Redux 2.2_CW Exercises

image

ex01: create actions

challenge

In a file called `actions.js`, define the actions for adding income and expenses with amount.

solution

```
export const addIncome = (amount) => {
  return {
    type: 'ADD_INCOME',
    payload: amount,
  }
}

export const addExpense = (amount) => {
  return {
    type: 'ADD_EXPENSE',
    payload: amount,
  }
}
```

COPY

ex02: create reducers

challenge

In a file called `reducers.js`, create reducers to manage income and expenses.

solution

```
const initialState = {
  income: 0,
  expenses: 0,
}

const financeReducer = (state = initialState, action) => {
  switch (action.type) {
    case 'ADD_INCOME':
      return {
        ...state,
```

```
        income: state.income + action.payload,
      }
    case 'ADD_EXPENSE':
      return {
        ...state,
        expenses: state.expenses + action.payload,
      }
    default:
      return state
  }
}

export default financeReducer
```

COPY

ex03: create a redux store

challenge

In a file called `store.js`, create the Redux store and connect it to the reducer.

solution

```
import { createStore } from 'redux'
import financeReducer from './reducers'

const store = createStore(financeReducer)

export default store
```

COPY

ex04: create a `IncomeExpenseForm` component

challenge

Create a React component for your app. You can create an `IncomeExpenseForm.js` component to input income and expenses.

solution

```
import React, { useState } from 'react'

function IncomeExpenseForm() {
  const [amount, setAmount] = useState(0)

  return (
    <div>
      <input
```

```
        type='number'
        value={amount}
        onChange={(e) => setAmount(e.target.value)}
      />
      <button>Add Income</button>
      <button>Add Expense</button>
    </div>
  )
}

export default IncomeExpenseForm
```

COPY

ex05: create a `FinanceSummary` component

challenge

You can create a `FinanceSummary.js` component to display the current financial status.

solution

```
import React from 'react'

function FinanceSummary() {
  return (
    <div>
      <h2>Finance Summary</h2>
      <p>Income: $100</p>
      <p>Expenses: $50</p>
      <p>Balance: $50</p>
    </div>
  )
}

export default FinanceSummary
```

COPY

ex06: connect components to redux

challenge

In your components that need access to the Redux store, use the `useSelector` and `useDispatch` hooks to connect them. For example, in `IncomeExpenseForm.js`

solution

```
// IncomeExpenseForm.js
import React, { useState } from 'react'
```

```
import { useDispatch } from 'react-redux'
import { addIncome, addExpense } from './actions'

function IncomeExpenseForm() {
  const dispatch = useDispatch()
  const [amount, setAmount] = useState(0)

  const handleAddIncome = () => {
    dispatch(addIncome(parseFloat(amount)))
    setAmount(0)
  }

  const handleAddExpense = () => {
    dispatch(addExpense(parseFloat(amount)))
    setAmount(0)
  }

  return (
    <div>
      <input
        type='number'
        value={amount}
        onChange={(e) => setAmount(e.target.value)}
      />
      <button onClick={handleAddIncome}>Add Income</button>
      <button onClick={handleAddExpense}>Add Expense</button>
    </div>
  )
}

export default IncomeExpenseForm
```

COPY

ex07: display finance summary

challenge

In your `FinanceSummary.js` component, use the `useSelector` hook to display the current income and expenses.

solution

```
import React from 'react'
import { useSelector } from 'react-redux'

function FinanceSummary() {
  const income = useSelector((state) => state.income)
  const expenses = useSelector((state) => state.expenses)

  return (
    <div>
      <h2>Finance Summary</h2>
      <p>Income: ${income}</p>
      <p>Expenses: ${expenses}</p>
```

```
      <p>Balance: ${income - expenses}</p>
    </div>
  )
}

export default FinanceSummary
```

COPY

ex08: connect app component

challenge

In your main `App.js` component, wrap your components with the `Provider` and render them.

solution

```
import React from 'react'
import { Provider } from 'react-redux'
import store from './store'
import IncomeExpenseForm from './IncomeExpenseForm'
import FinanceSummary from './FinanceSummary'

function App() {
  return (
    <Provider store={store}>
      <div className='App'>
        <IncomeExpenseForm />
        <FinanceSummary />
      </div>
    </Provider>
  )
}

export default App
```

COPY

entire solution

https://codesandbox.io/s/redux-without-toolkit-with-react-rx2-forked-xdckzn