Udemy – Curso React.js Ninja: Módulo React + Webpack

<https://www.udemy.com/cart/subscribe/course/1196394/>

1 – MRW#01 – Apresentando e configurando o ambiente

Sobre o webpack – é um module bundler. Cria um bundle (um pacote). Para entregar para o navegador. webpack.github.io

Repositório: <https://github.com/da2k/curso-reactjs-ninja>

1 – MRW#02 – Configuração básica webpack

- Criar um package.json

- Instalar webpack como dependência de desenvolvimento

- Comando para criar arquivo: echo {} >> package.json ou npm init

- Criar arquivo webpack.config.js e config para ler e gerar pacote (bundle)

- Criar arquivo src/index.js

- Gerar o bundle com webpack, comando: *node\_modules/.bin/webpack*

1 – MRW#03 – Usando o server webpack

- Instalar a dependência para o server – webpack-dev-server

npm install --save-dev webpack-dev-server@1

- Adicionar o publicPath no webpack.config. Quando executa o servr não gera arquivo físico, mas sim virtual e o publicPath set onde encontrar o virtual. Salva o bundle.js em /static/

- Subir o server: node\_modules\.bin\webpack-dev-server

1 – MRW#04 – Modularizando uma aplicação

- Criar um arquivo em src/app.js

- Entender o require e o module.exports do CommonJS

app.js

'use strict'

function sum(v1, v2) {

return v1 + v2

}

module.exports = sum

index.js

'use sctrict'

var sum = require('./app')

console.log(sum(1, 2))

1 – MRW#05 – Criando uma aplicação em React

- Instalar as dep. React e ReactDOM;

- Criar componente em src/app.js

- OBS: document.querySelector('[data-js="app"]') não utiliza ID pq cada vez que utiliza um id para o html ele gera uma variável global pendurada em window.

1 – MRW#06 – Modularizando a aplicação

- E se utilizar o ES2015?

- Instalar dependências:

- Babel e babel presets (predefinições)

npm install --save-dev babel-core@6 babel-loader@6 babel-preset-es2015@6 babel-preset-stage-0@6

- Criar arquivo .babelrc

- Passando os presets

- Atualizar o webpack.config.js para fazer a leitura ou utilize o babel para compilar.

- Atualizar o app.js para export do ES2015 e o index.js para import

- Import com shorthand notation: quando o valor da chave tem o mesmo nome da chave pode user apenas a chave. EX: { render: render } -> { render }

MRW#06.1 – Aula do future sobre ES module

- export SEM DEFAULT é chamado de export nomeado

1 - MRW#07 – Configurando JSX no babel

- Adicionar o preset do React

npm install --save-dev babel-preset-react@6

- Atualizar o .babelrc, adicionando o preset

- Atualizar src/app.js e src/index.js para usar p JSX

- Utilizando sourcemap no webpack.config.js

1-MRW#08 – Configurando a aplicação para usar o React Hot Loader

- Sempre que atualizar um componente, atualize somente o que for necessário

npm install --save-dev react-hot-loader@3.0.0-beta.2

- Atualizar o webpack.config.js

- Atualizar o .babelrc: plug-ins

- Criar um server.js para rodar com nodejs

- Criar entrada start no package.json

1 – MRW#10 – Configurando o linter

- Adicinar as dependências: standart e standart-loader

npm install --save-dev standard standard-loader@4

- Adicionar no webpack.config.js