



Redux

Gold - Chapter 7 - Topic 4

**Selamat datang di Chapter 7 Topik 4 online
course Fullstack Web dari Binar Academy!**

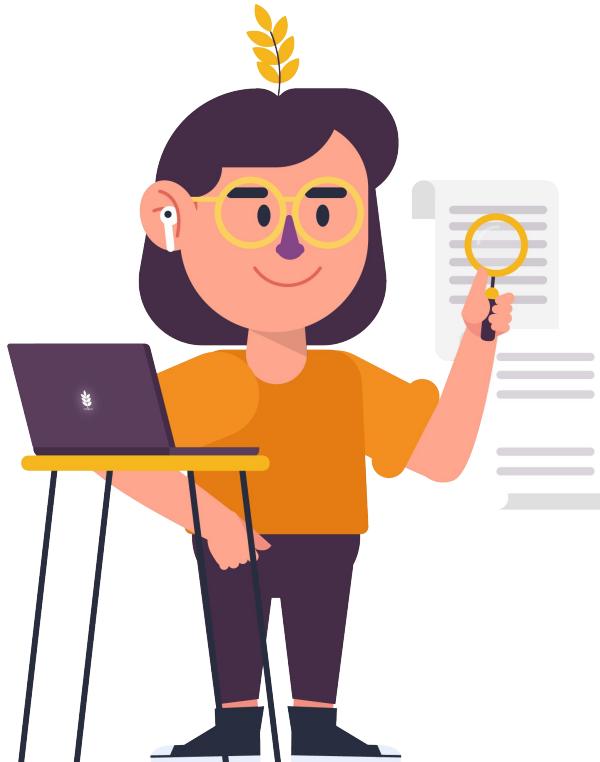


Buah kendondong buah alpukat, selamat datang di Topik 4! 🎉

Pada topik sebelumnya kita sudah mempelajari konsep Oauth hingga implementasi google oauth pada ExpressJs dan ReactJS.

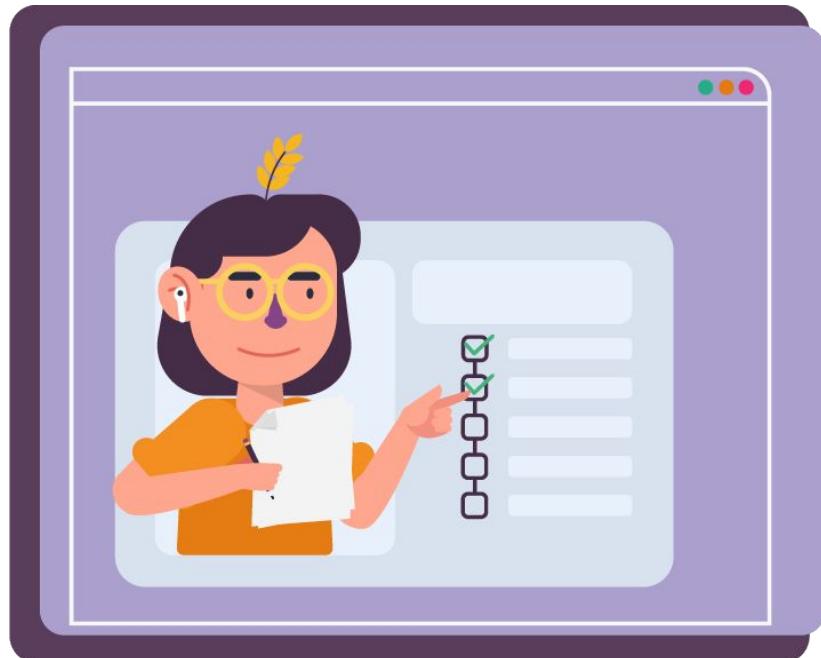
Di topik keempat ini, kita bakal bahas tentang **apa itu Redux. Mulai dari pengenalan State management, Redux, Redux Tool chain, React Redux dan Redux Thunk (Middleware).**

Markicuss~



Detailnya, kita bakal bahas hal-hal berikut ini:

- Pemahaman dasar State Management
- Memahami fungsi redux
- Cara menggunakan redux tool chain
- Implementasi Redux dalam ReactJS
- Mengenal Redux Thunk (Middleware)





Pertama, kita perlu bahas konsep nih. Apa sih sebenarnya **state management** maupun **Redux** itu? Yuk kita kupas tuntas satu persatu!



