

CMPS 356 – Fall 2022

Web Applications Design and Development

Lab 10

Store Usage and Data Synchronization

Objective

1. Slices and stores using Zustand
2. Server data synchronization using React Query

Prerequisites

1. Next API Endpoints: <https://nextjs.org/docs/api-routes/introduction>
2. React Query: <https://tanstack.com/query/v4>
3. Zustand: <https://github.com/pmndrs/zustand>

1. Stores and Slices using Zustand

1. Create a new directory 01-cart-store and reuse the code from the previous lab.
2. Install Zustand: “npm install zustand”.
3. Replace the context with a store using Zustand.
 - 3.1. Create a store with an initial state and a single dispatch function. This can be achieved using the redux middleware as well.
 - 3.2. Replace the dispatch function with multiple functions corresponding to the different action types.
4. Update your cart items to include the product price.
5. Add a computed property to return the number of items in the cart.
6. Add a computed property to return the price of all items in the cart.
7. Split the store into two separate slices: one for the list of products and another for the items in the cart.
8. Use the persist middleware to save the cart between sessions.

2. Mutations using React Query

1. Create a new directory 02-server-state and reuse the code from the previous exercise.
2. Install React Query: “npm install @tanstack/react-query”.
3. Serve the products through an API endpoint and fetch them in your store page.
4. Add an API endpoint to regenerate the list of products and allow the user to manually invoke it.
5. Add a mutation to keep the list of products in sync between the client and server.