



Introduction to React DevTools

Estimated time: 10 minutes

Objectives

After completing this reading, you will be able to:

- Explain React DevTools and its key features
- Explain how to install and set up React DevTools

Introduction

React DevTools is a browser extension that helps developers monitor, debug, and analyze React applications. Whether you're building web or cross-platform native mobile apps with React Native, React DevTools allows you to see the component tree, component state, and more.

What is React DevTools?

React DevTools is a browser extension that adds a number of specific features for the examination and diagnosis of React applications on top of the browser's default DevTools. It enables developers to:

- Review and modify changes done to the component tree.
- See the current props and state for each of the components represented on the page.
- Supervise the work of components and find out which rendering is slower using a profiler.

- Use advanced features like **hook inspection** and **context debugging**.

React DevTools is only supported as a browser extension within Google Chrome and Firefox and as a standalone application if we are using React TV.

Key Features of React DevTools

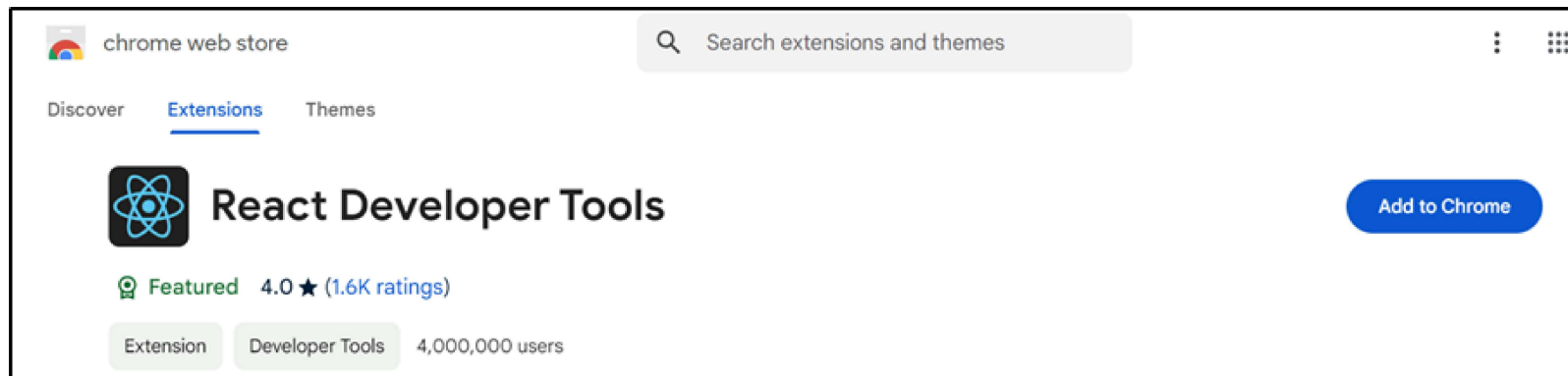
- **Component Tree Inspection:** This allows developers to see a list of the current application's React component tree in an effort to determine the relationships of parent and child components.
- **Props and State Inspection:** This feature shows the current props and state of any component which has been selected and aid developers to determine how data is being transferred through the application.
- **Performance Profiling:** Enhances the developers' capacity to evaluate the application by measuring and showing aspect times of rendering. This aspect is used in isolation and improvement of the slow components, which makes the consideration of this aspect significant.
- **Hooks Inspection:** It also provides information about parts that utilize React hooks, meaning it displays the values of hooks, such as `useState` or `useEffect`, at the current point in time.
- **Edit Component Properties:** DevTools is integrated into the development environment to enable the developers to modify some of the props and state, and infer how the application will operate with the modification.
- **Context API Debugging:** React DevTools is useful to see which component pulls the specific context. This enables developers to work with the Context API for state management more efficiently and to debug the applications with such a mechanism.

