HAZZA INSTITUTE OF TECHNOLOGY

iqracity

MERN – ES6 + React

Module 16: Redux - II

Outline

Module 16

HAZZA INSTITUTE OF TECHNOLOGY

- Understanding of Reducer Functions
- Understanding of Working with Stores
- Understanding of Combining Reducers
- Understanding of Integration with React and Asynchronous Operations

HAZZA INSTITUTE OF TECHNOLOGY

Introduction to Redux Continued…

Reducer Functions

HAZZA INSTITUTE OF TECHNOLOGY

- Follows the pattern of the **reduce** function available on the **Array** prototype in JavaScript
- Receives the current state and an action, the function produces a new state based upon the type of action, and its associated data
- Pure function – output results from inputs only, no side-effects
- Should be configured to create an initial state during the first run

Working with Stores

HAZZA INSTITUTE OF TECHNOLOGY

- Stores are the container for applying the action to the state using the reducer function, and they contain the current state
- Created with the **createStore** function
- The first parameter is the reducer function
- The second parameter is an optional initial state, if this is not provided the default state initialized by the reducer function will be used on the first run-through

Store Initialization

HAZZA INSTITUTE OF TECHNOLOGY

- When a store is created, the reducer function is executed with no action allowing the default parameter value and the reducer functions to initialize the application state
- If an initial state is passed into the **createStore** function, the initial state is passed in when the store is created

Handling Actions with the Store

HAZZA INSTITUTE OF TECHNOLOGY

- Actions are sent to the store using the **dispatch** function
- The **dispatch** function accepts the action object as an argument
- The action object must have a **type** property to identify what the action is, additional properties with other relevant data may be specified as well

Distributing the New State

HAZZA INSTITUTE OF TECHNOLOGY

- To distribute the new state produced from a dispatched action, a publisher/subscriber model is used through a **subscribe** function available on the store
- When actions are dispatched, they are processed by the reducer producing a new state, then all of the subscriber functions are invoked so they can process the new state
- The new state is retrieved in the subscriber function through the **getState** function on the store

Combining Reducers

HAZZA INSTITUTE OF TECHNOLOGY

- The state tree for an application can grow quickly especially when considering the first principle of Redux which is the entire state of the application is stored in one object
- Writing a single reducer function for the whole state tree results in a long, bloated and difficult to maintain function

Combining Reducers

HAZZA INSTITUTE OF TECHNOLOGY

- Commonly, reducer functions will be divided into multiple reducer functions with each function being responsible for one branch of the state tree
- Redux provides a **combineReducers** function to combine these multiple reducer functions into a single function for the store

Integration with React and Asynchronous Programming

HAZZA INSTITUTE OF TECHNOLOGY

- Redux works great with React, but Redux is not limited to only working React
- Nevertheless, the React/Redux combination is so popular there are special libraries for tying the two together and there are lots of resources online which explore this common combination of libraries
- Asynchronous programming introduces additional complexities to managing state

Integration with React

HAZZA INSTITUTE OF TECHNOLOGY

- This course will not focus on the **react-redux** available here: https://github.com/reactjs/react-redux
- Instead, this course will connect Redux into React using appropriate coding patterns
  - The store will be passed in as property to the top level state component of the React UI
  - React component lifecycle functions will be used to interact with the store's functionality

Asynchronous Programming

HAZZA INSTITUTE OF TECHNOLOGY

- Asynchronous Redux programming appears difficult at first, but really its quite easy
- The key is to understand how state and asynchronous operations work together
- Asynchronous operations have two states:
  - Pending Request State
  - Fulfilled Request State

Conclusion

HAZZA INSTITUTE OF TECHNOLOGY

- Redux, inspired by Flux, improves the management of state in JavaScript applications
- Its built on three principles: single source of truth, immutable state, and pure reducer functions
- Redux provides the container for applying the actions to produce new states based upon the logic of the reducer functions
- Reducer functions can work on different parts of the state tree and be combined together

Summary

Module 16

HAZZA INSTITUTE OF TECHNOLOGY

- Understanding of Reducer Functions
- Understanding of Working with Stores
- Understanding of Combining Reducers
- Understanding of Integration with React and Asynchronous Operations