Working with REST API

What is REST API?

- REST stands for Representational State Transfer.
- REST is an **architectural pattern** for creating an API that uses HTTP as its underlying communication method.
- In the context of a REST API, a resource typically represents a data entity like Product, Employee, Customer etc.
- The HTTP verb (GET, PUT, POST, DELETE) that is sent with each request tells the API what to do with the resource.
- Each resource is identified by a specific URI (Uniform Resource Identifier) or URL (Uniform Resource Locator). The following table shows some typical requests that you see in an API.

Resource	Verb	Outcome
/employees	GET	Gets list of employees
/employees/1	GET	Gets employee with Id = 1
/employees	POST	Creates a new employee
/employees/1	PUT	Updates employee with Id = 1
/employees/1	DELETE	Deletes employee with Id = 1

- Depending on the server side technology you use, there are many frameworks that we can use to build a REST API.
  For example,
  1. If your server side technology is Microsoft Dot Net, you can use **ASP.NET Web API** to create a REST API.
  2. If your server side technology is Java, you can use **Spring or Spring Boot** to create a REST API.

Using Fake REST API:

- We can create a fake REST API using **JSON Server** instead of working on **server-side technologies** directly.
- JSON Server provides a full fake REST API with zero coding in less than 30 seconds.
- Created with <3 for front-end developers who need a quick back-end for prototyping and mocking.

  https://www.npmjs.com/package/json-server

Instatllation Steps:

1. Execute the following command to install JSON server
   D:\ReactJS\hello-app >npm install -g json-server

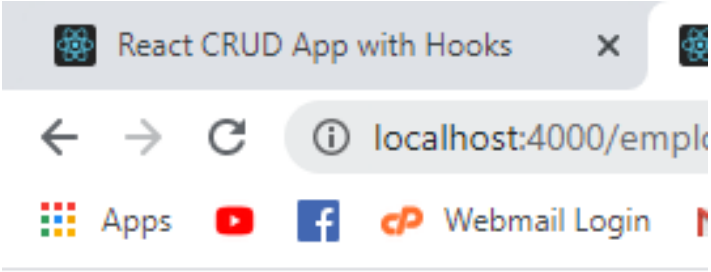
2. Execute the following command to start the server
   D:\ReactJS\hello-app>json-server --watch db.json --port 4000

This automatically creates **db.json** file in the root project folder.
Copy and paste the following JSON data in db.json file.

```
{
  "employees": [
    {
      "id": 1,
      "name": "Scott",
      "salary": 4000
    },
    {
      "id": 2,
      "name": "Allen",
      "salary": 7500
    },
    {
      "id": 3,
      "name": "Jones",
      "salary": 9200
    },
    {
      "id": 4,
      "name": "James",
      "salary": 9200
    },
    {
      "id": 5,
      "name": "Smith",
      "salary": 8400
    }
  ]
}
```

Testing
Open the browser and navigate to
http://localhost:4000/employees/

Note: You can test this REST API using a tool like Postman.




```
[
  {
    "id": 1,
    "name": "Scott",
    "salary": 4000
  },
  {
    "id": 2,
    "name": "Allen",
    "salary": 7500
  },
  {
    "id": 3,
    "name": "Jones",
    "salary": 9200
  },
  {
    "id": 4,
    "name": "James",
    "salary": 9200
  },
  {
    "id": 5,
    "name": "Smith",
    "salary": 8400
  }
]
```