Kenalan sama State Management kuy~

State management adalah **sebuah system yang mengelola perubahan state di dalam suatu aplikasi**. Salah satu state management library yang terkenal adalah **redux**. Pada materi ini kita bakal bahas gimana cara pake redux.

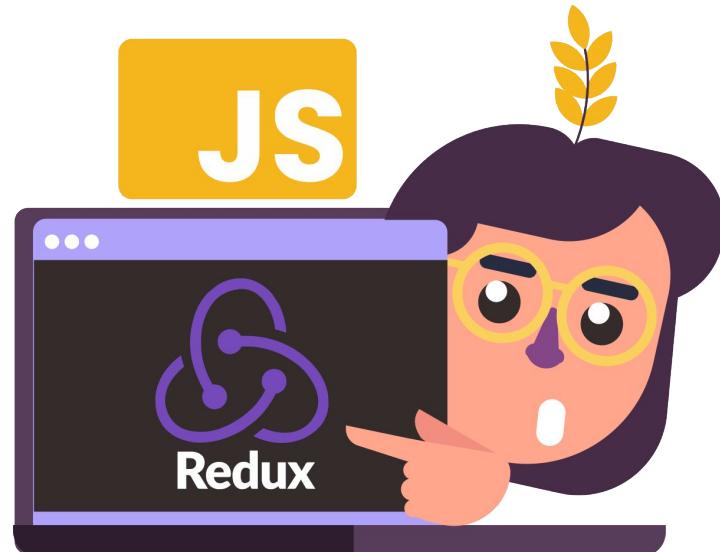




Redux

Redux adalah **state container untuk aplikasi JavaScript**.

Meskipun pada umumnya Redux digunakan bersama dengan React, tapi pada dasarnya Redux juga bisa digunakan untuk framework JavaScript yang lain.

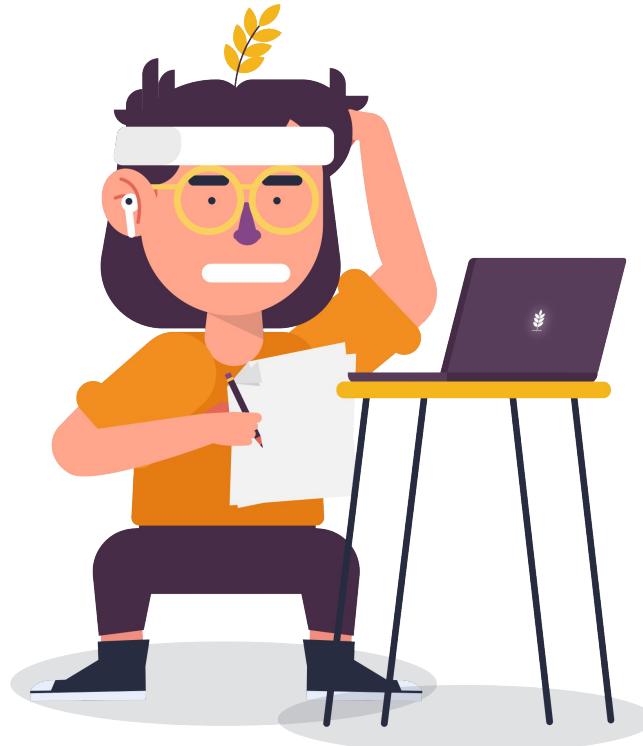




Mengapa Redux?

Mengatur hubungan antara state dengan component pada React dapat dilakukan dengan hanya menggunakan state dan props. Tetapi pada aplikasi yang memiliki banyak sekali component mengelola state bisa menjadi sedikit rumit.

Sebuah state bisa digunakan oleh banyak sekali component, seringkali state harus dipindah ke parent component (lifting up) agar state tersebut dapat digunakan oleh component lain.





Ketika menggunakan React dengan Redux kita tidak perlu lagi memindahkan state dari satu component ke component lain, karena Redux **mengubah state menjadi global state dan menempatkannya pada suatu tempat bernama store**. Setiap component dapat menggunakan state yang ada di store secara langsung.

Semua modifikasi state juga dilakukan melalui Redux. Store pada Redux menjadi satu-satunya tempat untuk mengelola state (single source of truth).

