[简介](#header-c9)  
 [运行](#header-c23)  
 [快速上手](#header-c42)  
 [框架信息](#header-c49)  
 [开发规范](#header-c50)  
 [scss/css规范](#header-c71)  
 [组件设计原则](#header-c89)  
 [目录结构](#header-c101)  
 [页面](#header-c105)  
 [页面详细功能](#header-c152)  
 [首页](#header-c153)  
 [引导页](#header-c157)  
 [问答页](#header-c161)  
 [产品页](#header-c165)  
 [常见问题页](#header-c169)  
 [报告书](#header-c173)  
 [自定义](#header-c177)  
 [意见反馈](#header-c181)  
 [文件命名](#header-c185)  
 [css 结构及模块化](#header-c192)  
 [其他](#header-c204)  
 [特性](#header-c215)  
 [参考资料](#header-c243)

## 简介

这个项目主要使用了React框架

并引入了react-router redux 等相关插件

以 ES6 写法为主

通过 webpack1.x 方式打包

通过 express 服务模拟接口数据

## 运行

安装好 nodejs

npm install webpack-dev-server -g (一次)

npm run dll (一次)

* 开发环境: npm start (本地模拟数据的话多开个： npm run mock)
* 编译打包: npm run [uat/prd]
* windows环境构建uat包: npm run windows-uat

## 快速上手

快速添加一个页面

// 第一步：src/routers中的 childRoutes 变量里，添加个组件  
require('./modules/Demo').default  
  
// 第二步：src/modules 下创建目录及文件  
Demo  
 |-- index.js 配置路由  
 |-- Demo.js 组件  
 |-- Demo.scss 样式  
  
// 第三步：编辑index.js  
export default {  
 path: 'demo',  
 getComponent:(nextState, cb) => {  
 require.ensure([], (require) => {  
 cb(null, require('./Demo').default)  
 },'demo')  
 }  
}  
   
// 第四步：编辑 Demo.js  
import React from 'react';  
import ReactDOM from 'react-dom';  
import {connect} from 'react-redux';  
import RComponent from '../../components';  
import styles from './Demo.scss';  
  
class Demo extends RComponent{  
 constructor(props,context){  
 super(props,context);  
 this.state = {  
 loading: false  
 }  
 }  
 componentDidMount(){  
 document.getElementById('blankPage').style.display = 'none';  
 }  
 render(){  
 return (  
 <h1 className={styles.demo}>this is a h1</h1>  
 )  
 }  
}  
export default connect(  
 state => ({  
   
 })  
)(Demo)  
   
// 第五步 编辑 demo.scss  
.demo {  
 color: red;  
}

## 框架信息

### 开发规范

* 所有代码必须进过编译运行才能提交代码库，避免因为报错影响其它开发进度
* 页面代码放modules文件夹,index.js为路由配置文件
* 组件代码放components文件夹
* 页面组件继承RComponent
* 页面组件自动引入Loading、Sdc组件
* 第三方组件必须经过评审才能引进

### scss/css规范

* 使用scss编写样式
* 公共样式放style文件夹
* 页面样式放modules对应模块文件夹
* 组件样式文件放跟组件同一个目录
* 以中横线区分名称，如图标以pi-icon-图标命名、按钮以pi-btn-样式｜大小｜等命名

### 组件设计原则

* 页面拆分成多个react组件
* 尽量使用无状态组件，父组件与子组件通过props传递
* 动画交互效果的组件可使用自有状态

### 目录结构

-- aa\_fe\_h\_cust  
 |-- config [webpack 配置文件]  
 |-- node\_modules [npm模块目录]  
 |-- server [server api]  
 |-- src [源代码]  
 |-- components [react 组件]  
 |-- constants [工程常量]  
 |-- images [图片]  
 |-- layout [布局]  
 |-- modules [模块，包含路由配置、页面、组件、actions、scss样式文件]  
 |-- reducers [reducer]  
 |-- styles [通用.scss样式文件]  
 |-- utils [各类helper]  
 |-- app.js [入口文件]  
 |-- root.js [顶级组件]  
 |\_\_ routes.js [路由配置]  
 |-- .babelrc [转码配置]  
 |-- .eslintrc  
 |-- .gitignore  
 |-- build.sh [jenkins构建脚本]  
 |-- index.js [服务器程序]  
 |-- package.json   
 |-- README.md

### 页面

共9个页面

|  |  |  |
| --- | --- | --- |
| 名称 | 路径 | 模块 |
| 首页 [资产配置建议页] | / | Home |
| 引导页 | /#/guideAnimation | guideAnimation |
| 问答页 | /#/askTest | askTest |
| 产品页 | /#/products | Products |
| 常见问题页 | /#/question | Question |
| 报告书 | /#/report | AssetsReport |
| 自定义页面 | /#/custom | Custom |
| 意见反馈 | /#/suggestion | Suggestion |
| 404页面 | /#/notfound | notfound |

### 页面详细功能

#### 首页

模块 - 客户信息 调用suggestList接口 [资产配置建议主页接口]  
模块 - 平安证券定制组合 调用suggestList接口 [资产配置建议主页接口]  
模块 - 历史收益对比 调用hisProfitGraphList接口 [主页收益曲线]  
模块 - 推荐组合大类配置 调用suggestList接口 [资产配置建议主页接口]  
内容 - 推荐理由   
链接 - 常见问题  
链接 - 资产配置介绍  
链接 - 资产配置报告书  
链接 - 意见反馈  
链接 - 查看推荐产品

#### 引导页

总共 4 个页面  
引用 react-swipe 插件 官网：https://github.com/voronianski/react-swipe  
最后一个页面，跳转到下一个页面  
动画效果使用 css3

