|  |
| --- |
| - |
|  |

|  |
| --- |
| - 使用方法：程序的入口文件为根目录下的index.html，所有js已经经过babel转译和browserify打包，打开index即可见demo效果。 |
|  |

|  |
| --- |
| - |
|  |

|  |
| --- |
| - |
|  |

|  |
| --- |
| - |
|  |

|  |
| --- |
| - |
|  |

|  |
| --- |
| - |
|  |

|  |
| --- |
| - |
|  |

|  |
| --- |
|  |

|  |
| --- |
| - |
|  |

|  |
| --- |
| - |
|  |

|  |
| --- |
| - |
|  |

|  |
| --- |
| - |
|  |

|  |
| --- |
| - |
|  |

|  |
| --- |
| - |
|  |

|  |
| --- |
| - |
|  |

|  |
| --- |
| - |
|  |

|  |
| --- |
| - |
|  |

|  |
| --- |
| - |
|  |

|  |
| --- |
| -工程化步骤： |
|  |

|  |
| --- |
| -1 --文件分类准备 |
|  |

|  |
| --- |
| - 1.1css目录，分为主界面css和component文件夹，component文件夹里面的css，主要写组件的样式； |
|  |

|  |
| --- |
| - |
|  |

|  |
| --- |
| - 1.2js目录，主要分为source和biuld文件夹，source文件夹为原始js，build为经过babel转译的js文件，source和build文件夹都分app.js和componengt.js,app.js主要为调用小组件的入口，component立面为组件js编写； |
|  |

|  |
| --- |
| - |
|  |

|  |
| --- |
| - 1.3根目录下放置index.html； |
|  |

|  |
| --- |
| - |
|  |

|  |
| --- |
| - 1.4根目录建立source文件夹，里面放app需要用到的静态文件资源； |
|  |

|  |
| --- |
| - |
|  |

|  |
| --- |
| -2 --编译 |
|  |

|  |
| --- |
| - 2.1使用babel编译，使用babel需要先下载react的依赖包; |
|  |

|  |
| --- |
| - 下载react依赖包需要在根目录下package.json文件 |
|  |

|  |
| --- |
| - package.json通过在命令行，键入npm init可以创建 |
|  |

|  |
| --- |
| - 字段填写随意，创建玩package.json之后。安装react依赖。 |
|  |

|  |
| --- |
| - 命令为： |
|  |

|  |
| --- |
| - npm install --save-dev react (--save-dev写进package.json意思) |
|  |

|  |
| --- |
| - npm install --save-dev react-dom |
|  |

|  |
| --- |
| - npm install --save-dev babel-preset-react |
|  |

|  |
| --- |
| - npm install --save-dev babel-preset-es2015. |
|  |

|  |
| --- |
| - |
|  |

|  |
| --- |
| - 通过babel编译转为es5命令（根目录下执行） |
|  |

|  |
| --- |
| - ：babel --preset react,es2015 js/source -d js/build |
|  |

|  |
| --- |
| - (转译规则为添加的react和es2015，把js/source目录下所有js文件转到 js/build文件夹下面) |
|  |

|  |
| --- |
| - |
|  |

|  |
| --- |
| - ！！！！注意！！！！在调用babel命令前，必须新建babel的配置文件.babelrc |
|  |

|  |
| --- |
| - ，立面键入{ |
|  |

|  |
| --- |
| - "presets": [ |
|  |

|  |
| --- |
| - "es2015","react", ], |
|  |

|  |
| --- |
| - "plugins": [] |
|  |

|  |
| --- |
| - } |
|  |

|  |
| --- |
| - |
|  |

|  |
| --- |
| - |
|  |

|  |
| --- |
| - 其中es2015，react为编码规则。 |
|  |

|  |
| --- |
| - |
|  |

|  |
| --- |
| -3 --打包转译后的js文件 |
|  |

|  |
| --- |
| - 用browserify,命令为 |
|  |

|  |
| --- |
| - ：browserify js/build/app.js -o bundle.js |
|  |

|  |
| --- |
| - (打包js/build中的app.js，这个为应用入口，打包成一个bundle.js在根目录下) |
|  |

|  |
| --- |
| - |
|  |

|  |
| --- |
| - --打包css同理，css由于无依赖，可以手动打包为bundle.css |
|  |

|  |
| --- |
| - 放于根目录下。 |
|  |

|  |
| --- |
| - |
|  |

|  |
| --- |
| -4 --index.html中引用bundle.css和bundle.js即可 |
|  |

|  |
| --- |
| - |
|  |

|  |
| --- |
| -据说可以把babelrc的配置文件，写入packjason，尝试了下失败了。目前就用这套方案，日后再说，还有webpack先使用着学起来了。 |
|  |

|  |
| --- |
| - |
|  |

|  |
| --- |
| - |
|  |

|  |
| --- |
| - 2017/4/20 |
|  |

|  |
| --- |
| - |
|  |

|  |
| --- |
| - |
|  |

|  |
| --- |
| - |
|  |

|  |
| --- |
| - |
|  |

|  |
| --- |
| -state与props的大坑，在es6， |
|  |

|  |
| --- |
| -class ?? extends React.Component{()}中已经没有 |
|  |

|  |
| --- |
| -getinitistate方法，也没有getdefaultprops和propTypes方法 |
|  |

|  |
| --- |
| -，在es6中使用 |
|  |

|  |
| --- |
| -state的话: |
|  |

|  |
| --- |
| -constructor(props){ |
|  |

|  |
| --- |
| -super(props); |
|  |

|  |
| --- |
| -this.state={ss:'xx',} |
|  |

|  |
| --- |
| -}}这种 |
|  |

|  |
| --- |
| - |
|  |

|  |
| --- |
| -默认prop和prop用 |
|  |

|  |
| --- |
| - |
|  |

|  |
| --- |
| -static defaultPorps={ |
|  |

|  |
| --- |
| -x:'123', |
|  |

|  |
| --- |
| -y:'312', |
|  |

|  |
| --- |
| - |
|  |

|  |
| --- |
| -} |
|  |

|  |
| --- |
| - |
|  |

|  |
| --- |
| -static propTypes={ |
|  |

|  |
| --- |
| -x:React.PropTypes.bool.isRequired, |
|  |

|  |
| --- |
| - y: React.PropTypes.number.isRequired, |
|  |

|  |
| --- |
| - |
|  |

|  |
| --- |
| -} |
|  |

|  |
| --- |
| - |
|  |

|  |
| --- |
| -2017/4/26 |
|  |

|  |
| --- |
| - |
|  |

-