Aplikasi menjadi lebih konsisten dan mudah untuk dilakukan test.





Lantas, haruskah kita pakai Redux?

Jawabannya tidak wajib. **Meskipun pengelolaan state dilakukan di satu tempat bukan berarti mengelola state menjadi lebih mudah.**

Redux sendiri menggunakan pattern yang kompleks dan butuh waktu untuk dipahami dan menguasainya.

Mengubah state menggunakan Redux harus melewati beberapa proses, hal ini berakibat tingkat kerumitan dari aplikasi bisa bertambah.

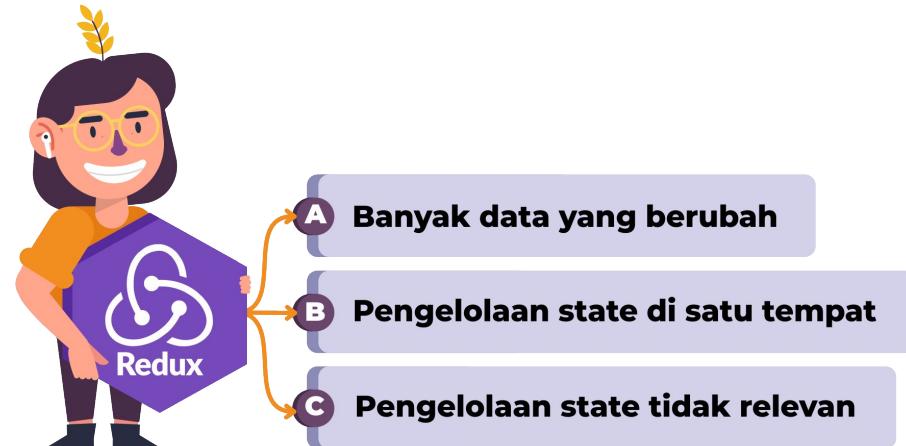
Hmmm, terus kira-kira kapan ya waktu yang sesuai untuk memakai Redux?





Sesuai rekomendasi, kita dianjurkan menggunakan Redux jika :

- Banyak data yang berubah dari waktu ke waktu
- Pengelolaan state harus dilakukan di satu tempat (single source of truth)
- Pengelolaan state di top-level component sudah tidak lagi relevan



Istilah Konsep Dasar Redux

Paling tidak ada 5 istilah yang digunakan dalam Redux yang wajib untuk diketahui.

- **Actions**
- **Reducers**
- **Store**
- **Dispatch**
- **Selector**





Actions

Sebuah JavaScript Object mewakili apa yang terjadi di dalam aplikasi.

Reducers

Function yang menerima object state dan object action, bertugas menentukan bagaimana suatu state diubah. Output reducer adalah state baru.

Syntax: `(state, action) ⇒ newState`

Store

Tempat dimana global state disimpan.



Dispatch

Satu-satunya cara untuk mengubah state di dalam store adalah dengan memanggil method bernama dispatch yang berisi object action, kemudian Redux akan mengeksekusi reducer yang sesuai.

Selector

Function yang digunakan untuk mendapatkan data dari state yang ada di dalam store.





3 Konsep Dasar Redux

Single Source of Truth

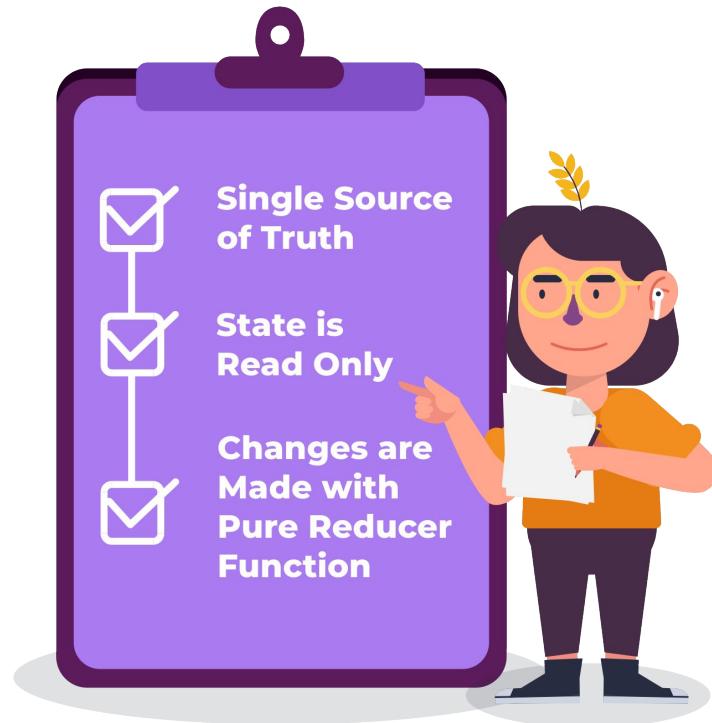
Redux menggunakan store sebagai single source of truth, dimana semua global state disimpan dalam bentuk object di dalam store.

