Redux Middleware and Redux Thunk

Jogesh K. Muppala

THE DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING
計算機科學及工程學系

香港科技大學
THE HONG KONG UNIVERSITY OF
SCIENCE AND TECHNOLOGY

Redux Middleware

- Provides the capability to run code after an action is dispatched, but before it reaches the reducer
  - Third-party extension point
  - e.g., logging, async API calls
- Middleware:
  - Forms pipeline that wraps around the dispatch()
  - Pass actions onward
  - Restart the dispatch pipeline
  - Access the store state

Redux Middleware

- Middleware typically used for:
  - Inspecting the actions and the state,
  - Modify actions,
  - Dispatch other actions,
  - Stop actions from reaching the reducers, etc.
- The applyMiddleware() function:
  - Sets up the middleware pipeline
  - Returns a “store enhancer” that is passed to createStore()

Thunk

- In programming, a thunk is a subroutine used to inject an additional calculation in another subroutine
  - Delay a calculation until its result is needed,
  - Insert operations at the beginning or end of the other subroutine

https://en.wikipedia.org/wiki/Thunk

Redux Thunk

- Middleware that allows you to write action creators that return a function instead of an action
  - Can be used to delay the dispatch of an action, or
  - Dispatch only if a certain condition is met
- Inner function receives the dispatch() and getState() store methods

Redux Thunk

- Useful for complex synchronous logic
  - Multiple dispatches
  - Conditional dispatches
  - Simple Async logic
- Redux Saga: Uses ES6 generators to control pausable functions
  - Comples async logic
  - Ongoing “background thread” like processing behavior