Advanced ReactJS

Routing and Deployment

React Router

Router Types

BrowserRouter: This is the most commonly used router in React applications. It uses the HTML5 history API to keep track of the current URL and update the UI accordingly. It is ideal for applications that need clean and simple URLs, and can be used for both server-side rendering and client-side rendering.

HashRouter: This router uses the hash portion of the URL to keep track of the current location. It is a fallback for older browsers that do not support the HTML5 history API, and can be used for static websites or applications that don't require clean URLs.

MemoryRouter: This router is used for testing and development purposes. It keeps the history of the location in memory and does not update the URL in the address bar.

StaticRouter: This router is used for server-side rendering. It renders a route to a static HTML file and can be used with any server-side rendering framework.

NativeRouter: This router is used for React Native applications. It uses the React Native Navigation library to handle navigation.

React Router

Params

- URL parameter is a way to pass information about a click through its URL
- useParams hook enable us to use parameters to create dynamic routing in React applications

Deployment

Week 10

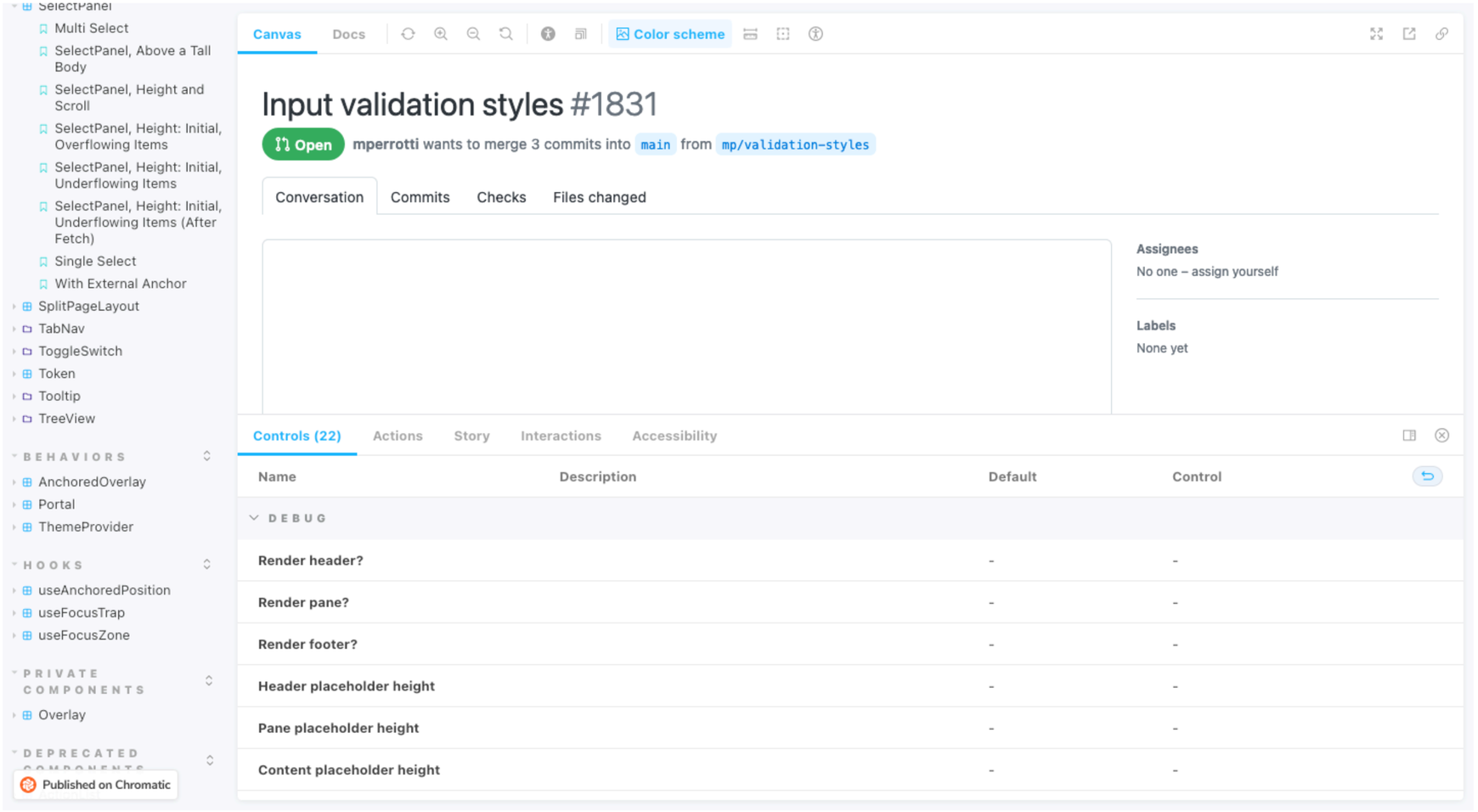
Storybook

React Storybook is an open-source tool that allows developers to build and test UI components in isolation from the rest of the application. It provides a user interface for visually displaying and testing individual components, allowing developers to rapidly iterate on and fine-tune their components.

Storybook is widely used in the React community as a tool for improving the quality and consistency of UI components.

https://storybook.js.org

Storybook



Screenshot from Github's Storybook

Environment Variables

Environment variables are values that can be set outside of the application code and are used to configure the application behavior based on the environment it is running in. They are commonly used to store sensitive information such as API keys or database credentials, as well as configuration values such as the application's URL.

To use environment variables in CRA, you can use the "process.env" object to access their values.

“API_KEY" can be accessed with “process.env.API_KEY"

Note that you must prefix the environment variable with "REACT_APP_" in order for it to be recognized by React, so it should be named "REACT_APP_API_KEY".

Environment Variables

With Vite

`import meta.env.ENV_NAME`

https://vitejs.dev/guide/env-and-mode

Deploying React Applications

Some of the popular platforms that support automatic-deployment from a Git repo

- Vercel

- Netlify

- Render

- Github Pages