

REACT JS

Redux Dev Tools

Sreekanth M. E.

Freelance Trainer & Consultant

<http://www.SreekanthME.com>

Redux DevTools record every **action**, as well as the **state** after an action has completed.

State of the SPA at any point in its history can be viewed.

It even allows to shuttle back and forth in time using time travel debugging features.

Redux Devtools Extension has to be installed from the chrome store.

This will add a new tab named “Redux” in chrome DevTools.



Redux DevTools

offered by remotedev.io

★★★★★ (283)

[Developer Tools](#)

252,461 users

ADDED TO CHROME

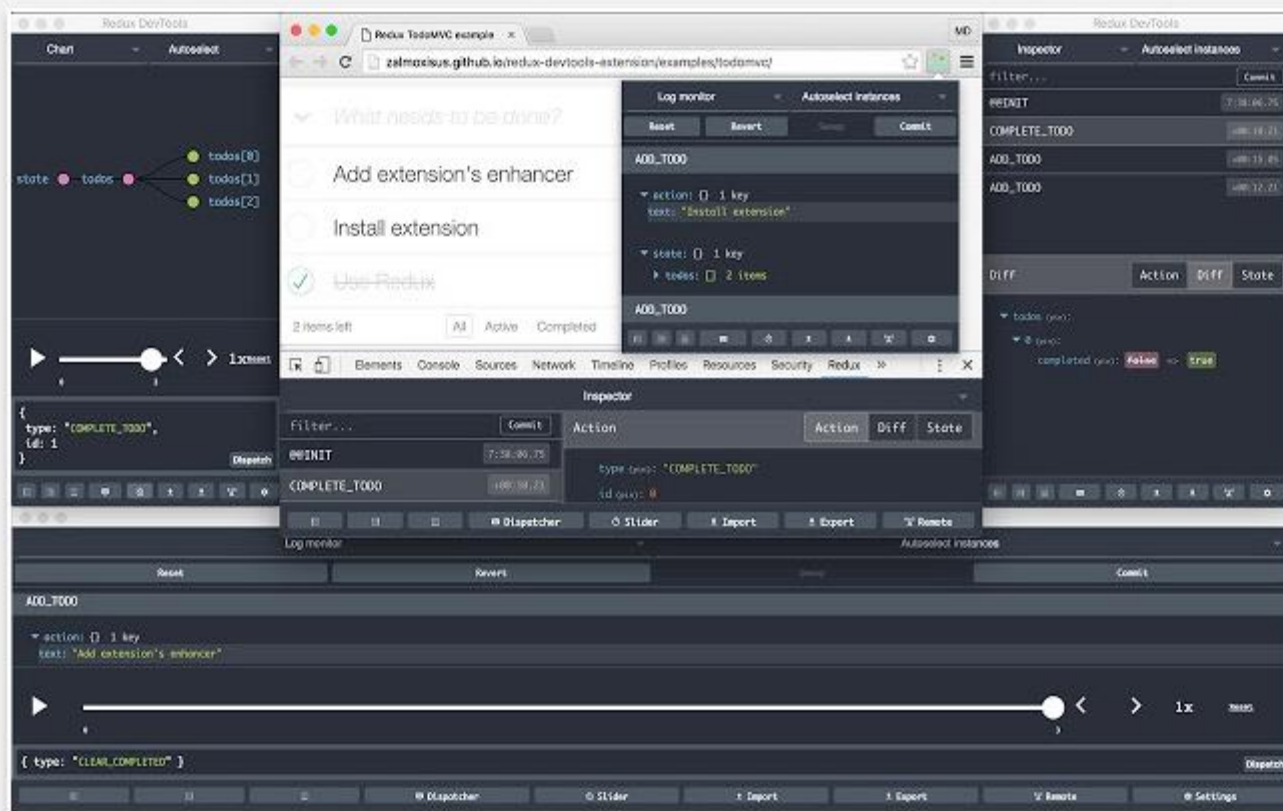


OVERVIEW

REVIEWS

SUPPORT

RELATED



Compatible with your device

Redux DevTools for debugging application's state changes.

The extension provides power-ups for your Redux development workflow. Apart from Redux, it can be used with any other architectures which handle the state.

It's an opensource project. See the official repository for more details:
<https://github.com/zalmoxisus/redux-devtools-extension>

[Website](#)

[Report Abuse](#)

Additional Information

Version: 2.15.1

Updated: May 7, 2017

Size: 1.54MiB

Language: English

Just installing chrome extension is not enough.
Need to add some additional code to the
React App.

Otherwise, "No store found." message will be
displayed in Redux tab.

<https://github.com/zalmoxisus/redux-devtools-extension>

`npm install --save redux-devtools-extension`

TOP SECTION:

In this section, a dropdown shows three modes of Redux DevTools: Log Monitor, Inspector & Chart.

Inspector: The most used mode and the default mode. Shows action log, state and details of each action.

Chart: Shows the state in a tree-like structure.

Log Monitor: Logs state and actions.

MIDDLE LHS SECTION:

Has action log that shows, in real time, the actions that are firing on the current page.

When individual action is clicked, it displays two options (**Jump** and **Skip**). Jump will take SPA to the state of the app when the action was fired. Skip disables the action and shows the SPA as if that action never occurred.

This is one way of time travel for debugging.

MIDDLE RHS SECTION:

Has four tabs named Action, State , Diff, and State which show details of the action selected in action log.

State: Shows entire state tree at the time of the action. The three sub tabs; **Tree** , **Chart** , and **Raw** display the same data in different formats.

Action: Shows action **type** and its **payload** (if any).

Diff: Shows only what the individual action changed in the state tree.

Test: Creates a test format in some pre-provided testing frameworks.

MAGIC TOOLBAR (BOTTOM SECTION):

Starting from the far left, the first three options are for creating an undocked window of Redux tab either to the left, right, or bottom.

Pause: Stops recording of actions. But doesn't block them. This is just to avoid cluttering the action log & the recording.

Lock: Freezes changes to the state by blocking any action from being fired.

Persist: Keeps current state across page reloads.

Dispatcher: Toggles the dispatching module. **Dispatching module can be used to fire custom actions.**

MAGIC TOOLBAR (BOTTOM SECTION):

Slider: Time travels through the action log, from the beginning, and live updates the app to show the running state. It comes with various options to play/pause the recording. This is another way to time travel for debugging.

Import and Export: Imports/Exports state as JSON file. Helps to run app off of a custom state.

Remote: Creates undocked window of Redux tab.

CHANGING THE HISTORY

It is possible to remove past actions from history while preserving actions that happened later.

This can be done by disabling actions in Inspector mode and using the buttons named **Reset**, **Revert**, **Sweep**, and **Commit** in Log Monitor mode.

CHANGING THE HISTORY

Sweep: Permanently deletes disabled actions.

Commit: Removes all actions from action log and the current SPA state becomes the initial state of the SPA.

Reset: Permanently deletes all actions and goes back to initial state.

Revert: Rollback all actions since last commit.

This is another way of time travel for debugging.

Keeping Redux DevTools enabled in Production doesn't cause any problems.

Usually end-users don't have the browser extension installed.

However, hacking the system might become easier.

END OF CHAPTER

Several thin, white, parallel lines of varying lengths and orientations are positioned in the bottom right corner of the slide, creating a modern, abstract graphic element.

APPENDIX