



Redux Middleware

Code 401 (Class 33)

Runs between dispatch and update

dispatch({action}) => [middleware] => update store

Remember in express, we had req, res, next?

You could jack stuff onto the req (body.xxx) as you passed it through

Action (req)

Store (res)

Next (next)

Basically, take a bunch of functions and operate on the store based on the action, and then do the next thing.

What we know as “dispatch” today ... is actually the last middleware.

Reporter Middleware

Redux Reporter Middleware

Try ...

- Log action

- Update state

- Log state

Catch ...

- If error

- Log error

Wiring Middleware Into Your App

Define the middleware module

```
let reporter = store => next => action => {  
  ...  
}  
  
export default reporter;
```

Add it to the store (store.js):

```
import {createStore, applyMiddleware} from 'redux';  
import reducer from 'reducer'  
import reporter from 'redux-reporter'  
Import xyz from 'some-other-middleware'  
  
export default() => createStore(reducer, applyMiddleware(xyz,reporter));  
// MW run in order from left to right, ending with og-Dispatch //
```

Now ...

dispatch => reporter => ogDispatch => update state