**What is the difference between Single Page applications and Multi Page applications?**

Single-page Applications are perfect for creating dynamic, fast-loading websites, especially if you are designing to create a mobile app. The biggest downside of this model is that it has poor search engine optimization and is ranked lower in search engines. Therefore, it is ideal for social networks, private groups, and SaaS applications where SEO is not required. Gmail, Google Maps, Facebook, and GitHub are among the examples.

On the Contrary, Multi-Page Applications are valuable for online stores, catalogs, and commercial websites because they can handle large amounts of data that require several pages, features, and menus. However, React Js developers are working on ways that will allow single-page applications to handle large amounts of data. If you want to design a SPA and need a good framework, you should [look into hiring a React JS application development Company](https://www.brihaspatitech.com/react-js-development-company). Amazon and eBay are examples.

**What are the advantages of a single-page application?**

The following are some of the advantages of using single-page applications (SPAs):

1. All webpage resources are loaded only once throughout the application because SPA is faster, and data is the sole resource that is sent.
2. Also, SPA effectively caches local storage because it sends only one request, retains all of the data, and uses it even when offline.
3. In fact, SPA simplifies and streamlines development tasks because it eliminates the need to create code to render pages on the server.
4. Chrome makes it simple to debug SPAs since it allows you to explore page elements and watch network activity.

**When should you avoid using single-page applications?**

While SPA has its benefits, there are some situations where it is not appropriate to use. So, there are some points where you should not use Single page apps. Even you can hire react js developer who can guide you on where you should not use the library:

* **SEO difficulties**: Any web app is written in JavaScript loads data without reloading the page and only when the user requests it. This means there are no specific URLs optimized for search engines, and the content is not visible to search engines. The problem can only be solved by server-side rendering.
* **Time to download**: Users' browsers will take longer to load content if the platform is sophisticated, vast, and poorly optimized.
* **Support for JavaScript**: You won't be able to use the full functionality of an app without this feature. Users who disable JS in their browser will be unable to fully utilize the app.

**How can you make a single-page React.js app?**

To make a simple single-page application with React, follow these steps:

**Step 1: Check the background settings**

1. Firstly, Install all libraries and packages in your development environment.
2. To begin React, create an HTML page.
3. Install all of the React components you will need.
4. To get the web app, use webpack or another bundler to fuse the installed "components."

**Step 2: Create the App**

1. To create a React App in the chosen location, use "npx create-react-app app-name."
2. Although, it also generates a directory called "app-name" that contains some default files.

**Step 3: Further, run the following commands to install react-router-dom and route the requests:**

npm install react-router-dom

**Step 4: To wrap the App component, we need a router.**

1. HashRouter – to create URLs like example.com/#/about
2. BrowserRouter – to create URLs like example.com/#/about

**Step 5: Include the below code in your src/index.js file.**

import React from ‘react’

import { render } from ‘react-dom’

import { BrowserRouter } from ‘react-router-dom’

import App from ‘./App’

render(

<BrowserRouter>

<App />

</BrowserRouter>,

document.querySelector(‘#root’)

)

**Step 6: Write the following code in a file called src/pages/HomePage.js:**

import React from “react”;

export default function HomePage() {

return (

<>

<h1>Hey from HomePage</h1>

<p>This is your awesome HomePage subtitle</p>

</>

);

}

**Step 7: With the below code, create a file of src/pages/UserPage.js:**

import React from “react”;

import { useParams } from “react-router-dom”;

export default function UserPage() {

let { id } = useParams();

return (

<>

<h1>Hello there user {id}</h1>

<p>This is your awesome User Profile page</p>

</>

);

}

**Step 8:**

By Using Switch and Route, choose and integrate the routers you want to utilize. Switch combines all routes and ensures that they are prioritized from top to bottom. Individual routes, on the other hand, are defined by the route. Also, the routes should be included in your App.js file.

import React from ‘react’

import { Route, Switch } from ‘react-router-dom’

// We will create these two pages in a moment

import HomePage from ‘./pages/HomePage’

import UserPage from ‘./pages/UserPage’

export default function App() {

return (

<Switch>

<Route exact path=”/” component={HomePage} />

<Route path=”/:id” component={UserPage} />

</Switch>

)

}

The code above maps the root route (/) to HomePage and dynamically maps additional pages to UserPage.

**Step 9. Use Link to navigate to a page within the SPA.**

Now, Include the following code in the **src/pages/HomePage.js** file.

import React from ‘react’

import { Link } from ‘react-router-dom’

export default function HomePage() {

return (

<div className=”container”>

<h1>Home </h1>

<p>

<Link to=”/your desired link”>Your desired link.</Link>

</p>

</div>

)

}

**Step 10: Run the code and inspect the development server on localhost.**

Further, to create any simple app, we have a "primary parent component."

1. Next, each app page becomes a separate component that feeds into the "main component"; "React Router" assists in determining which "components" to show and which to conceal.]
2. **Displaying the Initial Frame -> Creating the Content Pages -> Using React Router -> Fixing the Routing -> Adding Some CSS -> Highlighting the Active Link** are the procedures.

**Wrapping Up**

On the whole, Single-page React.js apps don't require switching to a new page. In this context, the pages are referred to as views, and they often load inline within the same page. When altering the page content, AJAX is constantly active. Indeed, you can [hire react js developer to create your Single-page app](https://www.brihaspatitech.com/hire-react-js-developer) for your business. No doubt, react comes with a slew of routing mechanisms to aid you. Routing helps you match URLs to destinations other than physical pages, such as "individual views" in single-page apps.