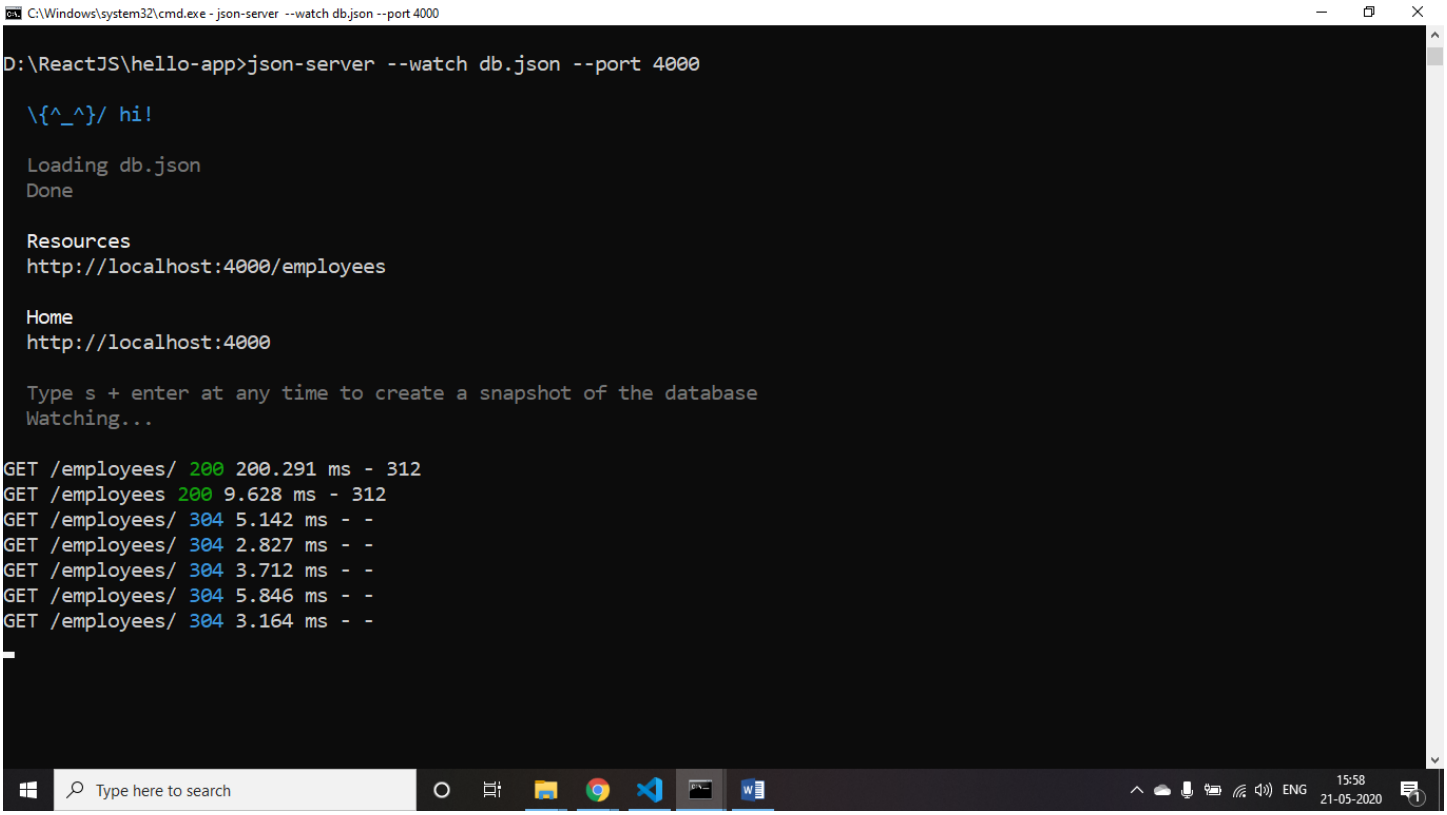
Example to process Get Request

Example:

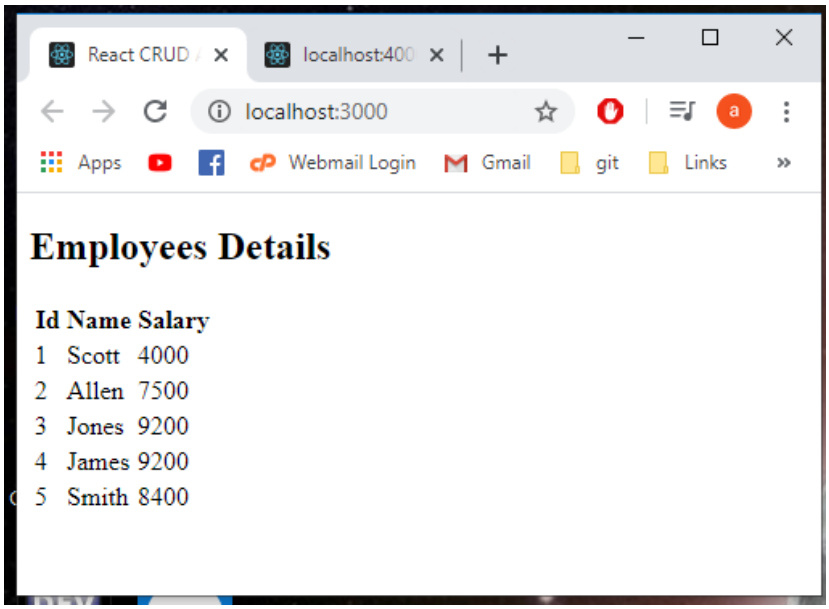
```
import React from 'react'
import ReactDOM from 'react-dom'

class EmployeeComponent extends React.Component {
    constructor(props) {
        super(props);
        this.state = {
            employees: []
        };
    }
    componentDidMount() {
        fetch("http://localhost:4000/employees/")
            .then(res => res.json())
            .then(
                (result) => {
                    this.setState({
                        employees: result
                    });
                }
            );
    }

    render() {
        return (
            <div>
                <h2>Employees Details</h2>
                <table>
                    <thead>
                        <tr>
                            <th>Id</th>
                            <th>Name</th>
                            <th>Salary</th>
                        </tr>
                    </thead>
                    <tbody>
                        {this.state.employees.map(emp => (
                            <tr key={emp.id}>
                                <td>{emp.id}</td>
                                <td>{emp.name}</td>
                                <td>{emp.salary}</td>
                            </tr>
                        ))}
                    </tbody>
                </table>
            </div>
        );
    }
}
```

```
const element = <EmployeeComponent></EmployeeComponent>
ReactDOM.render(element, document.getElementById("root"));
```

Test
Open CMD and run the json-server
D:\ReactJS\hello-app>json-server --watch db.json --port 4000



Open another CMD and run react application
D:\ReactJS\hello-app>npm start



Handling REST api using Axios

- Promise based HTTP client for the browser and node.js

Features

1. Make **XMLHttpRequests** from the browser
2. Make http requests from node.js
3. Supports the **Promise** API
   https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Promise
4. Intercept request and response
5. Transform request and response data
6. Cancel requests
7. Automatic transforms for JSON data
8. Client side support for protecting against **XSRF**

Installation of Axios:
D:\ReactJS\hello-app>npm install --save axios

Example:

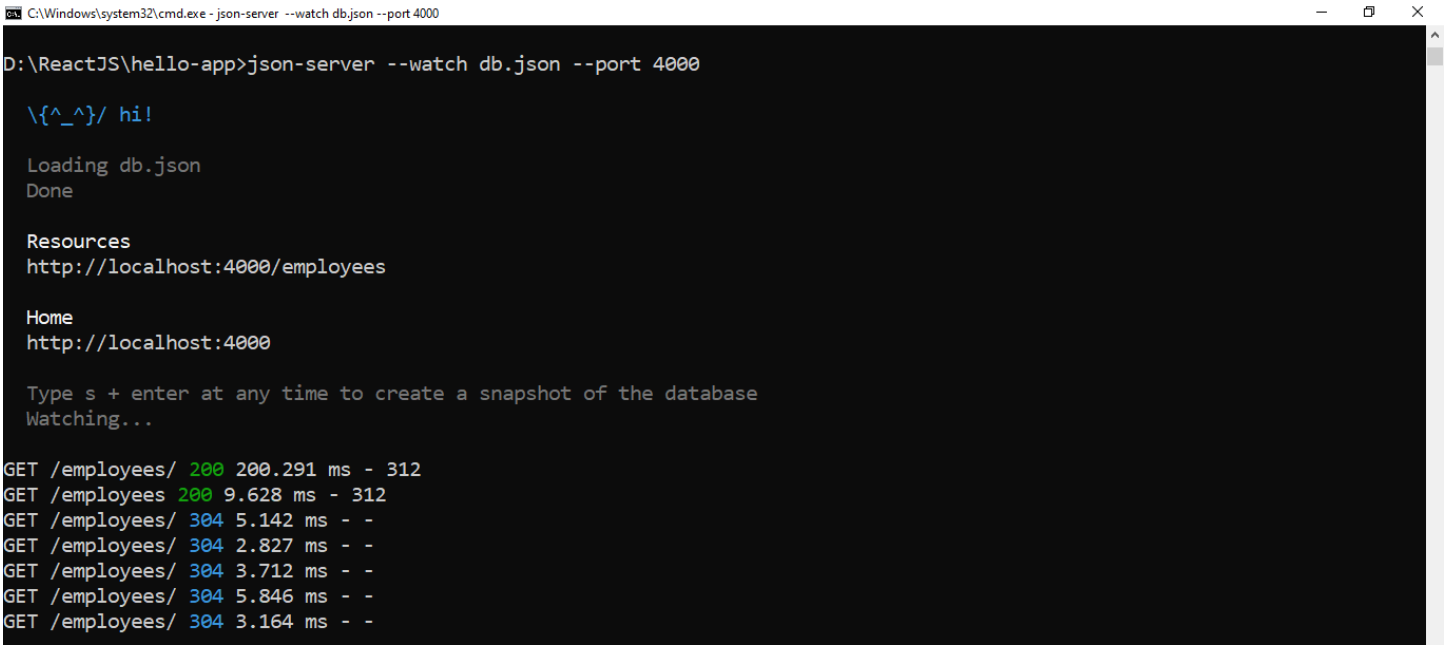
```
import React from "react";
import ReactDOM from "react-dom";
import axios from "axios";

class EmployeeComponent extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      employees: [],
    };
  }
  componentDidMount() {
    axios.get("http://localhost:4000/employees/").then((response) => {
      console.log(response);
      this.setState({
        employees: response.data,
      });
    });
  }

  render() {
    return (
      <div>
        <h2>Employees Details</h2>
        <table>
          <thead>
            <tr>
              <th>Id</th>
```

```
              <th>Name</th>
              <th>Salary</th>
            </tr>
          </thead>
          <tbody>
            {this.state.employees.map((emp) => (
              <tr key={emp.id}>
                <td>{emp.id}</td>
                <td>{emp.name}</td>
                <td>{emp.salary}</td>
              </tr>
            ))}
          </tbody>
        </table>
      </div>
    );
  }
}

const element = <EmployeeComponent></EmployeeComponent>;
ReactDOM.render(element, document.getElementById("root"));
```

Test

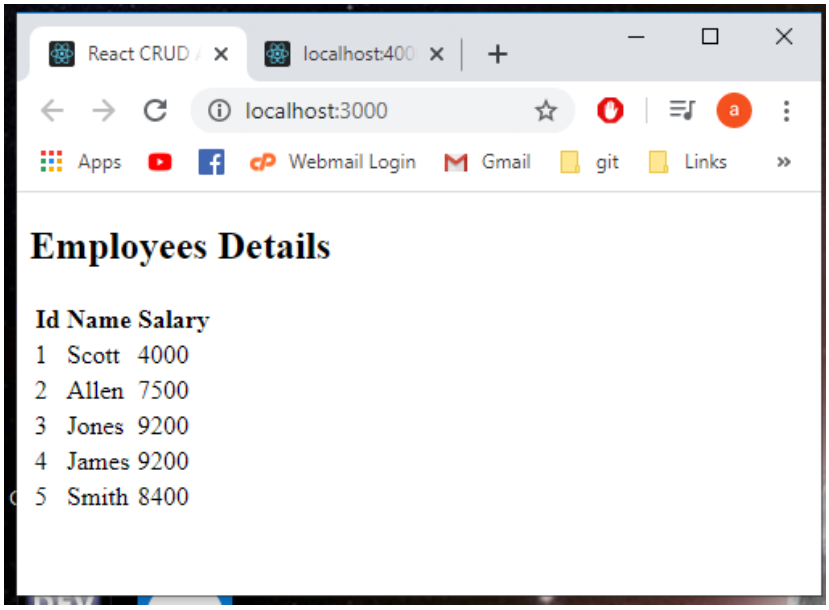
Open CMD and run the json-server

D:\ReactJS\hello-app>json-server --watch db.json --port 4000



Open another CMD and run react application

D:\ReactJS\hello-app>npm start


React CRUD
localhost:400
localhost:3000
Apps
Webmail Login
Gmail
git
Links
Employees Details
Id Name Salary
1 Scott 4000
2 Allen 7500
3 Jones 9200
4 James 9200
5 Smith 8400

