一、搭建React

1、必须安装nodejs

2、安装cnpm用cnpm替代npm

地址：http://npm.taobao.org/

安装cnpm:

npm install -g cnpm --registry=<https://registry.npm.taobao.org>

3、用yarn替代npm

yarn的安装：

第一种方法：参考官方文档<https://yarn.bootcss.com/>

第二种方法：cnpm install -g yarn 或者 npm install -g yarn

4.1、搭建React开发环境的第一种方法（老-现在推荐）：

①、必须要安装nodejs

注意：安装nodejs稳定版本 教程中的nodejs版本:v8.11.2 教程中的npm版本:v5.6.0

②、安装脚手架工具 （单文件组件项目生成工具） 只需要安装一次

$ npm install -g create-react-app / cnpm install -g create-react-app

③、创建项目 （可能创建多次）

找到项目要创建的目录执行：

$ create-react-app reactdemo

④、cd 到项目里面

$ cd reactdemo

$npm start/yarn start运行项目

$npm run build/ yarn build 生成项目

4.2、搭建React的开发环境的第二种方法（新-未来推荐）：

①、必须要安装nodejs

注意：安装nodejs稳定版本 教程中的nodejs版本:v8.11.2 教程中的npm版本:v5.6.0

②、安装脚手架工具并创建项目

找到项目要创建的目录执行：

$ npx create-react-app reactdemo

备注：npx create-react-app reactdemo这条命令会临时安装 create-react-app 包，命令完成后create-react-app 会删掉，不会出现在 global 中。下次再执行，还是会重新临时安装。

③cd 到项目里面

$ cd reactdemo

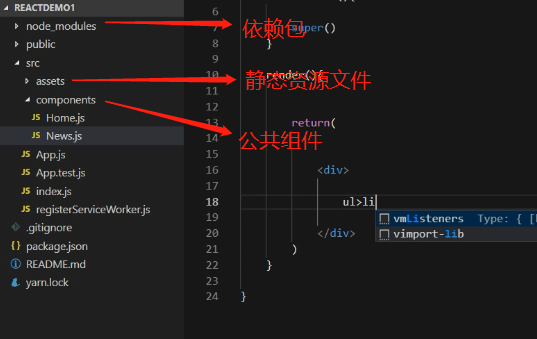
$npm start/yarn start运行项目（调试）

$npm run build/ yarn build 生成项目（发布）

npx介绍：npm v5.2.0引入的一条命令（npx），引入这个命令的目的是为了提升开发者使用包内提供的命令行工具的体验。npx 会帮你执行依赖包里的二进制文件。

二、React目录结构

1、React目录结构



三、绑定属性( 绑定class 绑定style)、引入图片、循环数组渲染数据

1、所有的模板要被一个根节点包含起来

2、模板元素不要加引号

3、{}绑定数据

4、绑定属性注意：

①class 要变成 className

②for 要变成 htmlFor

③style属性和以前的写法有些不一样

<div style={{'color':'blue'}}>{this.state.title}</div>

<div style={{'color':this.state.color}}>{this.state.title}</div>

5、循环数据要加key

6、组件的构造函数中一定要注意 super

子类必须在constructor方法中调用super方法，否则新建实例时会报错。这是因为子类没有自己的this对象，而是继承父类的this对象，然后对其进行加工。如果不调用super方法，子类就得不到this对象

constructor(props){

super(props); /\*用于父子组件传值 固定写法\*/

this.state={

userinfo:'张三'}

}

7、组件名称首字母大写、组件类名称首字母大写

四、绑定事件处理函数this的几种方法

在以类继承的方式定义的组件中，为了能方便地调用当前组件的其他成员方法或属性（如：this.state），通常需要将事件处理函数运行时的 this 指向当前组件实例。

第一种方法：（改变this指向的方法）

run(){

alert(this.state.name)

}

<button onClick={this.run.bind(this)}>按钮</button>

第二种方法：（在构造函数中改变this指向的方法）

构造函数中改变

this.run = this.run.bind(this);

run(){

alert(this.state.name)

}

<button onClick={this.run>按钮</button>

第三种方法：（通过箭头函数）

run=()=> {

alert(this.state.name)

}

<button onClick={this.run>按钮</button>

五、快速创建

添加插件：VS Code ES7 React/Redux/React-Native/JS snippets

1.快速创建命令：

①rce：快速创建类

②rconst：快速创建构造方法

③met→：快速生成箭头函数（methodName = (params) => { }）

④sst：快速生成this.setState方法

六、

1、阻止submitd的提交事件：event.preventDefault()。

2、list渲染成input（type="checkbox"）控件时，要注意this指向。

this.state.hobby.map((value, key) => {

return (

<span>

<input type="checkbox" checked={value.checked} onChange={this.handelHobby.bind(this, key)} />{value.title}

</span>)})

七、父子组件传值（react 父子组件通信）
1、父组件给子组件传值
①、在调用子组件的时候定义一个属性 <Header msg='首页'></Header>
②、子组件里面 this.props.msg
说明：父组件不仅可以给子组件传值，还可以给子组件传方法,以及把整个父组件传给子组件,可以让子组件给父组件传值。
2、父组件主动获取子组件的数据(新版本已废弃)
①、调用子组件的时候指定ref的值 <Header ref='header'></Header>
②、通过this.refs.header 获取整个子组件实例 (dom（组件）加载完成以后获取 )

