WEB TECHNOLOGIES

AXIOS

Prof. Pavan A C

Department of Computer Science and Engineering

Acknowledgement:

Teaching Assistants(Harini B and Chandana MS)

AXIOS

Axios

- Axios is a popular open-source JavaScript library used to make HTTP requests from web browsers or Node.js environments.
- It simplifies the process of sending asynchronous HTTP requests to REST endpoints and handling responses.

- Features :
  - Promise-based API.
  - Works in both NodeJS and browsers.
  - Automatically transforms JSON data.
  - Supports request and response interceptors.
  - Allows easy handling of timeouts and cancellation of requests.
  - Supports making GET, POST, PUT, DELETE, and other HTTP requests.

AXIOS

Installation

- **Installing via npm**
  - To install Axios in a Node.js project, run:

    ```
    npm install axios
    ```

  - After installing axios can be imported by including the below code.

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

- **Including in HTML**
  - For use in a browser environment, include Axios via a CDN link in your HTML file:

```
<script src="https://cdn.jsdelivr.net/npm/axios/dist/axios.min.js"> </script>
```

AXIOS

Usage in GET

Making a GET Request

```
const axios = require('axios');
// Making a GET request
axios.get('https://exampleget.com/data')
  .then(response => {
    console.log(response.data);
  })
  .catch(error => {
    console.error('Error fetching data:', error);
  });
```

Here if there is a correct response the content is logged or else error is logged.

AXIOS

Usage in POST

Sending POST Requests

```
        axios.post('https://examplepost.com/data', {
         name: Name,
         age: 20
        })
        .then(response => {
         console.log(response.data);
        })
        .catch(error => {
         console.error(error);
});
```

Here if there is a correct response the content is logged or else error is logged.

Errors can be handled easily with the catch block. Any number of logging statements or actions can be taken

AXIOS

Apart from the basic GET and POST requests, Axios also supports a variety of advanced request options and features, making it a powerful and flexible HTTP client.

1. **Common Request Options**

- Setting Headers: Customize HTTP headers for your requests, such as Authorization tokens, content types, or custom headers.
- Query Parameters (params): Easily append query strings to your URLs by providing an object of key-value pairs, making your GET requests more dynamic.
- Timeouts: Define a maximum time to wait for a server response, so your app doesn’t hang if a request takes too long.
- Response Types: Specify the expected format of the response data like JSON, text, blob, or stream, allowing appropriate handling.

AXIOS

```
// EXAMPLE
axios.get('/user', {
  params: { id: 123 },
  headers: { 'Authorization': 'Bearer your_token' },
  timeout: 5000,
  responseType: 'json'
})
.then(response => {
  console.log(response.data);
})
.catch(error => {
  console.error('Request failed:', error);
});
```

AXIOS

2. Handling Responses

Axios makes it straightforward to access parts of the HTTP response:

- response.data contains the response payload (body).
- response.status is the HTTP status code (e.g., 200 OK, 404 Not Found).
- response.headers holds the response headers.

3. Global Configuration

You can configure Axios to have default settings that apply to all requests made via an instance like timeout or base URL. This reduces repetition

Practical Best Practices and Real-world Use-cases for Axios

- Always handle errors: Use .catch() or try-catch with async/await to gracefully handle network or server errors and show user-friendly messages.
- Validate response data: Verify the structure and type of data received before using it
- Use async/await syntax: Use async functions and await instead of chaining .then().

Real-World Use-cases:

- RESTful API calls from popular frontend frameworks such as React, Vue, or Angular to fetch and send data.
- Server-side data fetching in Node.js environments (e.g., for server-side rendering or backend services) using Axios for consistent HTTP request handling.

AXIOS

MCQ

1. Which of the following is NOT a common request option supported by Axios?

   A) Setting HTTP headers

   B) Using query parameters (params)

   C) Defining request timeouts

   D) Automating UI refresh

2. Which property of the Axios response object contains the main data sent from the server?

   A) response.status

   B) response.config

   C) response.data

   D) response.headers

3. How can you globally set a base URL for all Axios requests?

A) By calling axios.setBaseURL() method

B) Using axios.defaults.baseURL property

C) Assigning axios.base property

D) Globally configuring axios.config

4. Which of the following statements about Axios is correct?

A) Axios can only be used in Node.js environments, not in browsers

B) Axios automatically transforms JSON data in HTTP requests and responses

C) Axios requires callbacks instead of Promises for asynchronous code

D) Axios cannot set custom HTTP headers in requests

MCQ

Answer 1: D) Automating UI refresh

Answer 2: C) response.data

Answer 3: B) Using axios.defaults.baseURL property

Answer 4 : B) Axios automatically transforms JSON data in HTTP requests and responses

THANK YOU

Prof. Pavan A C

Department of Computer Science and Engineering