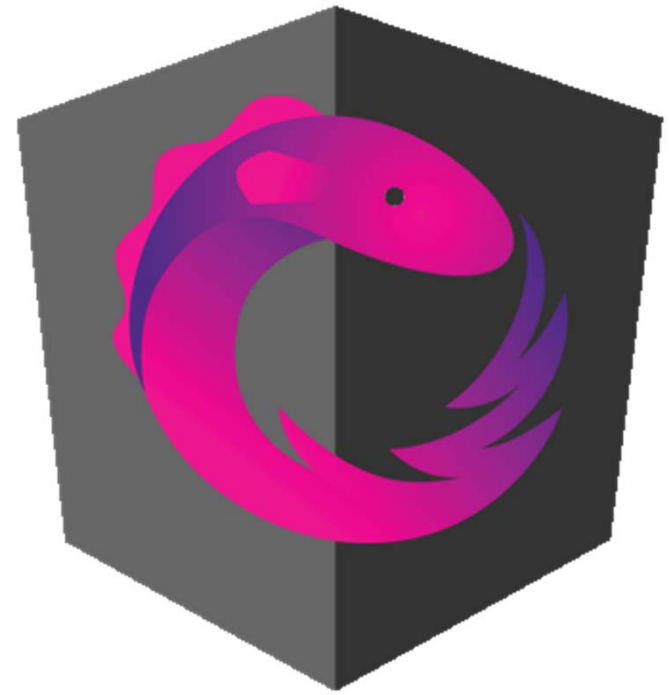
The background is a solid teal color with various white and light blue abstract patterns. These include concentric circles, a crescent moon in the upper right, a ladder-like structure on the left, and various geometric shapes and lines scattered throughout, creating a complex, layered visual effect.