2.1父组件主动获取子组件的数据（新版本）

①

import React, { Component } from 'react'

export class Father extends Component {

click = (e) => {

this.child.myName()

}

render() {

return (

<div>

<Child ref={child => { this.child = child }} />

<button onClick={this.click} >click</button>

</div>

)

}

}

class Child extends Component {

myName = () => alert('xiaohesong')

render() {

return (

<div>

</div>

)

}

}

②（推荐）

import React, {Component} from 'react';

export default class Parent extends Component {

render() {

return(

<div>

<Child onRef={(ref) => {this.child = ref}} />

<button onClick={this.click} >click</button>

</div> ) }

click = (e) => {

this.child.myName() }

}

class Child extends Component {

constructor(props) {

super(props)

this.props.onRef(this)

}

myName = () => alert('xiaohesong')

render() {

return ('woqu') }

}

3、defaultProps：父子组件传值中，如果父组件调用子组件的时候不给子组件传值，可以在子组件中使用defaultProps定义的默认值

4、propTypes：验证父组件传值的类型合法性

①、引入import PropTypes from 'prop-types';

②、类名.propTypes = {
name: PropTypes.string
};

注意：defaultProps和propTypes都是定义在子组件里面

八、react获取服务器APi接口的数据

react中没有提供专门的请求数据的模块，但是我们可以使用任何第三方请求数据模块实现请求数据。

1、axios（https://github.com/axios/axios）：axios的作者觉得jsonp不太友好，推荐用CORS方式更为干净（后端运行跨域）

①、安装axios模块npm install axios --save / npm install axios --save

②、在哪里使用就在哪里引入import axios from 'axios'

③、看文档使用
var api='http://www.phonegap100.com/appapi.php?a=getPortalList&catid=20';
axios.get(api)
.then(function (response) {
console.log(response);
})
.catch(function (error) {
console.log(error);
});

2、fetch-jsonp（https://github.com/camsong/fetch-jsonp）

①、安装 npm install fetch-jsonp --save

②、import fetchJsonp from 'fetch-jsonp'

③、看文档使用

fetchJsonp('/users.jsonp')
.then(function(response) {
return response.json()
}).then(function(json) {
console.log('parsed json', json)
}).catch(function(ex) {
console.log('parsing failed', ex)
})

3、其他请求数据的方法也可以...自己封装模块用原生js实现数据请求也可以...

十、react传值

1、react动态路由传值
①、动态路由配置
<Route path="/content/:aid" component={Content} />
②、对应的动态路由加载的组件里面获取传值
this.props.match.params

跳转：<Link to={`/content/${value.aid}`}>{value.title}</Link>
2、react get传值

①、通过url地址后拼接参数传值]

http://IP:HOST?id=123

②、获取 this.props.location.search

十一、react解析html

官方文档： https://reactjs.org/docs/dom-elements.html

例如：<div className="p\_content" dangerouslySetInnerHTML={{\_\_html: this.state.list.content}}> </div>

十二、js跳转路由

官方文档：https://reacttraining.com/react-router/web/example/auth-workflow
1、要引入Redirect
import {
BrowserRouter as Router,
Route,
Link,
Redirect,
withRouter
} from "react-router-dom";
2、定义一个flag
this.state = {
loginFlag:false
};
3、render里面判断flag 来决定是否跳转
if(this.state.loginFlag){
return <Redirect to={{ pathname: "/" }} />;
}
4、要执行js跳转
通过js改变loginFlag的状态
改变以后从新render 就可以通过Redirect自己来跳转

十三、路由模块化以及嵌套路由父子组件传值

官方文档：https://reacttraining.com/react-router/web/example/route-config

十四、React中使用Antd

antd官网：https://ant.design/index-cn

1、安装antd npm install antd --save / yarn add antd / cnpm install antd --save

2、在您的react项目的css文件中引入 Antd的css

@import '~antd/dist/antd.css';

3、看文档使用：

如使用Button：

①、在对应的组件中引入Antd：import { Button } from 'antd';

②、<Button type="primary">Primary</Button>

十五、React中使用Antd高级配置，按需引入css样式

官方文档：https://ant.design/docs/react/use-with-create-react-app-cn中高级配置

1、安装antd npm install antd --save

2、安装（react-app-rewired）一个对 create-react-app 进行自定义配置的社区解决方案

yarn add react-app-rewired / cnpm install react-app-rewired --save

3、修改 package.json

react-scripts 需改为react-app-rewired

"scripts": {

"start": "react-app-rewired start",

"build": "react-app-rewired build",

"test": "react-app-rewired test --env=jsdom",

"eject": "react-app-rewired eject"

}

4、安装babel-plugin-import babel-plugin-import是一个用于按需加载组件代码和样式的 babel 插件

yarn add babel-plugin-import / cnpm install babel-plugin-import --save

5、在项目根目录创建一个 config-overrides.js 配置文件：

const { injectBabelPlugin } = require('react-app-rewired');

module.exports = function override(config, env) {

config = injectBabelPlugin(

['import', { libraryName: 'antd', libraryDirectory: 'es', style: 'css' }],

config,

);

return config;

};

6、然后移除前面在 src/App.css 里全量添加的 @import '~antd/dist/antd.css'; 直接引入组件使用就会有对应的css

import { Button } from 'antd';

<Button type="primary">Primary</Button>