How to do CRUD operations in ReactJS ?

CRUD operation in React stands for Create, Read, Update, and Delete. CRUD is an important concept for organizing and managing data across the web application. We will perform CRUD operation in the React application with local storage in the form of JavaScript Objects instead of using JSON servers or Axios in React. CRUD operations are referred to as

- **Create:** Creating a post or adding the table row, adding data to the webpage, or creating content.
- **Read:** Reading or retrieving data from a web page
- **Update:** Updating or editing existing content on the webpage
- **Delete:** Deleting and removing the entry or content/post from the webpage

Approach

To implement CRUD operations in React JS on locally stored data, use a data file array.js and create different components home, create, edit and link them with the data present in the js file. Access the data from the JavaScript Localstorage API . Add routing for these components using react-router-dom ann style using the React-Bootstrap UI components and Bootstrap style classes.

Steps to create the application

Step 1: Let's start building the Front-end part with React. To create a new React App, enter the following code into the terminal and hit enter.

```
npx create-react-app crud_app
```

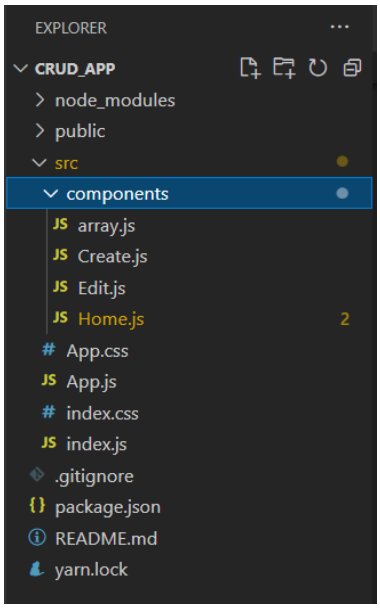
Step 2: Move into the React project folder.

```
cd crud_app
```

Step 3: Install other dependencies using this command

```
npm i react-bootstrap bootstrap@5.1.3 react-router-dom
```

Project Structure:



Updated dependencies in **package.json** file will look like:

```
"dependencies": {
        "@testing-library/jest-dom": "^5.17.0",
        "@testing-library/react": "^13.4.0",
        "@testing-library/user-event": "^13.5.0",
```

```
        "bootstrap": "^5.1.3",
        "react": "^18.2.0",
        "react-bootstrap": "^2.9.0",
        "react-dom": "^18.2.0",
        "react-router-dom": "^6.16.0",
        "react-scripts": "5.0.1",
        "web-vitals": "^2.1.4"
}
```

Example: This example show CRUD operations on a locally stored data file named **array.js** using React JS.

Javascript

```
// Filename - App.js

import React from "react";
import {
    BrowserRouter as Router,
    Route,
    Routes,
} from "react-router-dom";
import "./App.css";
import Create from "./components/Create";
import Edit from "./components/Edit";
import Home from "./components/Home";

function App() {
    return (
        <div className="App">
            <h1 className="geeks">GeeksforGeeks </h1>
            <h3>CRUD App</h3>
            <Router>
                <Routes>
                    <Route path="/" element={<Home />} />
                    <Route
                        path="/create"
                        element={<Create />}
                    />
                    <Route
                        path="/edit"
                        element={<Edit />}
                    />
                </Routes>
            </Router>
        </div>
```

```
    );
}

export default App;
```

Javascript

```
// Filename - components/array.js

// Javascript object named array
// with 3 key-values
const array = [
    {
        id: "1",
        Name: "Shivansh",
        Age: "23",
    },
    {
        id: "2",
        Name: "Simran",
        Age: "22",
    },
    {
        id: "3",
        Name: "Aakash",
        Age: "23",
    },
];

export default array;
```