Reactive Angular with RxJS @ngrx/store – Store DevTools

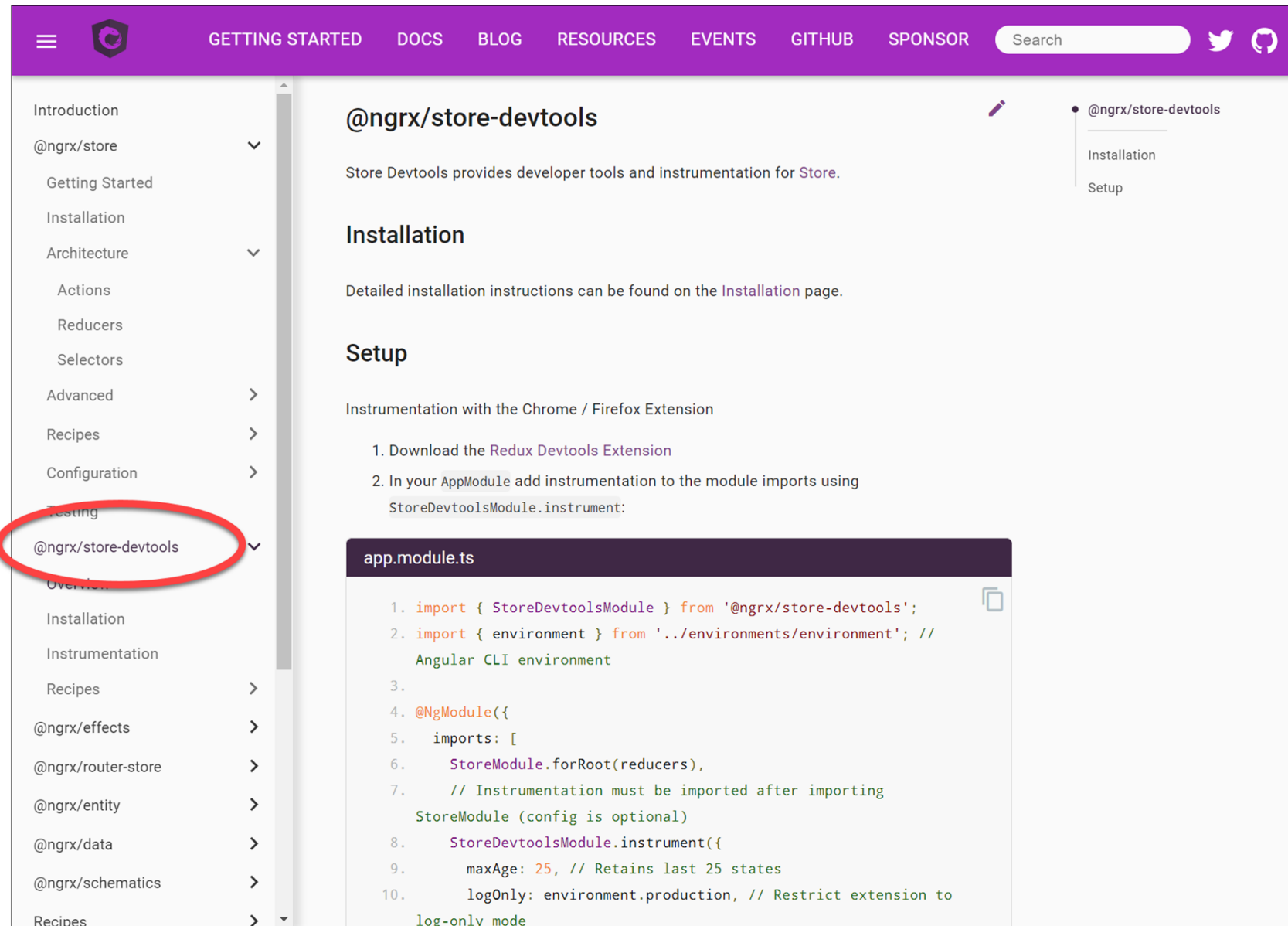
Peter Kassenaar –
info@kassenaar.com

What is @ngrx/store-devtools?

- Analyzing, inspection and debugging your State Management Store
- Time Travelling options



<https://ngrx.io/guide/store-devtools>



The screenshot displays the official documentation for @ngrx/store-devtools. The navigation sidebar on the left lists various guides, with '@ngrx/store-devtools' selected and circled in red. The main content area features a purple header with navigation links and a search bar. Below the header, the page title '@ngrx/store-devtools' is followed by a brief description: 'Store Devtools provides developer tools and instrumentation for Store.' The 'Installation' section states that detailed instructions are on the 'Installation' page. The 'Setup' section describes instrumentation with the Chrome / Firefox Extension and provides a two-step guide: 1. Download the Redux Devtools Extension, and 2. In your AppModule add instrumentation to the module imports using StoreDevtoolsModule.instrument(). A code block for 'app.module.ts' shows the implementation of these steps in TypeScript.

@ngrx/store-devtools

Store Devtools provides developer tools and instrumentation for Store.

Installation

Detailed installation instructions can be found on the [Installation](#) page.

Setup

Instrumentation with the Chrome / Firefox Extension

1. Download the [Redux Devtools Extension](#)
2. In your AppModule add instrumentation to the module imports using StoreDevtoolsModule.instrument:

```
app.module.ts

1. import { StoreDevtoolsModule } from '@ngrx/store-devtools';
2. import { environment } from '../environments/environment'; //
   Angular CLI environment
3.
4. @NgModule({
5.   imports: [
6.     StoreModule.forRoot(reducers),
7.     // Instrumentation must be imported after importing
     StoreModule (config is optional)
8.     StoreDevtoolsModule.instrument({
9.       maxAge: 25, // Retains last 25 states
10.      logOnly: environment.production, // Restrict extension to
        log-only mode
```