State is Read Only

Untuk menghindari data diubah tanpa sengaja atau terhapus, state hanya bisa diubah dengan cara dispatch suatu action.

Changes are Made with Pure Reducer Function

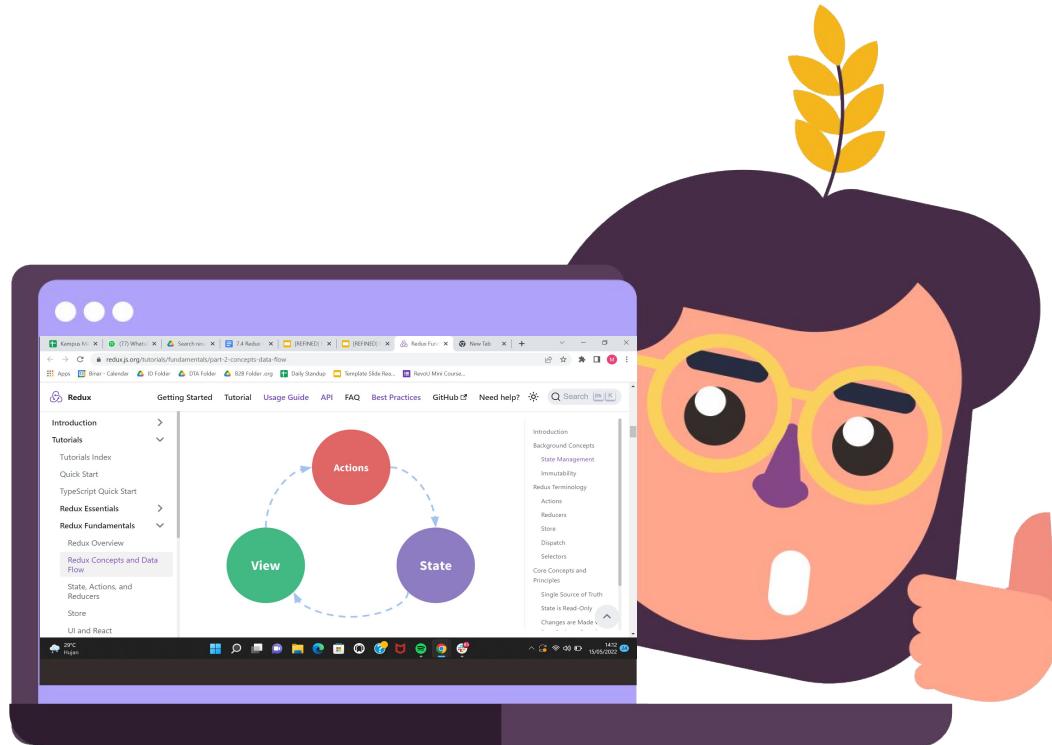
State diubah menggunakan pure function (Reducer), yaitu function menerima state sebelumnya dan sebuah action, kemudian menghasilkan state baru tanpa menghapus state sebelumnya.



Data Flow

Gambar disamping adalah alur data sederhana dari contoh aplikasi yang menggunakan Redux.

Contoh alur data ini diambil dari [halaman official](#) dokumentasi redux.





Udah mulai paham kan tentang redux?
Pastinya udah joss kan yaaa~

Selanjutnya Sabrina mau ngajakin kamu
buat memahami **Redux Toolchain**. Yuk
kita bahas aja langsung, kuy!



