIOS

The request is made by browser or Http clients to server using some of the methods like get, post, put etc

The most commonly used methods are get and post. Get method is considered as safe because it is used only to retrieve the data

AXIOS is one of the Http client which uses different request methods to make a request to server

In this module we will see how axios can be used to make http request.

Introduction to HTTP Client – AXIOS (cont...)

AXIOS is a http client and it is based on javascript promise concept

Promises are basically used to manage asynchronous requests, promises will be either resolved or rejected

The first step is to install axios using npm

`npm install axios --save-dev`

```
C:\reactdemos\axiosexample>npm install axios --save-dev
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.1.2 (node_modules\react-scripts\node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.1.2: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.1.3 (node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.1.3: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})
+ axios@0.17.1
added 230 packages and updated 1 package in 76.616s
```

Demo - AXIOS

Create new component(HttpDemo.js)

```
import React, { Component } from 'react';
import axios from 'axios';
class HttpDemo extends Component {
  constructor() {
    super();
    axios.get(" https://jsonplaceholder.typicode.com/posts")
      .then((response) => {
        console.log("Response=" + JSON.stringify(response.data));
      })
      .catch((error) => {
        console.log('There is error')
      })
  }
  render() {
    return (
      <div>
        <h3>AXIOS HTTP Client</h3>
      </div>
    );
  }
}
export default HttpDemo;
```

Demo – AXIOS – Explanation

```
import React, { Component } from 'react';
import axios from 'axios';
class HttpDemo extends Component {
  constructor() {
    super();
    axios.get(" https://jsonplaceholder.typicode.com/posts")
      .then((response) => {
        console.log("Response=" + JSON.stringify(response.data));
      })
      .catch((error) => {
        console.log('There is error')
      })
  }
  render() {
    return (
      <div>
        <h3>AXIOS HTTP Client</h3>
      </div>
    );
  }
}
export default HttpDemo;
```

As axios is already installed, it has to be imported in the code to utilize its functionalities

Get method is invoked using axios(Http client). Get method require url . In this example we are using JSONPlaceholder which is fake REST API for testing and prototyping only.

This API holds data in json format which will be requested by axios

Demo – AXIOS – Explanation(Cont...)

```
import React, { Component } from 'react';
import axios from 'axios';
class HttpDemo extends Component {
  constructor() {
    super();
    axios.get(" https://jsonplaceholder.typicode.com/posts")
      .then((response) => {
        console.log("Response=" + JSON.stringify(response.data));
      })
      .catch((error) => {
        console.log('There is error')
      })
  }
  render() {
    return (
      <div>
        <h3>AXIOS HTTP Client</h3>
      </div>
    );
  }
}
export default HttpDemo;
```

As mentioned Axios is promise based Http client, which will be resolved on successful completion of the request, otherwise the request will be rejected

“then” section will be executed on successful completion of the request. The response will be properly formatted in json and displayed on the console

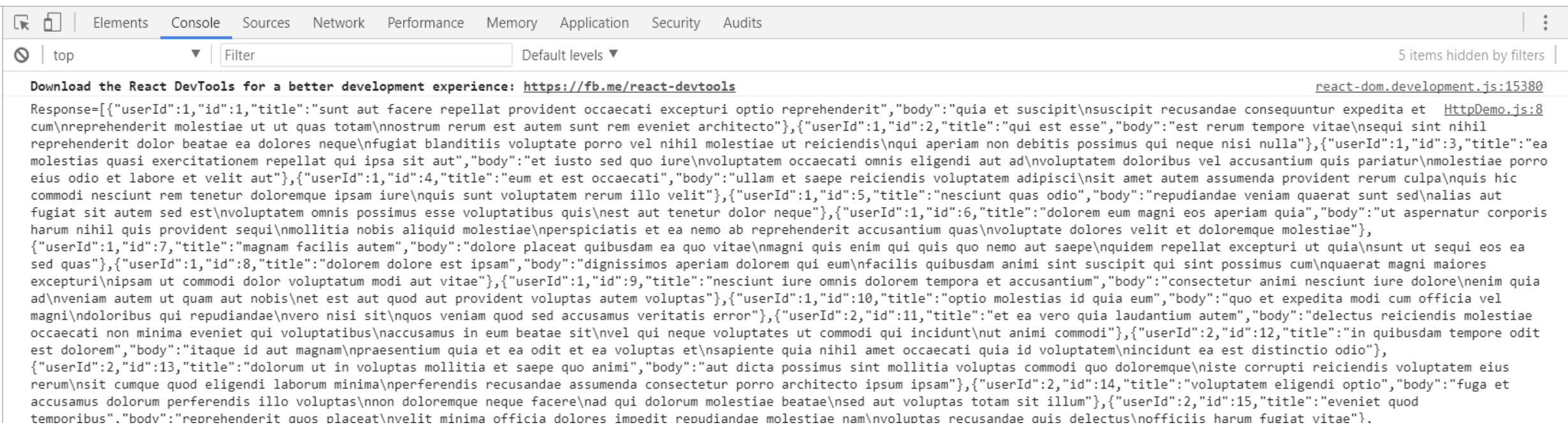
Demo – AXIOS – Explanation(Cont...)

```
import React, { Component } from 'react';
import axios from 'axios';
class HttpDemo extends Component {
  constructor() {
    super();
    axios.get(" https://jsonplaceholder.typicode.com/posts")
      .then((response) => {
        console.log("Response=" + JSON.stringify(response.data));
      })
      .catch((error) => {
        console.log('There is error')
      })
  }
  render() {
    return (
      <div>
        <h3>AXIOS HTTP Client</h3>
      </div>
    );
  }
}
export default HttpDemo;
```