Two Parts

1. Chrome/Firefox Extension/Add-On

2. Installation in your project



1. Install in Chrome

One-time installation of Redux DevTools Extension



chrome web store



peter.kassenaar@gmail.com ▾

[Home](#) > [Extensions](#) > Redux DevTools



Redux DevTools

Offered by: remotedevio

★★★★★ 485 | [Developer Tools](#) | 815,121 users

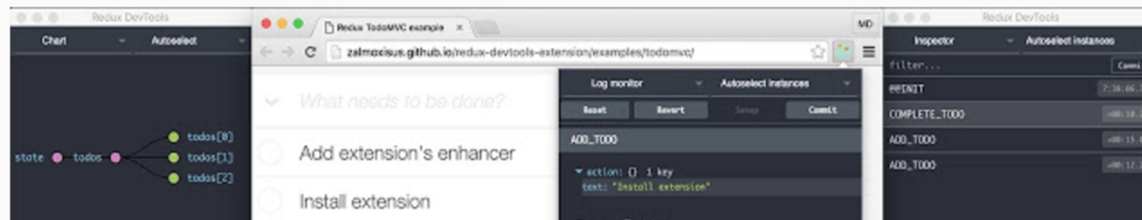
Remove from Chrome

Overview

Reviews

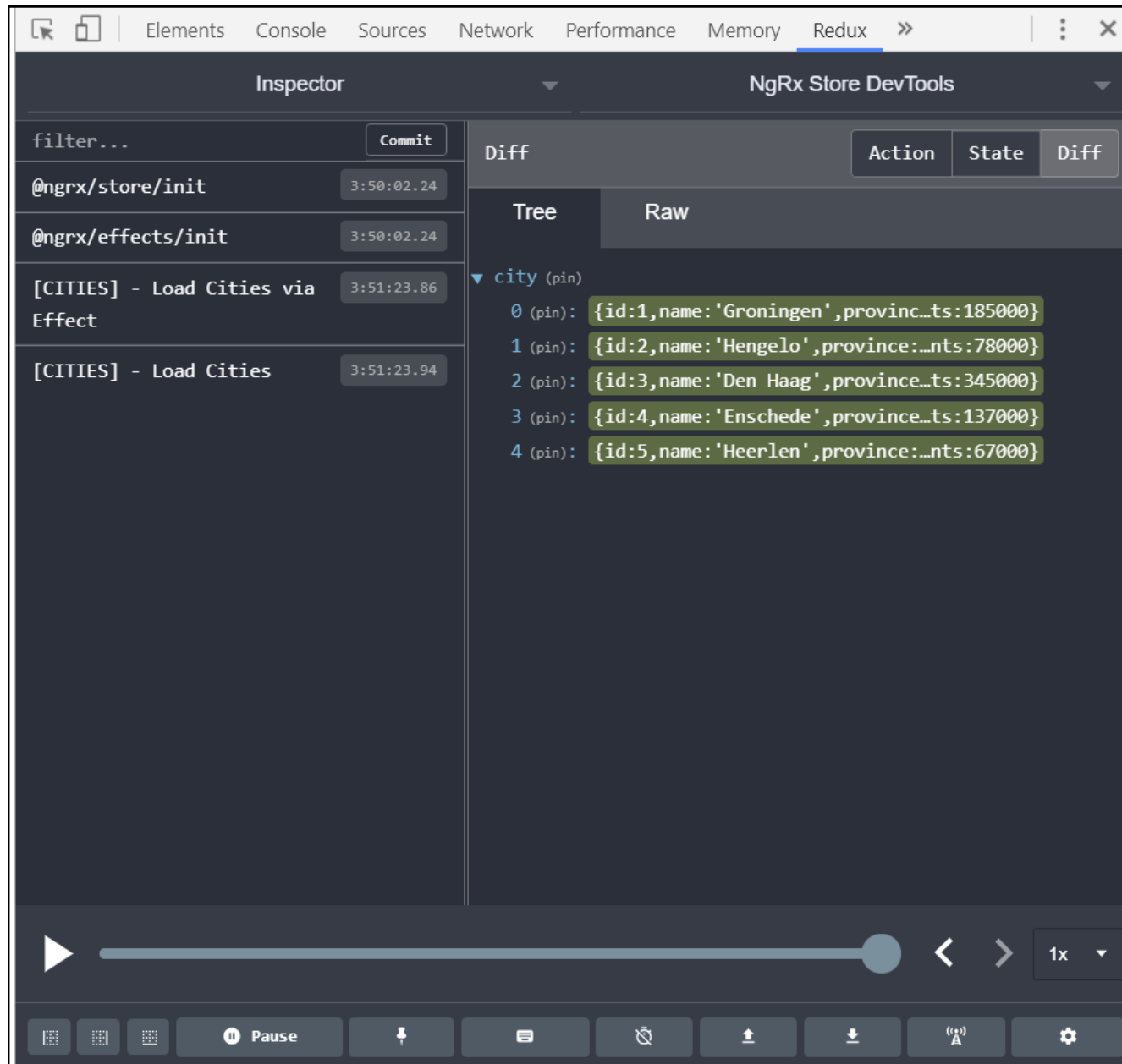
Support

Related



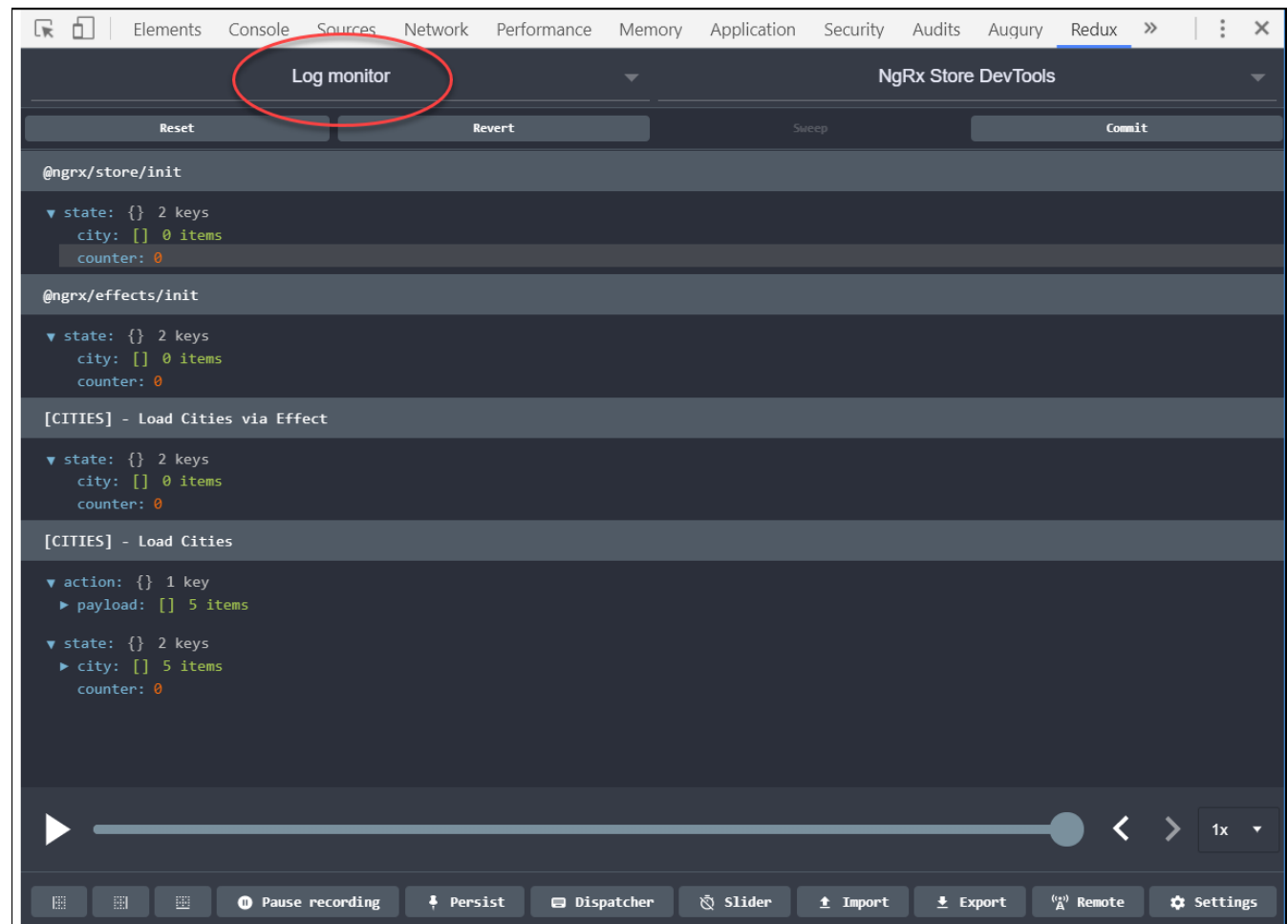
<https://chrome.google.com/webstore/detail/redux-devtools/lmhkpmbekcpmknkloeibfkpmmfiblj?hl=en>