Javascript

```
// Filename - components/Create.js

import React, { useState } from "react";
import { Button, Form } from "react-bootstrap";
import "bootstrap/dist/css/bootstrap.min.css";
import array from "./array";
import { v4 as uuid } from "uuid";
import { Link, useNavigate } from "react-router-dom";

function Create() {
    // Making usestate for setting and
    // fetching a value in jsx
    const [name, setname] = useState("");
    const [age, setage] = useState("");
```

```
// Using useNavigation for redirecting to pages
let history = useNavigate();

// Function for creating a post/entry
const handelSubmit = (e) => {
    e.preventDefault(); // Prevent reload

    const ids = uuid(); // Creating unique id
    let uni = ids.slice(0, 8); // Slicing unique id

    // Fetching a value from usestate and
    // pushing to javascript object
    let a = name,
        b = age;
    if (name == "" || age == "") {
        alert("invalid input");
        return;
    }
    array.push({ id: uni, Name: a, Age: b });

    // Redirecting to home page after creation done
    history("/");
};

return (
    <div>
        <Form
            className="d-grid gap-2"
            style={{ margin: "5rem" }}
        >
            {/* Fetching a value from input textfirld
                in a setname using usestate*/}
            <Form.Group
                className="mb-3"
                controlId="formBasicName"
            >
                <Form.Control
                    onChange={(e) =>
                        setname(e.target.value)
                    }
                    type="text"
                    placeholder="Enter Name"
                    required
                />
            </Form.Group>

            {/* Fetching a value from input textfirld in
                a setage using usestate*/}
```

```
                    <Form.Group
                        className="mb-3"
                        controlId="formBasicAge"
                    >
                        <Form.Control
                            onChange={(e) =>
                                setage(e.target.value)
                            }
                            type="number"
                            placeholder="Age"
                            required
                        />
                    </Form.Group>

                    {/* handing a onclick event in button for
                        firing a function */}
                    <Button
                        onClick={(e) => handelSubmit(e)}
                        variant="primary"
                        type="submit"
                    >
                        Submit
                    </Button>

                    {/* Redirecting back to home page */}
                    <Link className="d-grid gap-2" to="/">
                        <Button variant="info" size="lg">
                            Home
                        </Button>
                    </Link>
                </Form>
            </div>
        );
}

export default Create;
```

Javascript

```
// Filename - components/Home.js

import React from "react";
import { Button, Table } from "react-bootstrap";
import "bootstrap/dist/css/bootstrap.min.css";
import array from "./array";
import { Link, useNavigate } from "react-router-dom";

function Home() {
```

```
let history = useNavigate();

// You may skip this part if you're
// using react-context api or redux
function setID(id, name, age) {
    localStorage.setItem("id", id);
    localStorage.setItem("Name", name);
    localStorage.setItem("Age", age);
}

// Deleted function - functionality
// for deleting the entry
function deleted(id) {
    let index = array
        .map(function (e) {
            return e.id;
        })
        .indexOf(id);

    // deleting the entry with index
    array.splice(index, 1);

    // We need to re-render the page for getting
    // the results so redirect to same page.
    history("/");
}

return (
    <div style={{ margin: "5rem" }}>
        <Table striped bordered hover size="sm">
            <thead>
                <tr>
                    <th>Name</th>
                    <th>Age</th>
                </tr>
            </thead>
            <tbody>
                {/* Mapping though every element
                    in the array and showing the
                    data in the form of table */}
                {array.map((item) => {
                    return (
                        <tr>
                            <td>{item.Name}</td>
                            <td>{item.Age}</td>

                            {/* getting theid,name, and
                                age for storing these
                                value in the jsx with
```

```
                                onclick event */}
                            <td>
                                <Link to={`/edit`}>
                                    <Button
                                        onClick={(e) =>
                                            setID(
                                                item.id,
                                                item.Name,
                                                item.Age
                                            )
                                        }
                                        variant="info"
                                    >
                                        Update
                                    </Button>
                                </Link>
                            </td>

                            {/* Using thr deleted function passing
                                the id of an entry */}
                            <td>
                                <Button
                                    onClick={(e) =>
                                        deleted(item.id)
                                    }
                                    variant="danger"
                                >
                                    Delete
                                </Button>
                            </td>
                        </tr>
                    );
                })}
            </tbody>
        </Table>

        {/* Button for redirecting to create page for
            insertion of values */}
        <Link className="d-grid gap-2" to="/create">
            <Button variant="warning" size="lg">
                Create
            </Button>
        </Link>
    </div>
  );
}

export default Home;
```

