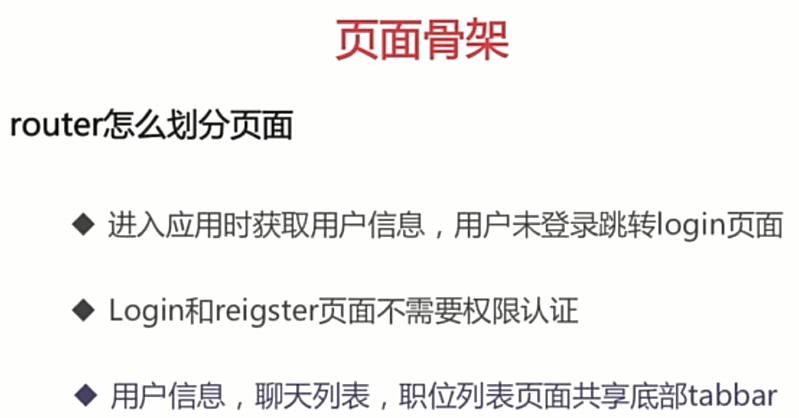
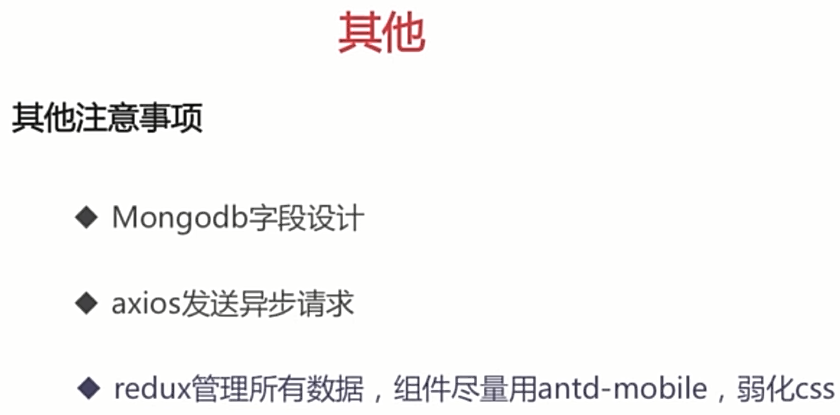
实时聊天招聘APP

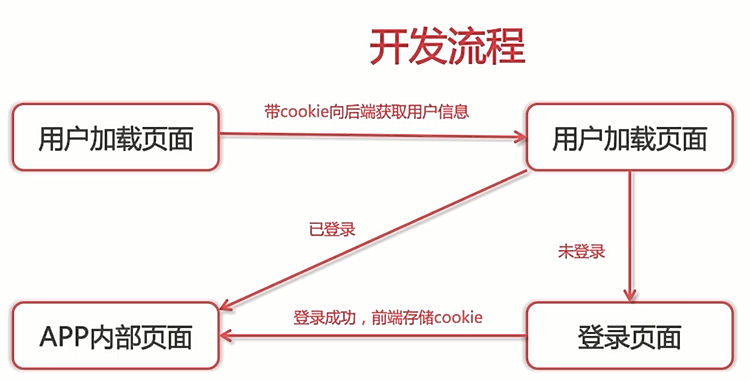












在React路由中，有这样一种写法，@withReact，效果就是把路由相关的方法通过props传给它包裹的组件的props上；

跟withRouter（component）是一个效果；

使用@这种写法的话，需要安装一个babel-plugin-transform-decorators-legacy包

cnpm i babel-plugin-transform-decorators-legacy -S

还需要在packag文件中的babel中配置...（待完善）

"plugins": [

[

"import",

{

"libraryName": "antd-mobile",

"style": "css"

}

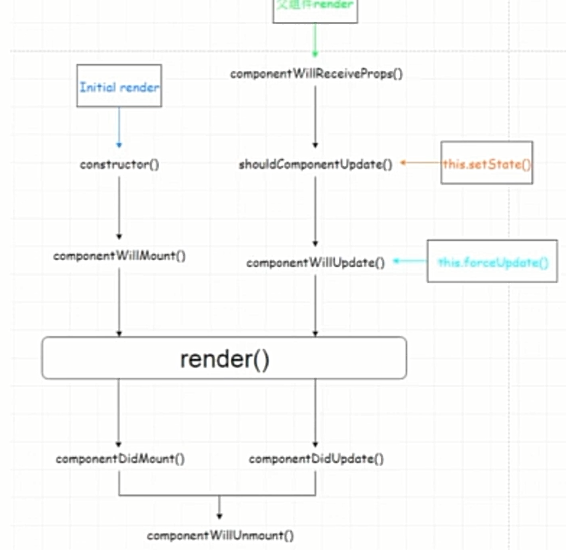
],

[

"transform-decorators-legacy"

]