Chrome Extension



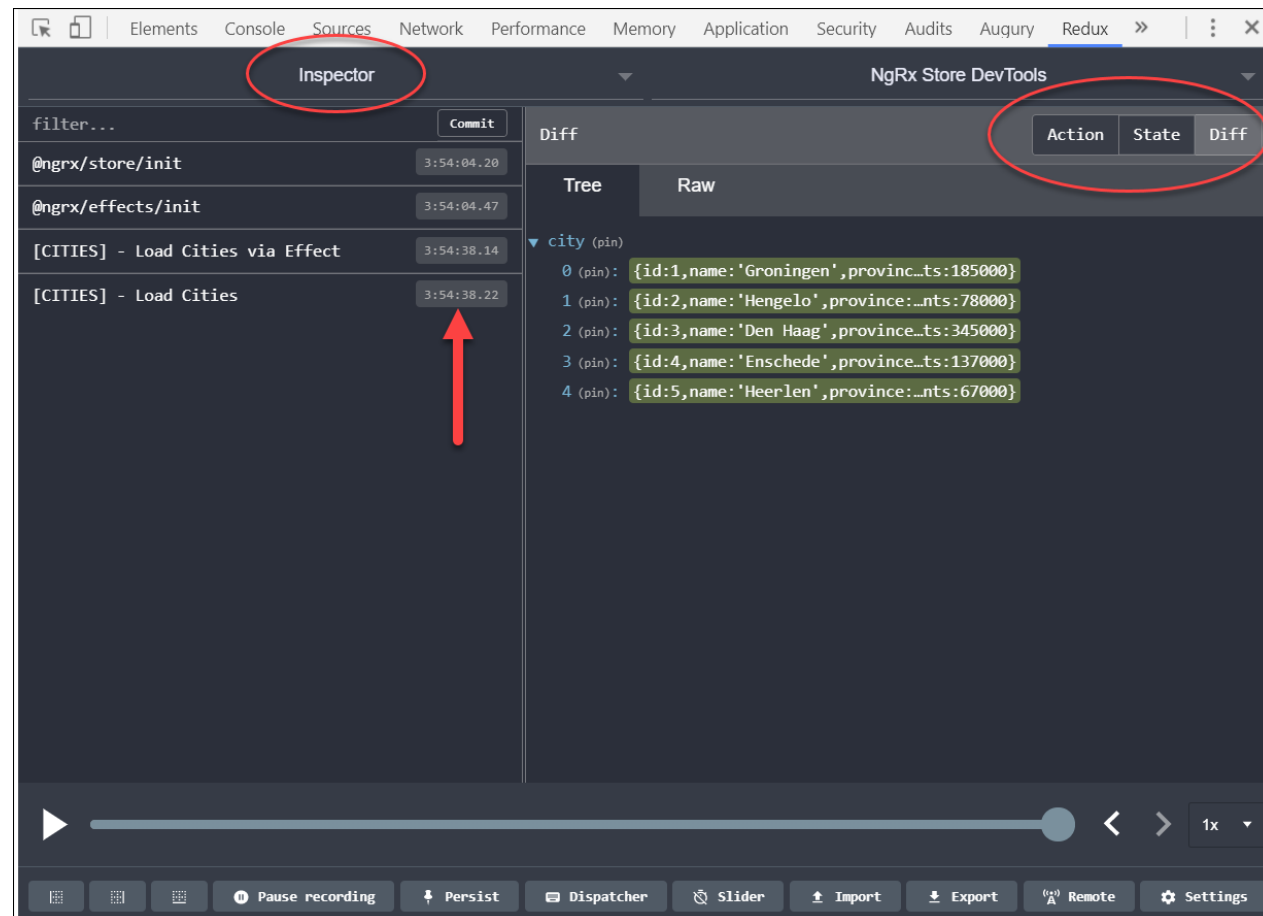
Options

- Log Monitor
 - Log actions from/to the store and see changes in Real Time
 - Reset/Revert actions



