**资源中心项目编码规范**

|  |
| --- |
| **公开** |

版本：v1.0

编号: 2018-08-05

历史变更记录

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 日期 | 内容 | 修订人 |
| 1.0 | 2018-08-05 | 创建文件 | 吕翠华 |
|  |  |  |  |
|  |  |  |  |

1. **项目目录整体架构说明**

scripts

该目录保存项目工程配置脚本，一般不用修改该目录下文件

public

该目录保存项目工程页面模板文件，只有需要修改页面模板时才需要修改

doc

该目录保存项目中所有相关的文档文件，可添加新的说明文档或修改现有文档

config

该目录保存项目的webpack配置文件

src

该目录保存工程中所有的源码和资源文件，我们所有修改都集中在该目录下

Components

保存工程中用到的所有基础组件

Constants

保存工程中所有用到的常用定义

Resource

保存工程中所有图像和字符串的定义

Images

项目中用到的所有图像目录

Strings

项目中用到的所有字符串文件目录

Styles

保存工程中所有用到的样式和less文件

Modules

按功能模块划分项目，每个功能模块一个目录，每个模块还可划分子模块，依次递归

1. **模块的划分**
   1. 划分原则

模块按照独立的功能划分不同的模块，对于功能独立的内容尽快划分成模块，模块可依层递归划分

* 1. 模块文件组成

1.index.js

index.js文件是模块里必须存在的文件，每个index文件必须至少导出一个ContentView(主显示对象，名字可以自定义)和Reducers对象。

index.js是模块对外的唯一接口文件，所有模块需要对外公开的功能通过该文件导出

2.View目录

View目录保存模块下所有的可视化组件，每个组件也可继续细分

3.store目录

每个模块都有自己的store目录，可自定义名称，如：模块名+store，该目录下保存所有该模块相关的 actions,actiontypes,reducers等

4.除了index.js外，其他文件都可自由命名，但必须遵守以下规则：

1.每个模块一个目录，目录名和组建名称相同，首字母大写

2.每个组件一个文件，文件名和组建名称相同，首字母大写

3.对于store相关，最好遵循: 模块名+store模块名+action …等命名方式

1. **引用路径的处理**

为了在工程中引用方便，项目工程里已经预定义了以下目录，对于一下目录中的文件引用，可直接使用预定义目录，而无需使用繁琐的相对路径

src 对应工程项目下的src目录

Styles 对应工程项目下的src/Styles目录

Modules 对应工程项目下的src/Modules目录

Componets 对应工程项目下的src/ Componets目录

Strings 对应工程项目下的src/ Resource/Strings目录

Images 对应工程项目下的src/ Resource/Images目录

Constants 对应工程项目下的src/Constants目录

ljbase 对应ljbase基础类库

thanos 对应公用基础类

1. **公用组件**

工程中所有显示组件，从BaseComponent中继承，不要直接从React.Component继承

1. **资源处理**

工程中所有用到的图像，音视频，字符串等都必须统一放在工程的resource目录下。

图像放在Images目录下，可按模块划分目录

所有字符串定义都放在Strings目录下，可按模块划分文件

其他所有资源放在Media目录下，可按模块划分目录

\*\*\*\*不要在代码中硬编码字符串\*\*\*\*\*

1. **项目store**

原则：

每个模块引用和使用自己的store,没有任何理由直接读取其他模块的store数据，更不能修改其他模块store中的数据

需要用到其他模块中的数据，需要在模块中导出函数来访问

不要直接修改store中的数据，请通过reducer返回

相关文件：

模块Actions.js

定义模块中所有用到的action

模块ActionTypes.js

定义模块中所有用到的action类型

模块Reducers.js

定义模块中所有action类型处理函数

模块store.js

定义模块在整体模块中用到的key,和初始值

必须导出 getMyState(state)函数

组件封装

mapStateToProps函数

const mapStateToProps = (state) =>{

let state = getMyState(state);//返回自己的state，以免破坏其他模块的state

}

1. **样式文件管理**
2. 所有的样式文件统一采用less文件格式
3. 所有less文件统一存放在src/Styles/css目录下
4. 样式应把一些共用的样式提炼到公用less文件中，统一存放在src/Styles/css/public目录下
5. 原则上每个页面对应一个less文件，less文件增加类前缀以试区分
6. 类前缀的命名规则： 模块名+页面名

@home\_indexPrefixCls: home\_index;

[.@{home\_indexPrefixCls}](mailto:.@%7bhome_indexPrefixCls%7d){

.article{

}

}

1. 所有页面对应的less文件应按照模块分对应目录
2. 在src/Styles目录下存在index.less文件，他将统一导入所有用到的样式less文件