How to Install and Set Up React DevTools

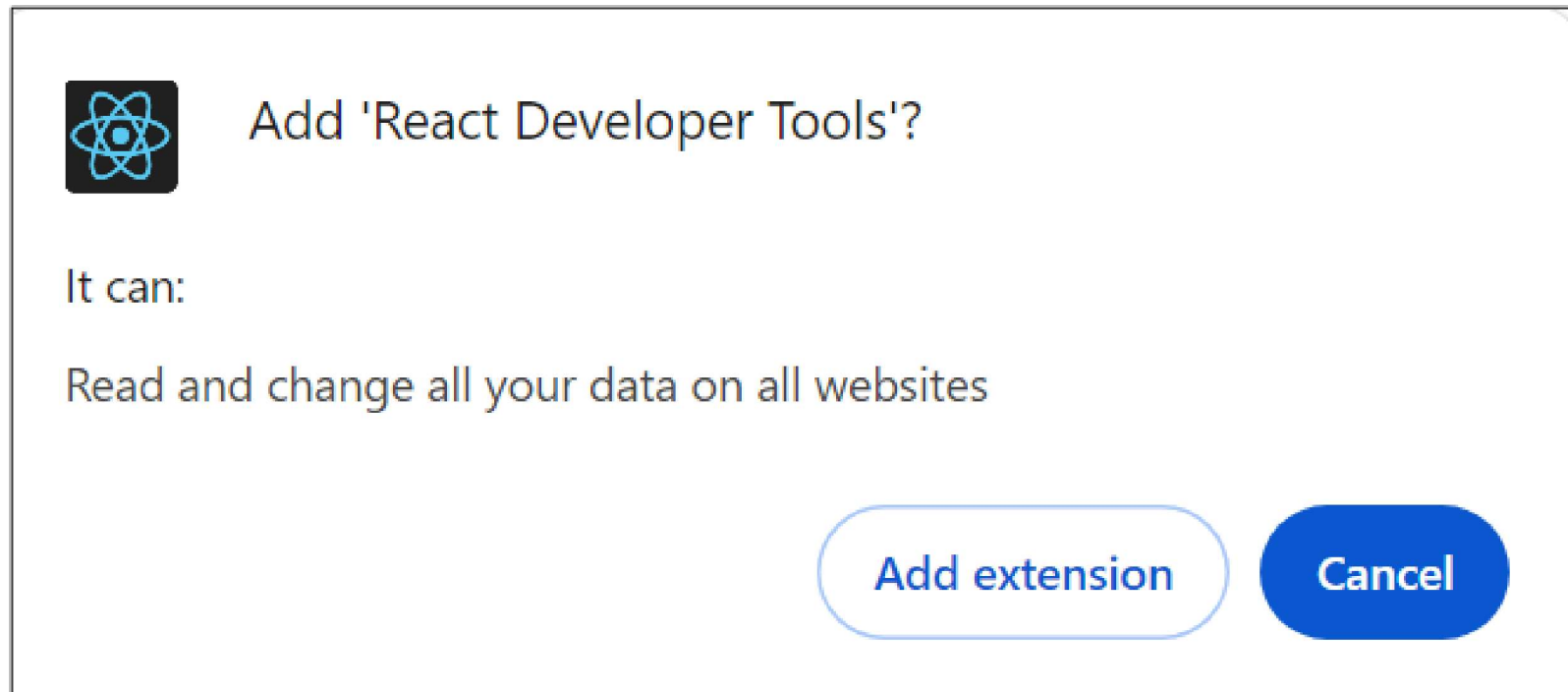
For Web applications

1. Google Chrome:
 - Visit the Chrome Web Store or search React Dev tool for Chrome in Google and search for the "React Developer Tools" name.

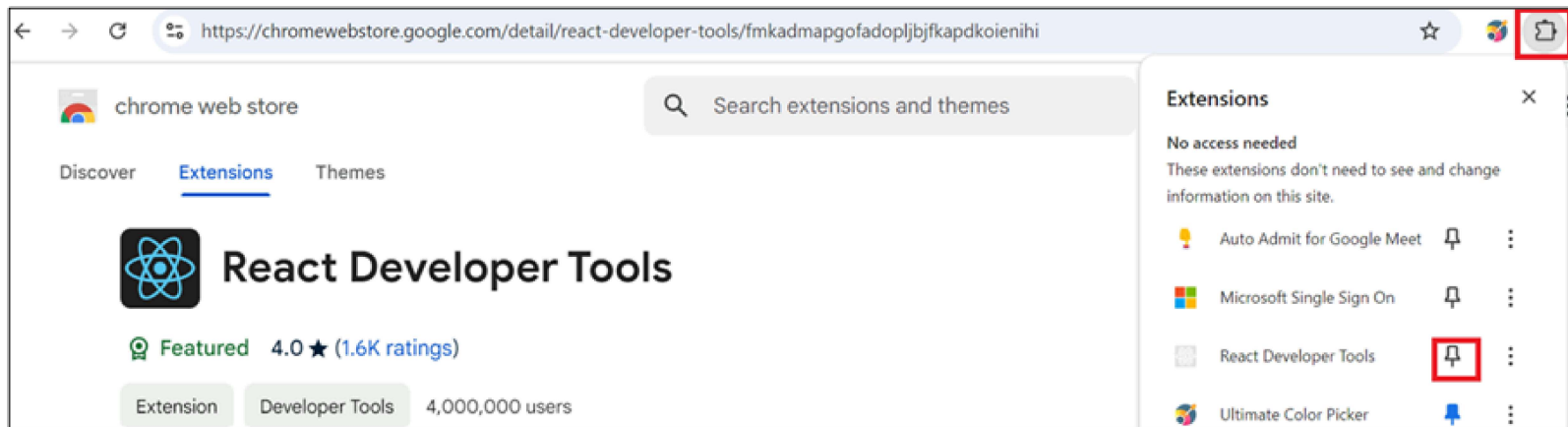
- Click "Add to Chrome" to install the extension.



- A prompt will appear asking to add the extension. Here, you need to click "Add extension."



- To make it appear on the browser, click the extension icon. Then, click the "React Developer Tool" pin.



- This action will add the icon for the tool in the menu bar of Chrome.



Note: If you are running any React Native application, then this icon will become red. If not, it will remain transparent.

2. Firefox:

- Similarly, you need to go to the Firefox Add-ons page and search for "React Developer Tools."
- Click "Add to Firefox" to install the extension.

Tips for effective use of React DevTools

- Keep DevTools open while developing: React DevTools offers updates in real-time, making it convenient to keep the tool open as you develop your application to monitor any problems.
- Debug context usage: When using Context API, use isn't very intuitive, but take advantage of React DevTools to identify which components are using a given context. This is useful when trying to get a feel of how data flows in a context or when dealing with the debugging process.
- Explore experimental features: There are many ways to create MFEs. React's DevTools often include new experimental features created by the React team. You should watch these so that you can adopt the new

tools and enhanced methods.

Conclusion

Anyone using React or React Native should necessarily use React DevTools as it gives more detailed information about the component tree, their dimensions, states, and other parameters of the application. When you learn how to work with it, you will enhance your development process by allowing you to quickly locate issues and resolve them effectively while also tweaking your app for greater performance and functionality. No matter the business, you are new, or you have been a developer for some time, incorporating React DevTools into your tool set is a step towards making better React applications.

Author(s)

Richa Arora