“catch” section will be executed if the request is rejected and error message is displayed on the console

Demo – AXIOS – Explanation(Cont...)

Output on browser
console



The screenshot shows the Chrome DevTools Console with the 'Console' tab selected. At the top, there's a toolbar with icons for 'top', 'Filter', and 'Default levels'. A message indicates '5 items hidden by filters'. Below this, a banner promotes downloading React DevTools. The main console area displays a large JSON array response from 'HttpDemo.js:8'. The response contains 15 objects, each with 'userId', 'id', 'title', and 'body' fields. The 'body' fields contain various Latin text snippets.

```
Response=[{"userId":1,"id":1,"title":"sunt aut facere repellat provident occaecati excepturi optio reprehenderit","body":"quia et suscipit\nsuscipit recusandae consequuntur expedita et cum\nreprehenderit molestiae ut ut quas totam\nnostrum rerum est autem sunt rem eveniet architecto"}, {"userId":1,"id":2,"title":"qui est esse","body":"est rerum tempore vitae\nsequi sint nihil reprehenderit dolor beatae ea dolores neque\nfugiat blanditiis voluptate porro vel nihil molestiae ut reiciendis\nqui aperiam non debitis possimus qui neque nisi nulla"}, {"userId":1,"id":3,"title":"ea molestias quasi exercitationem repellat qui ipsa sit aut","body":"et iusto sed quo iure\nvoluptatem occaecati omnis eligendi aut ad\nvoluptatem doloribus vel accusantium quis pariatur\nmolestiae porro eius odio et labore et velit aut"}, {"userId":1,"id":4,"title":"eum et est occaecati","body":"ullam et saepe reiciendis voluptatem adipisci\nsit amet autem assumenda provident rerum culpa\nquis hic commodi nesciunt rem tenetur doloremque ipsam iure\nquis sunt voluptatem rerum illo velit"}, {"userId":1,"id":5,"title":"nesciunt quas odio","body":"repudiandae veniam quaerat sunt sed\nnalias aut fugiat sit autem sed est\nvoluptatem omnis possimus esse voluptatibus quis\nnest aut tenetur dolor neque"}, {"userId":1,"id":6,"title":"dolorem eum magni eos aperiam quia","body":"ut aspernatur corporis harum nihil quis provident sequi\nmollitia nobis aliquid molestiae\nperspiciatis et ea nemo ab reprehenderit accusantium quas\nvoluptate dolores velit et doloremque molestiae"}, {"userId":1,"id":7,"title":"magnam facilis autem","body":"dolore placeat quibusdam ea quo vitae\nmagni quis enim qui quis quo nemo aut saepe\nnquidem repellat excepturi ut quia\nsunt ut sequi eos ea sed quas"}, {"userId":1,"id":8,"title":"dolorem dolore est ipsam","body":"dignissimos aperiam dolorem qui eum\nfacilis quibusdam animi sint suscipit qui sint possimus cum\nquaerat magni maiores excepturi\nipsam ut commodi dolor voluptatum modi aut vitae"}, {"userId":1,"id":9,"title":"nesciunt iure omnis dolorem tempora et accusantium","body":"consectetur animi nesciunt iure dolore\nenim quia ad\nveniam autem ut quam aut nobis\net est aut quod aut provident voluptas autem voluptas"}, {"userId":1,"id":10,"title":"optio molestias id quia eum","body":"quo et expedita modi cum officia vel magni\ndoloribus qui repudiandae\nvero nisi sit\nquos veniam quod sed accusamus veritatis error"}, {"userId":2,"id":11,"title":"et ea vero quia laudantium autem","body":"delectus reiciendis molestiae occaecati non minima eveniet qui voluptatibus\naccusamus in eum beatae sit\nvel qui neque voluptates ut commodi qui incidunt\nnut animi commodi"}, {"userId":2,"id":12,"title":"in quibusdam tempore odit est dolorem","body":"itaque id aut magnam\npraesentium quia et ea odit et ea voluptas et\nsapiente quia nihil amet occaecati quia id voluptatem\nincidunt ea est distinctio odio"}, {"userId":2,"id":13,"title":"dolorum ut in voluptas mollitia et saepe quo animi","body":"aut dicta possimus sint mollitia voluptas commodi quo doloremque\nniste corrupti reiciendis voluptatem eius rerum\nsit cumque quod eligendi laborum minima\nperferendis recusandae assumenda consectetur porro architecto ipsum ipsam"}, {"userId":2,"id":14,"title":"voluptatem eligendi optio","body":"fuga et accusamus dolorum perferendis illo voluptas\nnon doloremque neque facere\nad qui dolorum molestiae beatae\nsed aut voluptas totam sit illum"}, {"userId":2,"id":15,"title":"eveniet quod temporibus","body":"reprehenderit quos placeat\nvelit minima officia dolores impedit repudiandae molestiae nam\nvoluptas recusandae quis delectus\nnofficiis harum fugiat vitae"}]
```