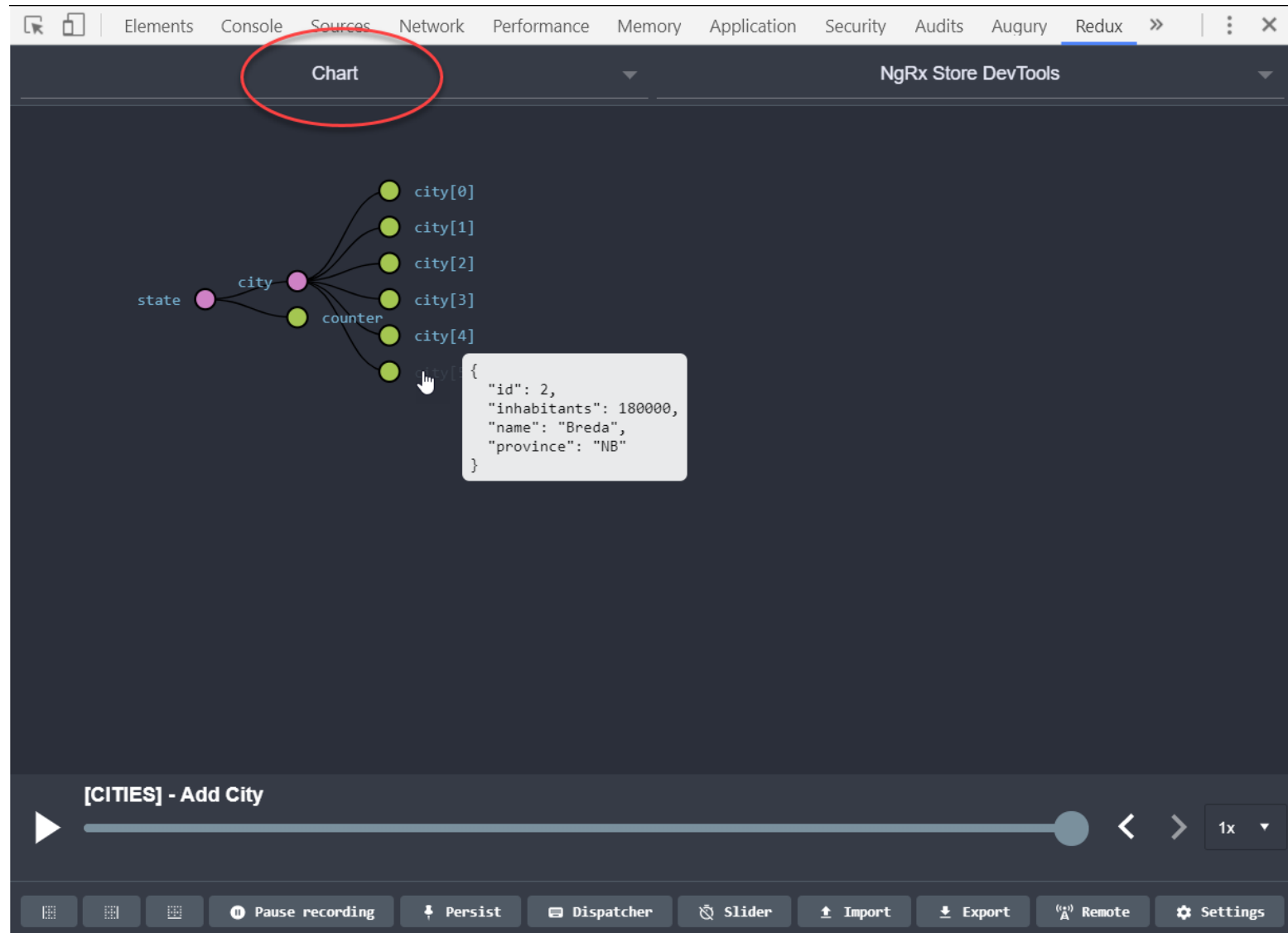
Inspector

- Filter Actions
- Jump/Skip actions
- Inspect Action, State or Diff compared to previous State