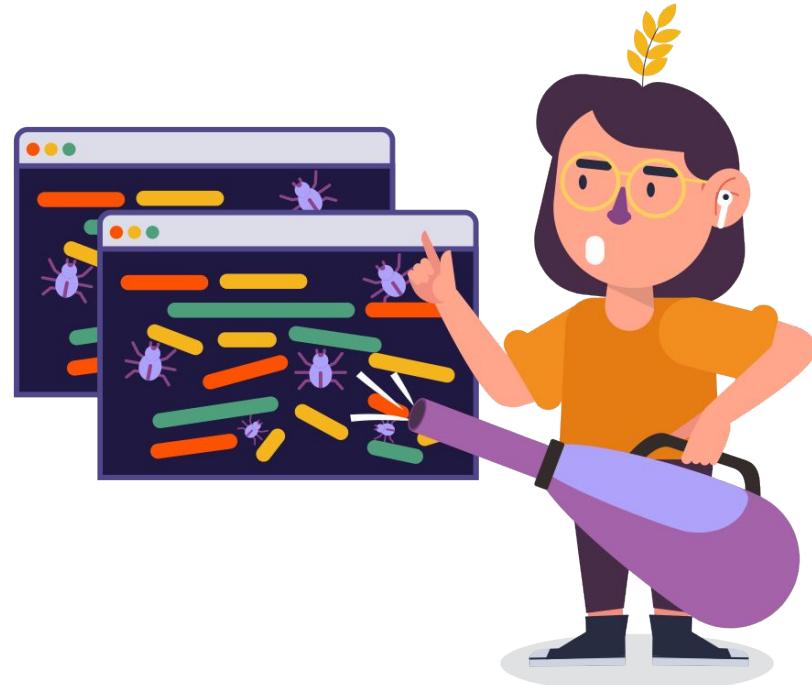


Redux Toolchain

Terdapat banyak redux toolchain, namun dalam sesi ini kita hanya akan bahas dan install satu hal, yaitu Redux Devtools.

Redux DevTools merupakan alat untuk memudahkan kita melakukan debugging pada aplikasi yang menggunakan Redux ataupun state management yang lain.

Untuk lebih jelasnya, mari kita lihat contohnya langsung saja~

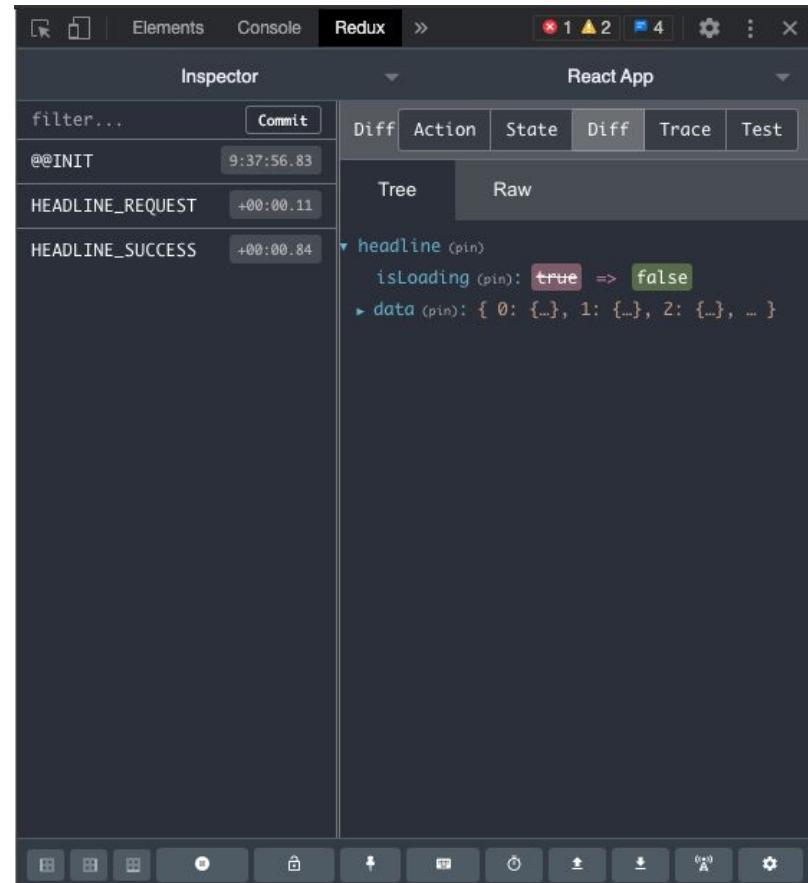




Gambar disamping adalah tampilan dari Redux DevTools.