Passing parameter in Get request

While requesting data from REST API , it is also possible that client may need only specific data instead of complete data

This can be achieved by passing parameter in the get request

Parameter can be passed either directly in the url or by defining it as “params” within get request

Demo - Passing parameter in Get request

Modify existing component(HttpDemo.js)

```
import React, { Component } from 'react';
import axios from 'axios';
class HttpDemo extends Component {
  constructor() {
    super();
    axios.get(" https://jsonplaceholder.typicode.com/posts", {
      params: {
        userId: 3
      }
    }).then((response) => {
      console.log("Response=" + JSON.stringify(response.data));
    }).catch((error) => {
      console.log('There is error')
    })
  }
  render() {
    return (
      <div>
        <h3>AXIOS HTTP Client</h3>
      </div>
    );
  }
}
export default HttpDemo;
```

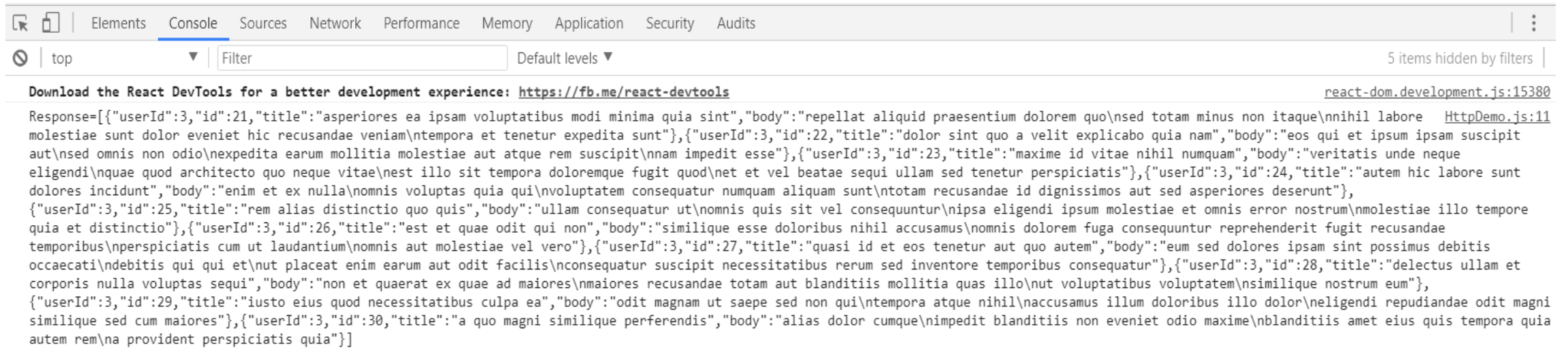
Demo - Passing parameter in Get request - Explanation

```
import React, { Component } from 'react';
import axios from 'axios';
class HttpDemo extends Component {
  constructor() {
    super();
    axios.get("https://jsonplaceholder.typicode.com/posts",{
      params:{
        userId:3
      }
    }).then((response) => {
      console.log("Response=" + JSON.stringify(response.data));
    }).catch((error) => {
      console.log('There is error')
    })
  }
  render() {
    return (
      <div>
        <h3>AXIOS HTTP Client</h3>
      </div>
    );
  }
}
export default HttpDemo;
```