Javascript

```
// Filename - Edit.js
import React, { useEffect, useState } from "react";
import { Button, Form } from "react-bootstrap";
import "bootstrap/dist/css/bootstrap.min.css";
import array from "./array";
import { Link } from "react-router-dom";
import { useNavigate } from "react-router-dom";

function Edit() {
    // Here usestate has been used in order
    // to set and get values from the jsx
    const [name, setname] = useState("");
    const [age, setage] = useState("");
    const [id, setid] = useState("");

    // Used for navigation with logic in javascript
    let history = useNavigate();

    // Getting an index of an entry with an id
    let index = array
        .map(function (e) {
            return e.id;
        })
        .indexOf(id);

    // Function for handling the edit and
    // pushing changes of editing/updating
    const handelSubmit = (e) => {
        // Preventing from reload
        e.preventDefault();
        if (name == "" || age == "") {
            alert("invalid input");
            return;
        }

        // Getting an index of an array
        let a = array[index];

        // Putting the value from the input
        // textfield and replacing it from
        // existing for updation
        a.Name = name;
        a.Age = age;


        // Redirecting to main page
```

```
        history("/");
    };

    // Useeffect take care that page will
    // be rendered only once
    useEffect(() => {
        setname(localStorage.getItem("Name"));
        setage(localStorage.getItem("Age"));
        setid(localStorage.getItem("id"));
    }, []);

    return (
        <div>
            <Form
                className="d-grid gap-2"
                style={{ margin: "5rem" }}
            >
                {/* setting a name from the
                    input textfiled */}
                <Form.Group
                    className="mb-3"
                    controlId="formBasicEmail"
                >
                    <Form.Control
                        value={name}
                        onChange={(e) =>
                            setname(e.target.value)
                        }
                        type="text"
                        placeholder="Enter Name"
                    />
                </Form.Group>

                {/* setting a age from the input textfiled */}
                <Form.Group
                    className="mb-3"
                    controlId="formBasicPassword"
                >
                    <Form.Control
                        value={age}
                        onChange={(e) =>
                            setage(e.target.value)
                        }
                        type="number"
                        placeholder="Age"
                    />
                </Form.Group>

                {/* Hadinling an onclick event
```

```
                    running an edit logic */}
                <Button
                    onClick={(e) => handelSubmit(e)}
                    variant="primary"
                    type="submit"
                    size="lg"
                >
                    Update
                </Button>

                {/* Redirecting to main page after editing */}
                <Link className="d-grid gap-2" to="/">
                    <Button variant="warning" size="lg">
                        Home
                    </Button>
                </Link>
            </Form>
        </div>
    );
}

export default Edit;
```

CSS

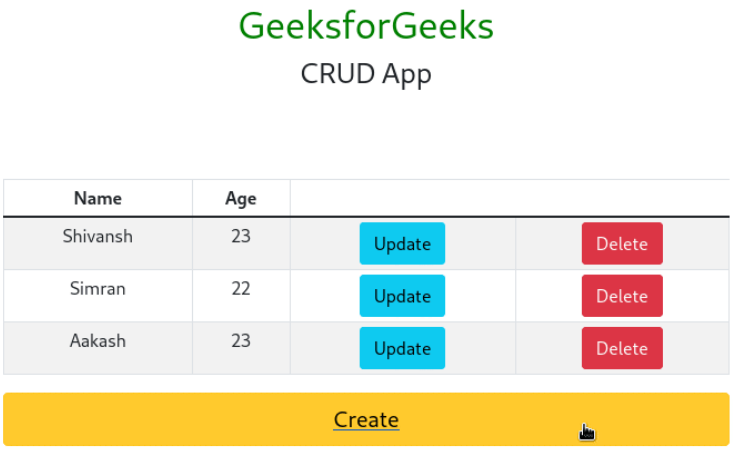
```
/* App.css */
.App {
    text-align: center;
}
.geeks {
    color: green;
}
```

Steps to run the application:

Step 1: Use this npm command in project directory to run the applicaion.

```
npm start
```

Output: This output will be visible on **http://localhost:3000/**



The CRUD operations in React JS are performed on Local Storage, for learning CRUD operation with ReactJS and NodeJS please refer to How to build a basic CRUD app with Node.js and ReactJS?

