Spring Boot + React

Jirawong Wongdokpuang

Course Agenda

- initial project
- babel config
- webpack config
- dev server config
- Actions [Login action]
- Reducers [Login Reducer]
- Index [Dom render]

Initial Project

- npm init
- npm install --save react react-dom react-redux redux redux-thunk
- npm install --save-dev babel-core babel-loader babel-preset-es2015 babel-preset-react babel-preset-stage-0 css-loader node-sass react-hot-loader sass-loader style-loader webpack webpack-dev-server

Dependency

- React
- Redux
- Babel
- *-loader
- webpack

Babel Configuration

- create .babelrc

```
{
  "presets": ["es2015","stage-0","react"]
}
```

webpack.config.js

- import

```
const webpack = require('webpack');
const path = require('path');
```

webpack.config.js

- config exports

```
module.exports = {
        //config here
}
```

webpack.config.js

- set devtool

devtool:'eval’

#why eval see.
https://webpack.github.io/docs/configuration.html#devtool

webpack.config.js

- config entry

```
entry: [
  'webpack-dev-server/client?http://0.0.0.0:8000',
  'webpack/hot/only-dev-server',
  './index.jsx'
]
```

webpack.config.js

- config output file

```
output: {
  publicPath: '/asset/',
  path: path.join(__dirname,'src/main/resoruces/static/asset'),
  filename: 'bundle.js'
}
```

webpack.config.js

- config dev server

```
devServer: {
  contentBase: './src/main/resources/static/',
  historyApiFallback: true,
  hot: true,
  port: 8000,
  publicPath: '/asset/',
  noInfo: false ,
  proxy: {
   '*': {
     target: 'http://localhost:8080',
     secure: false
     }
  }
}
```

webpack.config.js

- config plugin

```
plugins: [
 new webpack.HotModuleReplacementPlugin()
]
```

webpack.config.js

- config module

```
module: {
  loaders: [
    {
      test: /\.(jsx)$/,
      exclude: /node_modules/,
      loader: 'react-hot!babel-loader'
    },
    {
      test: /\.scss$/,
      exclude: /node_modules/,
      loader: 'style!css?sourceMap!sass?sourceMap'
    }
  ]
}
```

webpack.config.js

- config resolve

```
 resolve: {
   extensions: ['', '.js', '.jsx', '.css', '.scss']
}
```

server.js

- import

```
const webpack = require('webpack');
const WebpackDevServer = require('webpack-dev-server');
const config = require('./webpack.config');
```

server.js

- create server

```
new WebpackDevServer(webpack(config), config.devServer)
  .listen(config.devServer.port, '0.0.0.0', function (err) {
  if (err) {
   console.log(err);
  }
  console.log('Listening at localhost:' + config.devServer.port);
  console.log('Opening your system browser...');
 });
```

package.json

- add start server command

```
"scripts": {
  "test": "echo \"Error: no test specified\" && exit 1",
  "start": "node server.js"
},
```

actions.jsx

- login action
- export const LOGIN_REQUEST = 'LOGIN_REQUEST'
- export const LOGIN_SUCCESS = 'LOGIN_SUCCESS'
- export const LOGIN_FAILURE = 'LOGIN_FAILURE'

actions.jsx

- login action function

```
function requestLogin(creds) {
    return {
        type: LOGIN_REQUEST,
        isAuthenticated: false,
        creds
    }
}
```

actions.jsx

- login action function

```
function receiveLogin(user) {
    return {
        type: LOGIN_SUCCESS,
        isAuthenticated: true,
        id_token: user.id_token
    }
}
```

actions.jsx

- login action function

```
function loginError(message) {
    return {
        type: LOGIN_FAILURE,
        isAuthenticated: false,
        message
    }
}
```

actions.jsx

- login function

#show in github

reducers.jsx

- import

```
import { combineReducers } from 'redux';
import {
 LOGIN_REQUEST, LOGIN_SUCCESS, LOGIN_FAILURE
} from './actions'
```

reducers.jsx

- auth function

#show in github

index.jsx

- import

```
import React from 'react';
import ReactDom from 'react-dom';
import thunkMiddleware from 'redux-thunk';
import { createStore, applyMiddleware } from 'redux';
import { Provider } from 'react-redux';
import reducers from './reducers.jsx’;
```

index.jsx

- create store

let createStoreWithMiddleware = applyMiddleware(thunkMiddleware)(createStore);

let store = createStoreWithMiddleware(reducers);

index.jsx

- render provider

```
ReactDom.render(
 <Provider store={store}>
 …
 </Provider>,
 document.getElementById('app')
);
```

ABC…Z