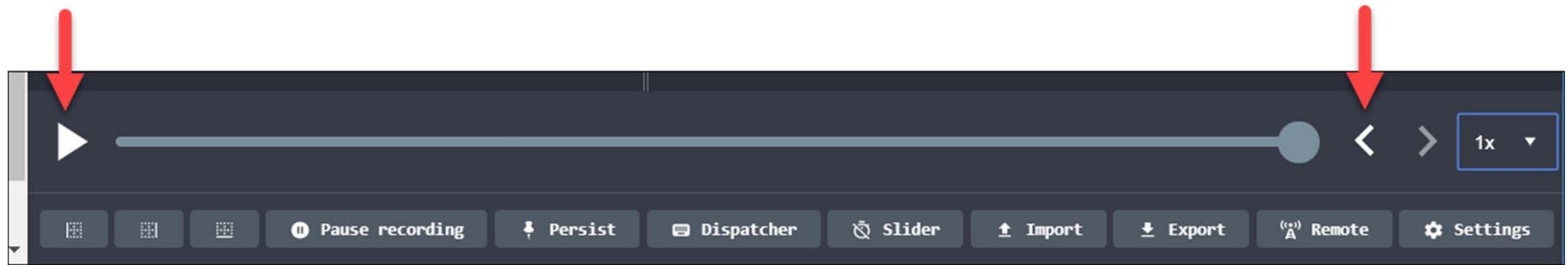
Chart

- Inspect flowchart of Actions, flow of data through the Store



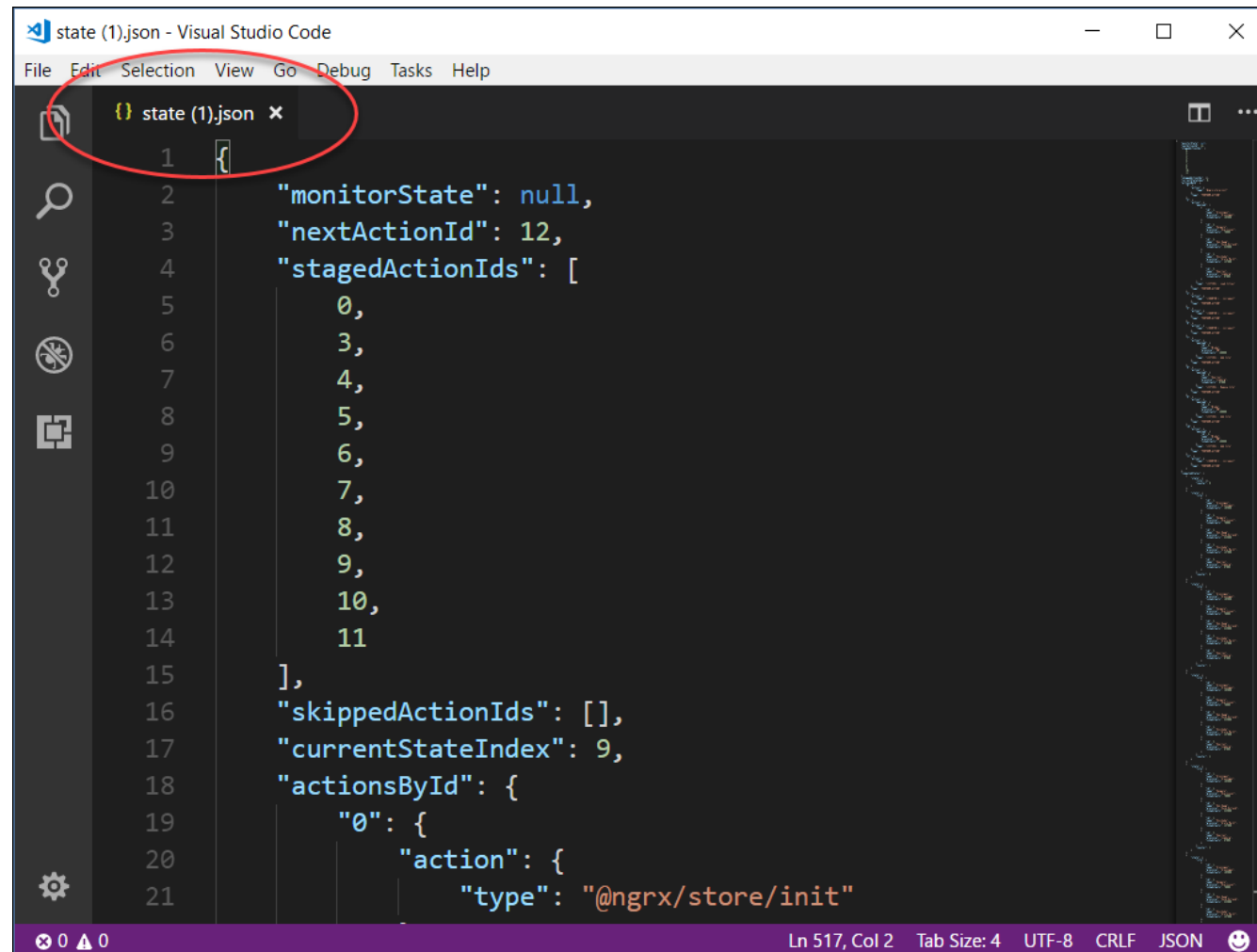
Slider

- (Re)play various actions and see changes in app and in state
- Jump to begin/end of timeline



Import/export State

- Save State changes to .json-file and import them to replay steps



```
state (1).json - Visual Studio Code
File Edit Selection View Go Debug Tasks Help

1 {
2   "monitorState": null,
3   "nextActionId": 12,
4   "stagedActionIds": [
5     0,
6     3,
7     4,
8     5,
9     6,
10    7,
11    8,
12    9,
13    10,
14    11
15  ],
16  "skippedActionIds": [],
17  "currentStateIndex": 9,
18  "actionsById": {
19    "0": {
20      "action": {
21        "type": "@ngrx/store/init"
```



2. Installation

Install in Google Chrome *and* in project

Installation in Project

```
npm install @ngrx/store-devtools --save
```

OR:

```
ng add @ngrx/store-devtools
```




A screenshot of a code editor showing a portion of a package.json file. The file lists several dependencies. A red rounded rectangle highlights the entry for "@ngrx/store-devtools": "^8.3.0".

```
18   "@angular/forms": "~8.2.9",  
19   "@angular/platform-browser": "~8.2.9",  
20   "@angular/platform-browser-dynamic": "~8.2.9",  
21   "@angular/router": "~8.2.9",  
22   "@ngrx/store": "^8.3.0",  
23   "@ngrx/store-devtools": "^8.3.0",  
24   "bootstrap": "^4.3.1",  
25   "jquery": "^3.4.1",  
26   "..." : "..."
```

Installation in app.module.ts

```
// app.module.ts
import {StoreDevtoolsModule} from '@ngrx/store-devtools';

@NgModule({
  ...
  imports      : [
    ...
    StoreDevtoolsModule.instrument({
      maxAge: 10}) // Redux Devtools debugging in Chrome
  ],
})
export class AppModule {
}
```



Install Redux DevTools

- Add from chrome://extensions

Redux DevTools
aangeboden door remotedev.io
★★★★★ (334) | [Ontwikkelaarstools](https://github.com/zalmoxisus/redux-devtools-extension) | 351.955 gebruikers

TOEGEVOEGD AAN CHROME

OVERZICHT RECENSIES ONDERSTEUNING GERELATEERD

Compatibel met je apparaat

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.