Saat panel Redux DevTools pertama kali dibuka, pada sisi sebelah kiri panel kita dapat melihat informasi terkait action type apa saja yang telah dieksekusi di aplikasi kita. Sedangkan di sebelah kanan panel kita bisa mengetahui state pada aplikasi kita, termasuk perubahannya. Di screenshot di atas, state isLoading yang awalnya bernilai “true” berubah menjadi “false”. Selain perubahan state, masih banyak lagi informasi yang bisa kita dapatkan dari fitur-fitur yang disediakan Redux DevTools

Untuk lebih lengkapnya kamu bisa lihat [disini](#) ya!





Nahhhh ~

Ditopik awal chapter ini kita sudah belajar apa itu React kan? Sekarang, kira-kira gimana sih penggunaan **Redux** pada **React**, dan kita akan bahas juga apa itu **Redux Thunk**!





Untuk mulai menggunakan react dan redux, kalian tinggal generate aja project pake CRA dengan template redux.



```
yarn create react-app contoh-repo-pake-redux --template redux
```

Maka project yang di-generate akan seperti gambar di samping :

Kalian bisa examine file **src/features/counter/Counter.js** disitu terdapat implementasi redux yang perlu kalian tahu. Selengkapnya bisa kalian lihat [disini](#).



```
package.json
public
  favicon.ico
  index.html
  logo192.png
  logo512.png
  manifest.json
  robots.txt
README.md
src
  app
    store.js
  App.css
  App.js
  App.test.js
  features
    counter
      counterAPI.js
      Counter.js
      Counter.module.css
      counterSlice.js
      counterSlice.spec.js
    index.css
    index.js
    logo.svg
    reportWebVitals.js
  setupTests.js
yarn.lock

5 directories, 23 files
```

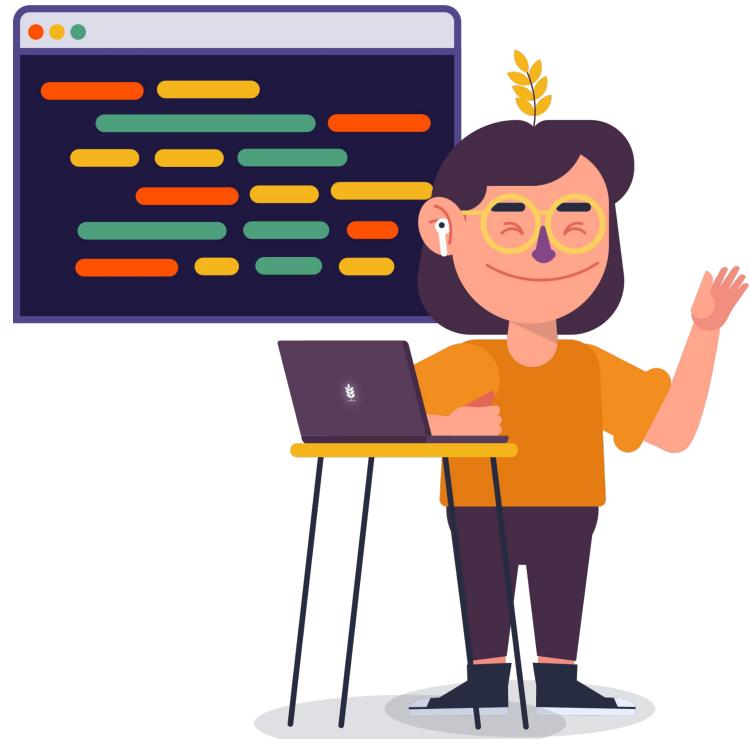


Redux Thunk

Redux thunk adalah **salah satu redux middleware yang by default adalah implementasi redux dari CRA dengan template redux.**

Redux thunk bekerja secara asynchronous dan dalam sistem action based, jadi setiap kali user ingin melakukan perubahan state, pasti perubahan state tersebut dinisiasi oleh sebuah action.

Detail implementasinya dapat kalian lihat pada repository diatas, atau dari artikel [berikut](#).



Referensi buat tambahan kamu belajar ~

- <https://devsaurus.com/redux-intro>
- <https://www.digitalocean.com/community/tutorials/redux-redux-thunk-id>
- <https://redux.js.org/introduction/ecosystem>



Terima Kasih!



Chapter ✓

completed