Whether you're preparing for your first job interview or aiming to upskill in this ever-evolving tech landscape, GeeksforGeeks Courses are your key to success. We provide top-quality content at affordable prices, all geared towards accelerating your growth in a time-bound manner. Join the millions we've already empowered, and we're here to do the same for you. Don't miss out - check it out now!

Last Updated : 11 Oct, 2023

10

Previous

How to Test React Components using Jest ?

Next

How to use class syntax in Typescript ?

Similar Reads

How to build a basic CRUD app with Node.js and ReactJS ?

How to create Shopping Cart Button in ReactJS?

ReactJS Introduction

ReactJS Onsen UI AlertDialogButton Component

ReactJS Rendering Elements

ReactJS MDBootstrap Spinner Component

ReactJS Evergreen Avatar Component

ReactJS Semantic UI Search Module

ReactJS Props - Set 1

ReactJS Evergreen Autocomplete Component

Complete Tutorials

JavaScript Project Ideas with Source Code

SAP - Systems Applications and Products | A Complete Learning Hub

Spring MVC Tutorial

Spring Boot Tutorial

shiv_ka_a...

d , React-Questions ,
/eb Technologies

Additional Information

A-143, 9th Floor, Sovereign Corporate Tower, Sector-136, Noida, Uttar Pradesh - 201305

Django Tutorial

Python Projects

Python Tkinter

Web Scraping

OpenCV Python Tutorial

Python Interview Question

DevOps

Git

AWS

Docker

Kubernetes

Azure

GCP

DevOps Roadmap

System Design

What is System Design

Monolithic and Distributed SD

High Level Design or HLD

Low Level Design or LLD

Crack System Design Round

System Design Interview Questions

Grokking Modern System Design

NCERT Solutions

Class 12

Class 11

Class 10

Class 9

Class 8

Complete Study Material

Commerce

Operating Systems

Computer Network

Database Management System

Software Engineering

Digital Logic Design

Engineering Maths

Competitive Programming

Top DS or Algo for CP

Top 50 Tree

Top 50 Graph

Top 50 Array

Top 50 String

Top 50 DP

Top 15 Websites for CP

JavaScript

TypeScript

ReactJS

NextJS

AngularJS

NodeJS

Express.js

Lodash

Web Browser

School Subjects

Mathematics

Physics

Chemistry

Biology

Social Science

English Grammar

Management & Finance

Accountancy

Business Studies

Indian Economics

Macroeconomics

Microeconimics

Statistics for Economics

Management

HR Managament

Income Tax

Finance

Economics

UPSC Study Material

Polity Notes

Geography Notes

History Notes

Science and Technology Notes

Economy Notes

Ethics Notes

Previous Year Papers

SSC/ BANKING

SSC CGL Syllabus

SBI PO Syllabus

SBI Clerk Syllabus

IBPS PO Syllabus

IBPS Clerk Syllabus

SSC CGL Practice Papers

Colleges

Indian Colleges Admission & Campus Experiences

Top Engineering Colleges

Top BCA Colleges

Top MBA Colleges

Top Architecture College

Choose College For Graduation

Companies

IT Companies

Software Development Companies

Artificial Intelligence(AI) Companies

CyberSecurity Companies

Service Based Companies

Product Based Companies

PSUs for CS Engineers

Preparation Corner

Company Wise Preparation

Preparation for SDE

Experienced Interviews

Internship Interviews

Competitive Programming

Aptitude Preparation

Puzzles

Exams

JEE Mains

JEE Advanced

GATE CS

NEET

UGC NET

More Tutorials

Write & Earn