Example to perform update operation.

```
import React from "react";
import ReactDOM from "react-dom";
import axios from "axios";

class EmployeeComponent extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      msg: null,
    };
  }
  componentDidMount() {
    axios
      .put("http://localhost:4000/employees/1", {
        id: 1,
        name: "Alice",
        salary: 2222,
      })
      .then((response) => {
        console.log(response);
        this.setState({
          msg: "Add Successfully",
        });
      });
  }

  render() {
    return (
      <div>
        <h2>Employees Component</h2>
        {this.state.msg}
      </div>
    );
  }
}

const element = <EmployeeComponent></EmployeeComponent>;
ReactDOM.render(element, document.getElementById("root"));
```

```
{
  "employees": [
    {
      "id": 1,
      "name": "Alice",
      "salary": 2222
    },
    {
      "id": 2,
      "name": "Allen",
      "salary": 7500
    },
    {
      "id": 3,
      "name": "Jones",
      "salary": 9200
    },
    {
      "id": 4,
      "name": "James",
      "salary": 9200
    },
    {
      "id": 5,
      "name": "Smith",
      "salary": 8400
    }
  ]
}
```

Example: To perform POST request.

index.js

```
import React from "react";
import ReactDOM from "react-dom";
import axios from "axios";

class EmployeeComponent extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      msg: null,
    };
  }
  componentDidMount() {
    axios
      .post("http://localhost:4000/employees/", {
        name: "Raja",
        salary: 1234,
      })
      .then((response) => {
        console.log(response);
        this.setState({
          msg: "Add Successfully",
        });
      });
  }

  render() {
    return (
      <div>
        <h2>Employees Component</h2>
        {this.state.msg}
      </div>
    );
  }
}

const element = <EmployeeComponent></EmployeeComponent>;
ReactDOM.render(element, document.getElementById("root"));
```

db.json

```
{
  "employees": [
    {
      "id": 1,
      "name": "Scott",
      "salary": 4000
    },
    {
      "id": 2,
      "name": "Allen",
      "salary": 7500
    },
    {
      "id": 3,
      "name": "Jones",
      "salary": 9200
    },
    {
      "id": 4,
      "name": "James",
      "salary": 9200
    },
    {
      "id": 5,
      "name": "Smith",
      "salary": 8400
    },
    {
      "name": "Raja",
      "salary": 1234,
      "id": 6
    }
  ]
}
```

Example to perform DELETE request:

index.js

```
import React from "react";
import ReactDOM from "react-dom";
import axios from "axios";

class EmployeeComponent extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      msg: null,
    };
  }
  componentDidMount() {
    axios.delete("http://localhost:4000/employees/1").then((response) => {
      console.log(response);
      this.setState({
        msg: "Deleted Successfully",
      });
    });
  }

  render() {
    return (
      <div>
        <h2>Employees Component</h2>
        {this.state.msg}
      </div>
    );
  }
}

const element = <EmployeeComponent></EmployeeComponent>;
ReactDOM.render(element, document.getElementById("root"));
```

db.json

```
{
  "employees": [
    {
      "id": 2,
      "name": "Allen",
      "salary": 7500
    },
    {
      "id": 3,
      "name": "Jones",
      "salary": 9200
    },
    {
      "id": 4,
      "name": "James",
      "salary": 9200
    },
    {
      "id": 5,
      "name": "Smith",
      "salary": 8400
    },
    {
      "name": "Raja",
      "salary": 1234,
      "id": 6
    }
  ]
}
```