Only the data specific to
userId=3 is displayed

Demo - Passing parameter in Get request

Output on browser
console



The screenshot shows the Chrome DevTools Console with the 'Console' tab selected. At the top, there's a toolbar with icons for back, forward, and search, followed by tabs for Elements, Console, Sources, Network, Performance, Memory, Application, Security, and Audits. Below the tabs, there's a filter input set to 'top' and a 'Default levels' dropdown. A message at the top of the console says 'Download the React DevTools for a better development experience: https://fb.me/react-devtools' with a link to 'react-dom.development.js:15380'. The main content of the console is a large JSON array representing a REST client response. The array contains 10 objects, each with 'userId', 'id', 'title', and 'body' properties. The 'body' property for each object is a long, repetitive Latin sentence. The response is truncated with '...'. The source of the response is indicated as 'HttpDemo.js:11'.

```
Response=[{"userId":3,"id":21,"title":"asperiores ea ipsam voluptatibus modi minima quia sint","body":"repellat aliquid praesentium dolorem quo\nsed totam minus non itaque\nnihil labore HttpDemo.js:11 molestiae sunt dolor eveniet hic recusandae veniam\ntempora et tenetur expedita sunt"}, {"userId":3,"id":22,"title":"dolor sint quo a velit explicabo quia nam","body":"eos qui et ipsum ipsam suscipit aut\nsed omnis non odio\nexpedita earum mollitia molestiae aut atque rem suscipit\nnam impedit esse"}, {"userId":3,"id":23,"title":"maxime id vitae nihil numquam","body":"veritatis unde neque eligendi\nquae quod architecto quo neque vitae\nest illo sit tempora doloremque fugit quod\net et vel beatae sequi ullam sed tenetur perspiciatis"}, {"userId":3,"id":24,"title":"autem hic labore sunt dolores incidunt","body":"enim et ex nulla\nomnis voluptas quia qui\nvoluptatem consequatur numquam aliquam sunt\ntotam recusandae id dignissimos aut sed asperiores deserunt"}, {"userId":3,"id":25,"title":"rem alias distinctio quo quis","body":"ullam consequatur ut\nomnis quis sit vel consequuntur\nipsa eligendi ipsum molestiae et omnis error nostrum\nmolestiae illo tempore quia et distinctio"}, {"userId":3,"id":26,"title":"est et quae odit qui non","body":"similique esse doloribus nihil accusamus\nomnis dolorem fuga consequuntur reprehenderit fugit recusandae temporibus\nperspiciatis cum ut laudantium\nomnis aut molestiae vel vero"}, {"userId":3,"id":27,"title":"quasi id et eos tenetur aut quo autem","body":"eum sed dolores ipsam sint possimus debitis occaecati\ndebitis qui qui et\nnut placeat enim earum aut odit facilis\nconsequatur suscipit necessitatibus rerum sed inventore temporibus consequatur"}, {"userId":3,"id":28,"title":"delectus ullam et corporis nulla voluptas sequi","body":"non et quaerat ex quae ad maiores\nmaiores recusandae totam aut blanditiis mollitia quas illo\nnut voluptatibus voluptatem\nsimilique nostrum eum"}, {"userId":3,"id":29,"title":"iusto eius quod necessitatibus culpa ea","body":"odit magnam ut saepe sed non qui\ntempora atque nihil\naccusamus illum doloribus illo dolor\nneligendi repudiandae odit magni similique sed cum maiores"}, {"userId":3,"id":30,"title":"a quo magni similique perferendis","body":"alias dolor cumque\nimpedit blanditiis non eveniet odio maxime\nblanditiis amet eius quis tempora quia autem rem\nna provident perspiciatis quia"}]
```

Activity

1. Create new project in firebase
2. Create titles.json file in the localdrive
3. Import titles.json file into firebase project
4. Create a component which will fetch data from titles.json and display the details in console.

MODULE SUMMARY

- Basics of Http
- Http methods and when to use which method
- How axios helps to communicate http



THANK YOU

