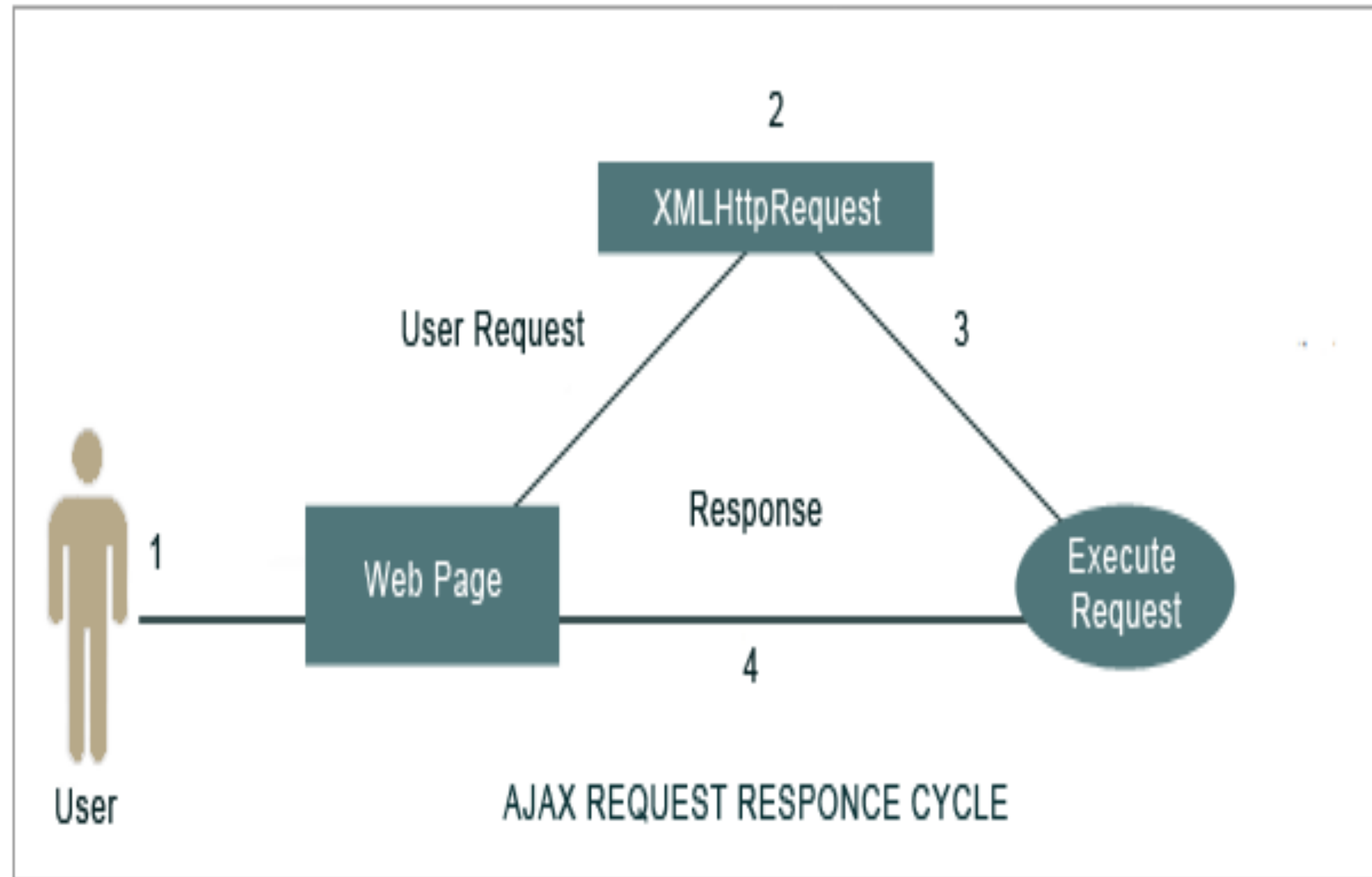


REACTJS - AXIOS AND INTERCEPTOR

ANECO ACADEMY

AJAX



AXIOS

Using Axios with React to Make API Requests



HTTP METHOD

GET	retrieve information
HEAD	retrieve resource headers
POST	submit data to the server.
PUT	save an object at the location
DELETE	delete the object at the location

POSTMAN FOR API REST

POST ▾

{{DEV}}/subbranch

Params

Send ▾

Save ▾

Authorization

Headers (2)

Body ●

Pre-request Script

Tests

Code

☐ form-data

☐ x-www-form-urlencoded

☒ raw

☐ binary

JSON (application/json) ▾

1 ▾

[

2 "subBranchName": "HDFC SUB",

3 "authorizedPersonName": "salman",

4 "email": "salmsasdsan4444444@gmail.com",

5 "mobile": "9894140922",

6 "userId": "sallu22",

7 "password": "12345",

8 "location": "chennai",

9 "account": 10001,

10 "status": "active",

11 "licenseId": "_23232323"

12]

AXIOS

```
E:\Tuto\ReactJs\TRAINING\Part-5\BaseCode\developer-connector>npm install axios
```

GET METHOD IN CLASS COMPONENT

```
import axios from 'axios';  
  
componentDidMount () {  
  axios.get( 'https://jsonplaceholder.typicode.com/posts' )  
    .then( response => {  
      // console.log( response );  
    } );  
}
```

GET METHOD - FUNCTION COMPONENT

```
import axios from 'axios';  
  
useEffect(()=>{  
  axios.get( 'https://jsonplaceholder.typicode.com/posts' )  
    .then( response => {  
      // console.log( response );  
    } );  
  },[])
```


HTTP - POST METHOD

```
postDataHandler = () => {  
    const data = {  
        title: this.state.title,  
        body: this.state.content,  
        author: this.state.author  
    };  
    axios.post('https://jsonplaceholder.typicode.com/posts', data)  
        .then(response => {  
            console.log(response);  
        });  
}
```

HTTP - DELETE METHOD

```
deletePostHandler = () => {  
  axios.delete('https://jsonplaceholder.typicode.com/  
    posts/' + this.props.id)  
    .then(response => {  
      console.log(response);  
    });  
}
```

HTTP - ERROR HANDLING

```
componentDidMount () {  
  axios.get( 'https://jsonplaceholder.typicode.com/posts' )  
    .then( response => {  
      const posts = response.data.slice(0, 4);  
      const updatedPosts = posts.map(post => {  
        return {  
          ...post,  
          author: 'Max'  
        }  
      });  
      this.setState({posts: updatedPosts});  
      // console.log( response );  
    } )  
    .catch(error => {  
      // console.log(error);  
      this.setState({error: true});  
    });  
}
```

INTERCEPTOR

```
import axios from 'axios';

axios.defaults.baseURL = 'https://jsonplaceholder.typicode.com';
axios.defaults.headers.common['Authorization'] = 'AUTH TOKEN';
axios.defaults.headers.post['Content-Type'] = 'application/json';

axios.interceptors.request.use(request => {
  console.log(request);
  // Edit request config
  return request;
}, error => {
  console.log(error);
  return Promise.reject(error);
});
```

INTERCEPTOR