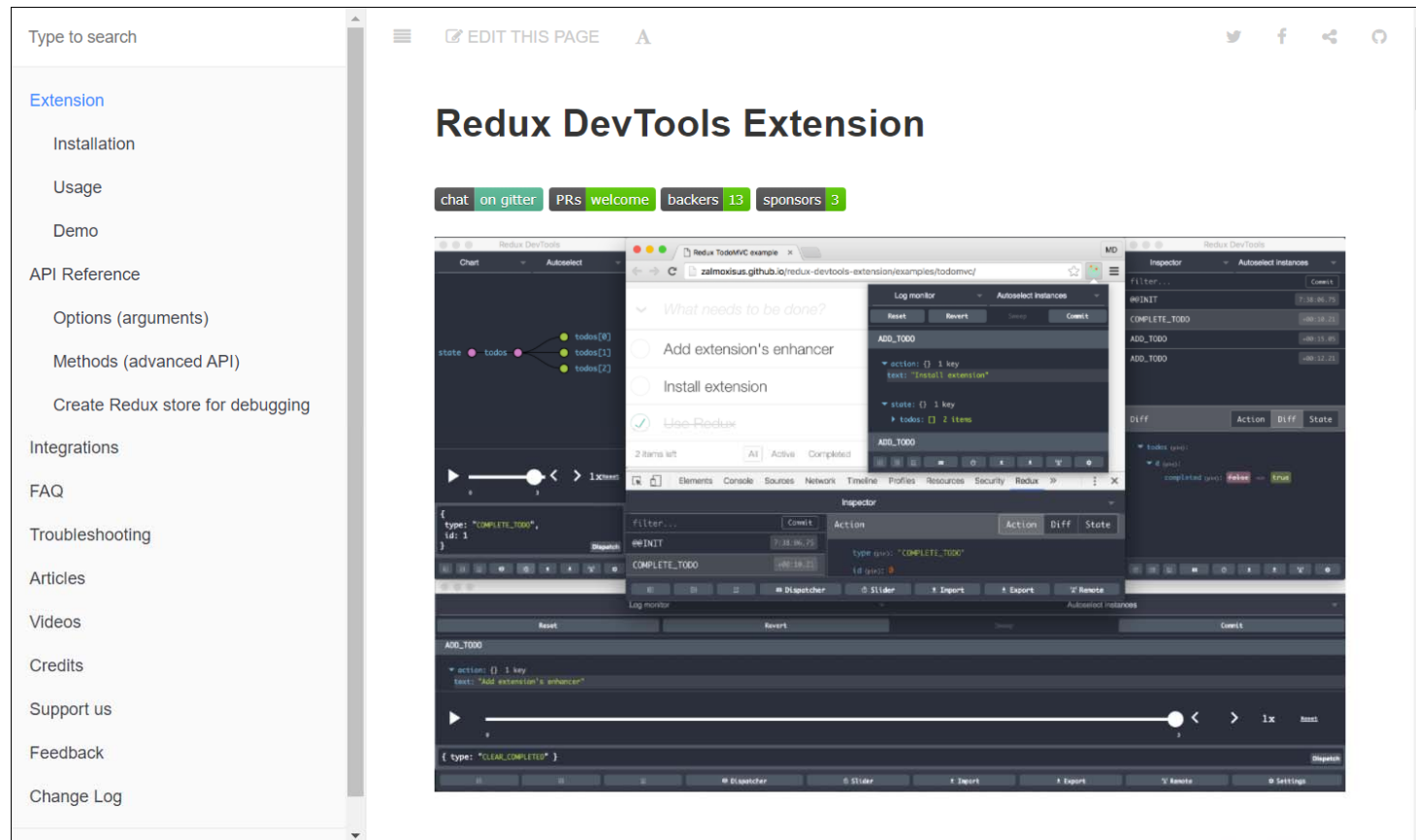
It's an opensource project. See the official repository for more details:
<https://github.com/zalmoxisus/redux-devtools-extension>

[Website](#)
[Misbruik melden](#)

Aanvullende informatie
Versie: 2.15.1
Laatst bijgewerkt: 7 mei 2017
Grootte: 1.54MIB
Taal: English

More info

- <http://extension.remotedev.io/>
- NOT specific to Angular – to be used with all Redux-like stores!



Workshop

- Use your own project, or start from one of the Store projects, for example `/200-ngrx-simple-store`
- Add Redux DevTools extension to Chrome (one-off action)
- Add `@ngrx/store-devtools` to current project via `npm` or `ng add`
- Activate devtools in `app.module.ts`
- Experiment with different views
 - Log Monitor
 - Inspector
 - Chart
- Export a State Tree to `.json` file, reload the project and import it again. See how this works

