Redux-Devtools provide us debugging platform for Redux apps. It allows us to perform time-travel debugging and live editing. Some of the features in official documentation are as follows −

* It lets you inspect every state and action payload.
* It lets you go back in time by “cancelling” actions.
* If you change the reducer code, each “staged” action will be re-evaluated.
* If the reducers throw, we can identify the error and also during which action this happened.
* With persistState() store enhancer, you can persist debug sessions across page reloads.

There are two variants of Redux dev-tools as given below −

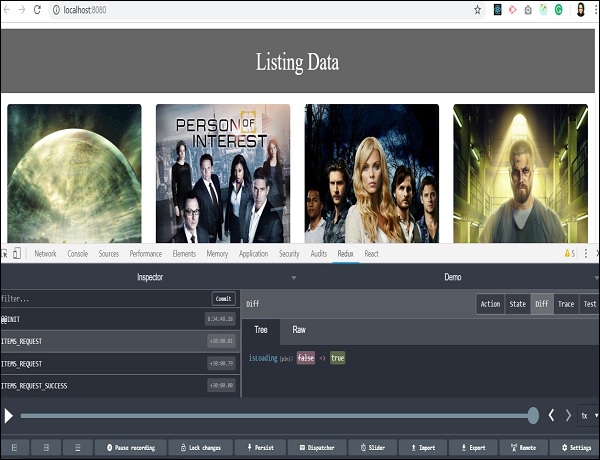
**Redux DevTools** − It can be installed as a package and integrated into your application as given below −

<https://github.com/reduxjs/redux-devtools/blob/master/docs/Walkthrough.md#manual-integration>

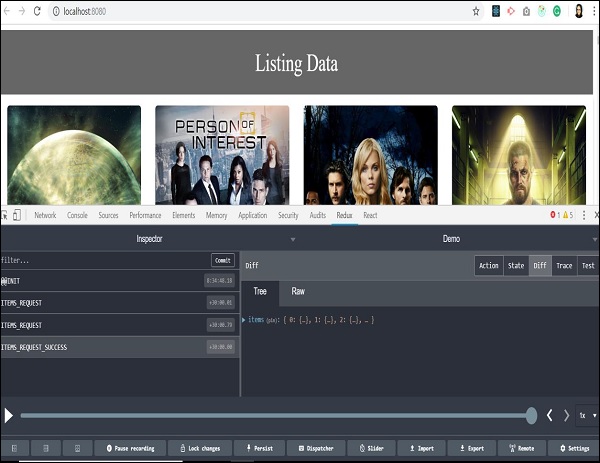
**Redux DevTools Extension** − A browser extension that implements the same developer tools for Redux is as follows −

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

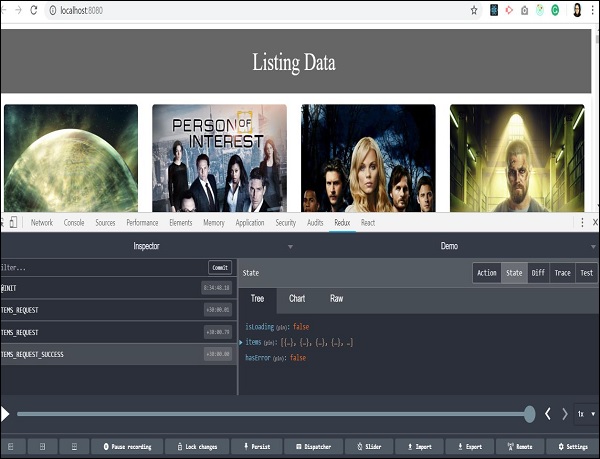
Now let us check how we can skip actions and go back in time with the help of Redux dev tool. Following screenshots explain about the actions we have dispatched earlier to get the listing of items. Here we can see the actions dispatched in the inspector tab. On the right, you can see the Demo tab which shows you the difference in the state tree.



You will get familiar with this tool when you start using it. You can dispatch an action without writing the actual code just from this Redux plugin tool. A Dispatcher option in the last row will help you with this. Let us check the last action where items are fetched successfully.

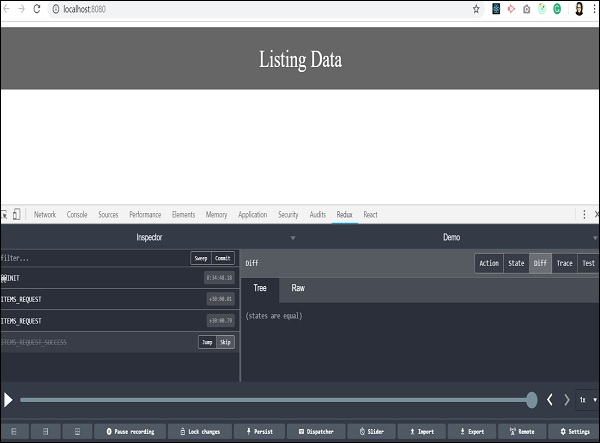


We received an array of objects as a response from the server. All the data is available to display listing on our page. You can also track the store’s state at the same time by clicking on the state tab on the upper right side.



In the previous sections, we have learnt about time travel debugging. Let us now check how to skip one action and go back in time to analyze the state of our app. As you click on any action type, two options: ‘Jump’ and ‘Skip’ will appear.

By clicking on the skip button on a certain action type, you can skip particular action. It acts as if the action never happened. When you click on jump button on certain action type, it will take you to the state when that action occurred and skip all the remaining actions in sequence. This way you will be able to retain the state when a particular action happened. This feature is useful in debugging and finding errors in the application.



We skipped the last action, and all the listing data from background got vanished. It takes back to the time when data of the items has not arrived, and our app has no data to render on the page. It actually makes coding easy and debugging easier.