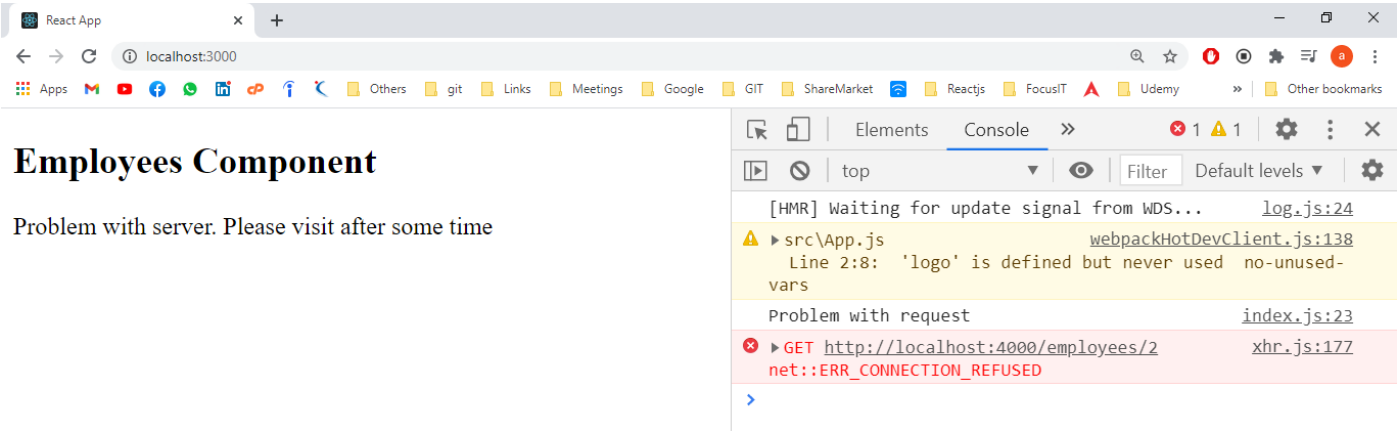
Example to get a single employee info.

```
import React from "react";
import ReactDOM from "react-dom";
import axios from "axios";

class EmployeeComponent extends React.Component {
  constructor(props) {
    super(props);
  }
  componentDidMount() {
    axios.get("http://localhost:4000/employees/2").then((response) => {
      console.log(response.data);
    });
  }

  render() {
    return (
      <div>
        <h2>Employee Component</h2>
      </div>
    );
  }
}

const element = <EmployeeComponent></EmployeeComponent>;
ReactDOM.render(element, document.getElementById("root"));
```

Error Handling with Axios.

Case 1: Handling request error or server error.

```
import React from "react";
import ReactDOM from "react-dom";
import axios from "axios";

class EmployeeComponent extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      msg: null,
      errorMsg: null,
    };
  }
  componentDidMount() {
    axios
      .get("http://localhost:4000/employees/2")
      .then((response) => {
        console.log(response.data);
      })
      .catch((err) => {
        if (err.response) {
          console.log("Problem with response", err.response.status);
        } else if (err.request) {
          console.log("Problem with request");
          this.setState({
            errorMsg: "Problem with server. Please visit after some time",
          });
        } else {
          // anything else
          console.log("Error", err.message);
        }
      });
  }

  render() {
    return (
      <div>
        <h2>Employees Component</h2>
        {this.state.errorMsg}
      </div>
    );
  }
}

const element = <EmployeeComponent></EmployeeComponent>;
ReactDOM.render(element, document.getElementById("root"));
```



Case 2: Handling Response error

```
import React from "react";
import ReactDOM from "react-dom";
import axios from "axios";

class EmployeeComponent extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      msg: null,
      errorMsg: null,
      error404: null,
    };
  }
  componentDidMount() {
    axios
      .get("http://localhost:4000/employees/22")
      .then((response) => {
        console.log(response.data);
      })
      .catch((err) => {
        if (err.response) {
          console.log("Problem with response", err.response.status);
          this.setState({
            error404: "No Response",
          });
        } else if (err.request) {
          console.log("Problem with request");
          this.setState({
            errorMsg: "Problem with server. Please visit after some time",
          });
        } else {
          // anything else
          console.log("Error", err.message);
        }
      });
  }
```

```
  render() {
    return (
      <div>
        <h2>Employees Component</h2>
        {this.state.errorMsg}
        {this.state.error404}
      </div>
    );
  }
}

const element = <EmployeeComponent></EmployeeComponent>;
ReactDOM.render(element, document.getElementById("root"));
```