#### 问答页

总共 4 个页面  
引用 react-swipe 插件 官网：https://github.com/voronianski/react-swipe  
进入页面调用接口 quickUserInfo [BI分析结果]  
提交问答调用接口 queryWealthHealthValue [财富健康值]

#### 产品页

模块 - 一年收益率 调用recommendProdsNew接口 [查询推荐产品]  
模块 - 买入金额 调用saveUserDefineAllocationNew接口 [保存客户自定义配置方案]  
模块 - 产品信息 调用recommendProdsNew接口 [查询推荐产品]  
模块 - 合计 调用recommendProdsNew接口 [查询推荐产品]  
链接 - 自定义配置   
链接 - 一键下单 调用 /custlogin/getuserinfo 接口 [用户信息接口]

#### 常见问题页

问题列表在 JSON 中 problem.js  
需要操作问题列表可以在 problem.js 中操作

#### 报告书

主要有4个图表  
圆饼图使用 svg 形式，在component下的SvgChart里  
另3个使用了 echarts 画出来  
默认柱子图 Bar.js  
横向柱子图 InvertedList.js  
热力图 ColorBox.js

#### 自定义

主要二部份  
1 圆饼图是使用 component下的 SvgChart  
2 滑动器使用的是 react-rangeslider 官网：https://github.com/whoisandie/react-rangeslider

#### 意见反馈

提交时调用 saveUserQuestionnaire 接口 [保存用户问卷调查]

### 文件命名

1 路由全部使用小写

2 modules 与 components下文件(除了index.js)，其他用大驼峰法(Products.js)

### css 结构及模块化

1 结构

styles 全局样式  
app  
|— \_flex.scss flexbox 布局公用类  
 |— \_animation.scss animation动画公用类  
 |— \_mixins.scss 常用mixin  
 |— \_variable.scss 变量声明  
|— app.scss 入口  
|— \_page.scss   
|— \_pi-button.scss ...  
components 组件下的样式一一对应  
modules 组件与样式一一对应

2 写法

/\*\*  
 \* 在 sass 文件中  
 \*/  
  
// 全局样式  
:global {  
 .class1 {}  
 .class2 {}  
}  
  
// 局部样式  
.class1 {}  
.class2 {}  
  
// 局部样式中嵌入全局样式  
.class1 {  
 :global {  
 .class2 {}  
 }  
}  
  
// 如果局部样式下是标签选择器，则默认就是全局  
.class1 {  
 h1 {}  
}  
  
// animation 动画全局  
@keyframes :global(fadeIn){  
 from {}  
 to {}  
}

/\*\*  
 \* 在组件中调用  
\*/  
import styles from './tmp.scss';  
  
// 简单调用  
<div className={styles.class1}></div>  
  
// 同时输出局部与全局类名  
<div className={`globalClass ${styles.localClass}`}></div>  
  
// 根据state状态判断是否需要输出  
<div className={this.state.foo ? "list" : "list2"}></div>

## 其他

1 layers 弹出框设置

/\*\*  
 \* 第一个参数： 内容，默认是：服务器忙  
 \* 第二个参数： 存在时间，单位s，默认2s  
 \*/  
layers.tip(content,time)  
  
/\*\*  
 \* 参数：  
 \* content： 内容  
 \* yesText: '确认文本', 默认确定  
 \* noText: '取消文本', 默认是 取消  
 \* shadeClose 是否允许点击遮罩关闭  
 \* yes: function(){} 确认回调函数  
 \* no:function(){} 取消回调函数  
 \*/  
layers.confirm({  
 content: '',  
 yesText:'',  
 noText:'',  
 shadeClose: false,  
 yes:function(){},  
 no:function(){}  
})

2 echarts 的使用

// 1 构建 echarts 库生成放在 utils 目录中，减少 echarts 大小  
// http://echarts.baidu.com/builder.html  
  
// 2 将 echarts-for-react 库弄一份到 utils 下命名为 react-echarts.js，并修改react-echarts.js  
将 var \_echarts = require('echarts'); 改成  
var \_echarts = require('./echarts');  
  
// 注释掉  
21行 \_elementResizeEvent  
23行 \_elementResizeEvent2  
58-60行   
  
// 3 react 调用参考：https://github.com/hustcc/echarts-for-react  
// 参数参考： http://echarts.baidu.com/option.html

## 特性

* react
* react-router
* redux
* redux-thunk
* react-router-redux
* es6
* webpack
* express

## 参考资料

* thinking in React
* <https://facebook.github.io/react/docs/thinking-in-react.html>
* <https://facebook.github.io/react/docs/interactivity-and-dynamic-uis.html>
* react 组件生命周期
* <http://reactjs.cn/react/docs/component-specs.html>
* react-router
* <https://github.com/reactjs/react-router/tree/master/docs/guides>
* redux-tutorial
* <https://github.com/react-guide/redux-tutorial-cn>
* Flux 架构入门教程
* <http://www.ruanyifeng.com/blog/2016/01/flux.html>
* react-redux
* <https://github.com/reactjs/react-redux/blob/master/docs/api.md#examples>
* fetch API
* <https://github.com/github/fetch>
* flex 布局
* <http://www.ruanyifeng.com/blog/2015/07/flex-grammar.html>
* <http://www.ruanyifeng.com/blog/2015/07/flex-examples.html>
* es6
* <http://es6.ruanyifeng.com>
* immutable-js
* <https://facebook.github.io/immutable-js/>
* git
* <http://www.ruanyifeng.com/blog/2015/12/git-cheat-sheet.html>