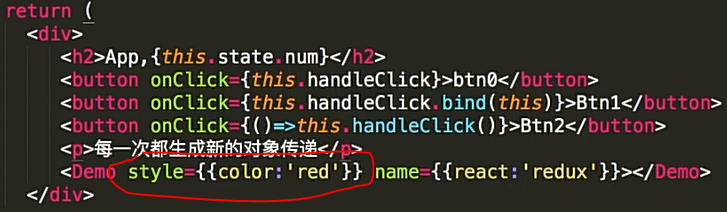
react生命周期



单组件性能优化







传递多余数据



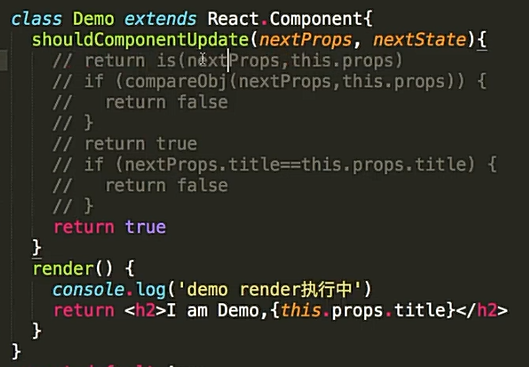
多组件优化

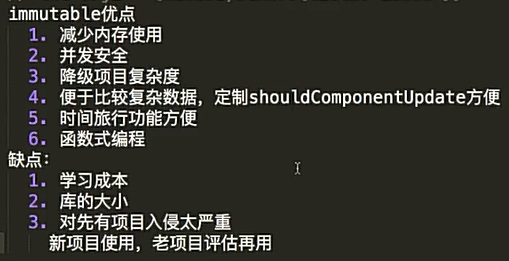
子组件属性没有变化，不需要重新渲染

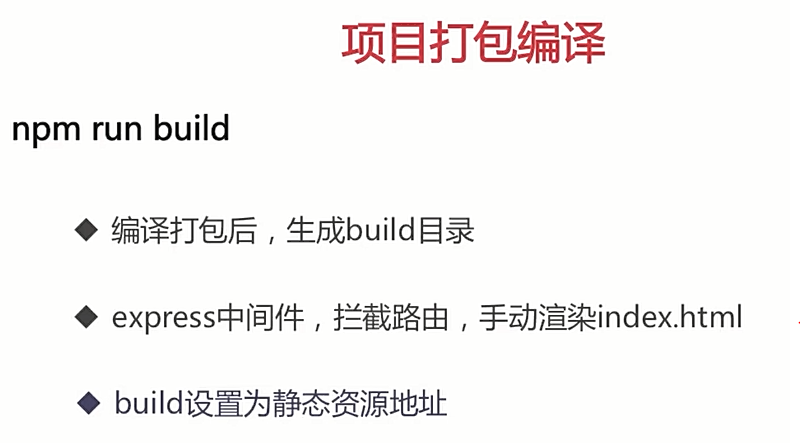


immutable.js









开发

mongodb/bin 文件路径放入path环境变量

mongodb 启动

mongod --dbpath F:mongodb-data

mongodb

show dbs

use boss

show tables

db.users.find({})

server目录下

nodemon server\_bak.js

项目启动

npm run start

localhost:3000/register

localhost:3000/login

localhost:9093

项目打包

npm run build

设置静态资源

package.json

"scripts": {

"server ":"nodemon server/server.js"

},

server.js

app.use(function(req,res,next){

    if(req.url.startsWith('/user/') || req.url.startsWith('/static/')){

        return next()

    }

return res.sendFile(path.resolve('build/index.html'));

})

app.use('/', express.static(path.resolve('build')))

npm run server

<http://localhost:9093/login>

服务器端渲染

node环境支持ES6 npm install babel-cli –save

require -> import from

npm install cross-env -g

"scripts": {

"start": "node scripts/start.js",

"build": "node scripts/build.js",

"test": "node scripts/test.js --env=jsdom",

"server": "cross-env NODE\_ENV=test nodemon --exec babel-node server/server.js",

"server\_bak":"nodemon server/server.js"

}

支持css npm install css-modules-require-hook

根目录下新建文件 cmrh,conf.js

module.exports = {

// Same scope name as in webpack build

generateScopedName: '[name]\_\_[local]\_\_\_[hash:base64:5]',

}

静态资源加载 npm install asset-require-hook --save

import assetHook from 'asset-require-hook'

assetHook({

extensions: ['png'],

limit: 8000

})

<img src={require('./job.png')} alt="" />

index.js

ReactDom.hydrate(

    (<Provider store={store}>

        <BrowserRouter>

            <App></App>

        </BrowserRouter>

    </Provider>),

    document.getElementById('root')

)

server.js

app.use(function(req,res,next){

    if(req.url.startsWith('/user/') || req.url.startsWith('/static/')){

        return next()

    }

    const context = {}

    const store = createStore(reducers, compose(

        applyMiddleware(thunk),

    ))

res.write(`<!doctype html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

<meta name="theme-color" content="#000000">

<link rel="stylesheet" href="/${staticPath['main.css']}">

<meta name="description" content="React开发招聘 App" />

<meta name="keywords" content="React,Redux,SSR,React-router,Socket.io" />

<meta name="author" content="Imooc" >

<title>Redux+React Router+Node.js全栈开发聊天App</title>

</head>

<body>

<noscript>

You need to enable JavaScript to run this app.

</noscript>

<div id="root">`

)

const markupStream = renderToNodeStream(

        (<Provider store={store}>

            <StaticRouter

                location={req.url}

         context={context}

     >

            <App></App>

            </StaticRouter>

        </Provider>)

    )

markupStream.pipe(res, { end: false });

markupStream.on('end', () => {

res.write(`

</div>

<script src="/${staticPath['main.js']}"></script>

</body>

</html>

`);

res.end();

});

})

npm run server