Redux 1.1_CW Exercises

introduction to redux concepts

ex00: setup

challenge

In the `**index.html**` file,

- Get a Cookie Button: Add a button with the text "Get a Cookie" with id `**add**`.
- Give Away a Cookie Button: Add a button with the text "Give Away a Cookie" with the id `remove`.
- Display Cookie Count: Add a h1 tag with id `**cookie-count**`.

https://codesandbox.io/s/rx1-1-cw-ex00-5h7ts9

- Solution

```
<button id="add">get a cookie</button>
<button id="remove">give away a cookie</button>

<h1>🍩 <span id="cookie-count"></span></h1>
```

COPY

ex01: create a reducer

understanding

Imagine you have a magic box that holds the number of cookies you have. This magic box is the state of your cookie count.

Now, we need a plan for what happens when you get more cookies or give away some. The reducer is that plan! We write a function that takes the current number of cookies and the action, and it gives us a new number of cookies.

That's how we change our state in a predictable way.

```
const initialState = { value: 0 }

const cookiesReducer = (state = initialState, action) => {
  switch (action.type) {
    case 'cookies/added':
      return { value: state.value + 1 }
    default:
      return state
  }
}

export default cookiesReducer
```

COPY

challenge

Create a cookies reducer, to add & remove a cookie in `cookiesReducer.js`.

https://codesandbox.io/s/rx1-1-cw-ex01-jgj2w9

solution

```
const initialState = { value: 0 }

const cookiesReducer = (state = initialState, action) => {
  switch (action.type) {
    case 'cookies/added':
      return { value: state.value + 1 }
    case 'cookies/removed':
      return { value: state.value - 1 }
    default:
      return state
  }
}

export default cookiesReducer
```

COPY

ex02: create a store

understanding

The store is like a manager that keeps track of our app's state changes. It's responsible for making sure everything happens in the right order.

In our case, the store keeps track of our state, which is the number of cookies. We tell the store how our state should change using the reducer we made earlier.

challenge

Inside `index.js`, import the required `createStore` function from Redux and `**cookiesReducer**`.

Create a Redux store by passing your `**cookiesReducer**` to the `**createStore**` function.

https://codesandbox.io/s/rx1-1-cw-ex02-ty8ckz

💡 Make sure to add Redux as a dependency

solution

```
// index.js

import { createStore } from 'redux'
import cookiesReducer from './cookiesReducer'

const store = createStore(cookiesReducer)
```

COPY

ex03: subscribe to the store

understanding

Subscribing lets us know whenever something changes in our app's state. It's like getting a message whenever someone takes or adds a cookie to our cookie jar.

challenge

Use the `subscribe` method of the Redux store to listen for state changes and log the state.

https://codesandbox.io/s/rx1-1-cw-ex03-8nmhn9

solution

```
store.subscribe(() => console.log(store.getState()))
```

COPY

ex04: dispatch actions

understanding

Dispatching actions is like telling our app what we want to do. It's like giving commands to our app to make things happen.

When we dispatch actions, we're telling our app what we want to do. We say, "Add a cookie" by dispatching an action that the store understands. The store then follows the plan with the help of reducer and changes the number of cookies accordingly.

```
store.dispatch({ type: 'cookies/added' })
```

COPY

challenge

Use the `dispatch` method of the Redux store to dispatch actions. Add 2 cookies & then remove 1.

https://codesandbox.io/s/rx1-1-cw-ex04-q2ql5t

solution

```
store.dispatch({ type: 'cookies/added' })
store.dispatch({ type: 'cookies/added' })
store.dispatch({ type: 'cookies/removed' })
```

COPY

ex05: i**nteracting with the cookie counter**

challenge

`index.html` contains a "get a cookie" button, a "give away a cookie" button, and a text element displaying the current count of cookies. Your task is to write javascript code that interacts with these elements with `document.getElementById`

https://codesandbox.io/s/rx1-1-cw-ex05-g2mcym

solution

```
const addCookie = document.getElementById('add')
const removeCookie = document.getElementById('remove')
const cookieCount = document.getElementById('cookie-count')
```

COPY

ex06: u**pdating the cookie count**

challenge

Implement event handlers that will dispatch these actions when interacting with the "get a cookie" and "give away a cookie" buttons. Make sure you comment the dispatch actions.

1. Create two functions named `addCookieHandler` and `removeCookieHandler`. Inside these functions, dispatch the respective actions (`cookies/added` and `cookies/removed`) to the Redux store.
2. Attach event listeners to the "get a cookie" and "give away a cookie" buttons (`addCookie` and `removeCookie`) using the `addEventListener` method. These event listeners should call the `addCookieHandler` and `removeCookieHandler` functions when the buttons are clicked.

https://codesandbox.io/s/rx1-1-cw-ex06-tvvkdg

solution

```
// store.dispatch({ type: "cookies/added" });
// store.dispatch({ type: "cookies/added" });
// store.dispatch({ type: "cookies/removed" });

const addCookieHandler = () => {
  store.dispatch({ type: 'cookies/added' })
}

const removeCookieHandler = () => {
  store.dispatch({ type: 'cookies/removed' })
}

addCookie.addEventListener('click', addCookieHandler)
removeCookie.addEventListener('click', removeCookieHandler)
```

COPY

ex07: displaying the cookie count

challenge

Your task is to write javascript code that will display the cookie count and update it whenever the state changes.

1. Create a function `updateCookieCount` that will be responsible for updating the displayed cookie count on the webpage. Inside this function, get the current state from the Redux store using `store.getState()`, and update the content of the `cookieCount` element with the cookie count value from the state.
2. Inside the `store.subscribe` method, after the log, call the function `updateCookieCount`
3. Call the `updateCookieCount` function after defining it to ensure that the initial cookie count is displayed correctly on page load.

https://codesandbox.io/s/rx1-1-cw-ex07-ywxj7j

solution

```
store.subscribe(() => {
  console.log(store.getState())
  updateCookieCount()
})
```

```
const updateCookieCount = () => {
  const state = store.getState()
  cookieCount.textContent = state.value
}

updateCookieCount()
```

COPY

entire solution

https://codesandbox.io/s/rx1-1-cw-entire-solution-vzhfcs