HAZZA INSTITUTE OF TECHNOLOGY

iqracity

MERN – ES6 + React

Module 15: Redux

Outline

Module 15

HAZZA INSTITUTE OF TECHNOLOGY

- Redux foundation
- Understanding of Why use Redux?
- Understanding of Three Principles of Redux
- Understanding of Essential JavaScript Concepts

HAZZA INSTITUTE OF TECHNOLOGY

Introduction to Redux

HAZZA INSTITUTE OF TECHNOLOGY

- Why Redux?
- Three Principles of Redux
- Essential JavaScript Concepts
- Reducer Functions
- Working with Stores
- Combining Reducers
- Integration with React and Asynchronous Operations

Why Redux?

HAZZA INSTITUTE OF TECHNOLOGY

- Managing state in JavaScript applications is very challenging
- Redux employs a predictable state container to simplify state management
- Execution of an application is an initial state followed by a series of actions

Why Redux?

HAZZA INSTITUTE OF TECHNOLOGY

- Each action reduces the state to a new predictable state, to which the application user interface transitions
- A state container, known as store, contains the reduction logic implemented as pure functions as well as the last reduced (current) state

Three Principles of Redux

- To enable state changes to be predictable, the following constraints applied to state changes
  - Single Source of Truth
  - State is Read-Only
  - Changes are made with Pure Functions

Single Source of Truth

HAZZA INSTITUTE OF TECHNOLOGY

- Following the pattern of Flux, all data flows through a Redux system in a unidirectional matter
- All changes to the state comes from actions applied to the state, and all actions are funneled into Redux
- No part of the system can ever receive data from two sources
- Additionally, the state managed by Redux is the state of the whole application (with minor exceptions, such as form control entry)

State is Read-Only

HAZZA INSTITUTE OF TECHNOLOGY

- State can never be mutated
- New states are produced by applying an action to the current state (known as reduction) from which a new state object is produced
- Immutable programming techniques need to be utilized

HAZZA INSTITUTE OF TECHNOLOGY

Changes are made with Pure Functions

- Pure functions accept inputs, and using only those inputs produce a single output
- The function produces no side effects
- Many pure functions can be composed together to process different parts of the state tree

Definition of Redux

HAZZA INSTITUTE OF TECHNOLOGY

- From the Redux website, "Redux is a predictable state container for JavaScript apps."
- Predictable – state changes follow the three principles
- State – the application's data, including data related to the UI itself
- Container – Redux is the container which applies actions to the pure reducer functions to return a new state
- Redux has been designed for JavaScript applications

Differences from Flux

HAZZA INSTITUTE OF TECHNOLOGY

- While Redux and Flux share similar concepts and principles, there are some differences
- Flux differentiates between the dispatcher and store, this is because Flux supports multiple stores
- Redux limits the application to one store which means the store and dispatcher can be combined into one dispatcher-store
- This dispatcher-store is created by Redux's **createStore** function

Development Environment

HAZZA INSTITUTE OF TECHNOLOGY

- Visual Studio Code + Google Chrome – using the Chrome extension for Visual Studio Code, in editor debugging of TypeScript code will be available
- REST Server provided by json-server, Web Server provided by browser-sync
- TypeScript is used for module support and strong-typing

Development Environment

- Dynamic Module Loading with SystemJS
- ES2015 code is not transpiled to ES5.1, it will run as ES2015 natively
- Node.js powers the development tooling

HAZZA INSTITUTE OF TECHNOLOGY

Setup Development Environment

Essential JavaScript and Web Browser API Concepts

HAZZA INSTITUTE OF TECHNOLOGY

- Object.assign
- Immutable Array Functions
- Function Parameter Default Values
- Arrow Functions
- Destructuring, Spread Operator, and Rest Operator
- Fetch & Promises
- ES2015 Modules

Essential JavaScript Concepts

HAZZA INSTITUTE OF TECHNOLOGY

- Object.assign – used to copy properties from one object to another
- Immutable Array Functions – produce a new array instead of mutating an existing array
- Function Parameter Default Values – used to initialize state when the application loads
- Arrow Functions – commonly used when lexical this or a simpler syntax is desired

Essential JavaScript Concepts

HAZZA INSTITUTE OF TECHNOLOGY

- Destructuring, Spreads, and Rest – makes working with properties easier
- Fetch & Promises – Fetch is the new API for making REST service calls instead of libraries such as jQuery or using the XHR object directly
  - Fetch is not a standard, but hopefully will be soon
  - Promises are an ES2015 standard with wide support

Essential JavaScript Concepts

- ES2015 Modules – Not supported natively, but TypeScript will transpile to UMD modules to be loaded by SystemJS
  - Eventually, ES2015 modules (static and dynamic – through SystemJS) should be available natively

Summary

Module 15

HAZZA INSTITUTE OF TECHNOLOGY

- Redux foundation
- Understanding of Why use Redux?
- Understanding of Three Principles of Redux
- Understanding of Essential JavaScript Concepts