数据易

2018年12月14日

内部资料 敬请保密

模块记录

| 版本 | 版本日期 | 修改人 | 备注 |
| --- | --- | --- | --- |
|  | 2018-12-11 | 郑铭浩 | 创建文档 |
|  | 2018-12-11 | 郑铭浩 | 新增菜单栏高亮功能说明 |

目录

[2018年9月18日 I](#_Toc525058185)

[内部资料 敬请保密 I](#_Toc525058186)

[1 文档概述 4](#_Toc525058187)

[1.1 目的 4](#_Toc525058188)

[2 产品概述 4](#_Toc525058189)

[2.1 功能层级结构 4](#_Toc525058190)

[2.2 名词说明 4](#_Toc525058191)

[3 版本迭代一览 5](#_Toc525058192)

[4 总体架构设计 6](#_Toc525058193)

[4.1 开发环境说明 6](#_Toc525058194)

[4.2 适配性说明 6](#_Toc525058195)

[4.3 功能模块化的设计与实现 6](#_Toc525058196)

[4.4 依赖库 6](#_Toc525058197)

[4.5 其他 6](#_Toc525058198)

[5 关键功能技术设计与实现 7](#_Toc525058199)

[5.1 关键功能的实现 7](#_Toc525058200)

[5.1.1 Token值 7](#_Toc525058201)

[5.1.2 菜单栏高亮功能 7](#_Toc525058202)

[5.2 通用组件 11](#_Toc525058203)

[5.2.1 http请求 11](#_Toc525058204)

[6 页面功能技术设计与实现 12](#_Toc525058205)

[6.1 首页 12](#_Toc525058206)

[6.1.1 轮播栏 12](#_Toc525058207)

[7 附录 13](#_Toc525058208)

# 文档概述

## 目的

记录项目复杂功能模块逻辑思路，方便员工阅读，快速了解项目整体架构。

# 产品概述

## 项目结构

|-- config webpack 配置文件

|-- doc 项目文档

|-- node\_modules 项目依赖

|-- public 静态文件目录

|-- scripts 脚手架自动化脚本

|-- src 项目源码目录

|--- assets 项目资源文件目录

|--- images 项目图片资源

|--- js 项目通用js

|--- css|less 项目通用样式

|--- components 通用组件

|--- config 项目配置文件

|--- data 测试数据目录

|--- layout 项目布局

|--- lib 内部依赖库

|--- sdcube 申迪组件库

|--- pages 项目页面组件

|--- module1 一级项目模块

|--- XXX.js 页面组件

|--- XXX.less 页面样式

|--- module2 多级项目模块

|--- subModule1 页面模块1

|--- XXX2.js

|--- XXX2.less

|--- subModule1 页面模块2

|--- XXX2.js

|--- XXX2.less

|--- api.js 数据接口

|--- App.js 应用级组件

|--- index.js 项目入口文件

|--- NotFound.js 404页面组件

|--- routeConfig.js 路由配置

|-- .gitignore gitlab提交忽略规则

## 名词说明

**token：**数据交互请求操作都需要使用token作为身份校验

**userId：**用户id，部分情况下可用于判断是否有某些模块的权限

# 版本迭代一览

本说明书及以下表格仅针对需求中，涉及特殊业务逻辑、交互逻辑的功能设计作重点说明。各版本的所有需求请看具体需求文档、设计原型。

|  |  |
| --- | --- |
| **版本号** | **功能** |
|  |  |
|  |  |
|  |
|  |

# 总体架构设计

## 开发环境说明

以react作为前端框架，axios作为http请求库，ant-design作为基础UI库进行。

## 适配性说明

目前仅以兼容谷歌为主，后续需考虑兼容火狐、搜狗、360等主流浏览器

## 功能模块化的设计与实现

前端代码文件结构

## 依赖库

React：https://reactjs.org/

ant-design：https://ant.design/docs/react/introduce-cn

axios: https://www.npmjs.com/package/axios

…

## 其他

记录项目代码中比较重要的细节。

# 关键功能技术设计与实现

## 关键功能的实现

### Token值

所有的接口访问都是基于Token的身份验证通过才能继续后续的业务操作。

Token需要用户名和密码才能获取，此后的Token验证必须带上用户名和Token两个信息才能验证是否具备权限。

目前Token有效期设定为2h。失效之后使用原Token将不能进行http请求的调用。Token超时，后台返回状态码需要变成401，前端通过拦截401重定向回到登录界面重新登录。

### 菜单栏高亮功能

实现菜单栏高亮的组件是NavMenu.js,通过固定的路由地址配置以及path-to-regexp库的依赖，获取到导航栏ID，实现导航栏高亮功能。

1. 路由配置规则说明

同级页面，路由层级相同，即项目test中存在同层级的页面a和页面b的情况下，路由地址设置分别为/test/a和/test/b；假设b页面存在点击下一级的子页面c，则页面c路由地址设置为/test/b/a，以便能匹配到该子页面父级的菜单用于高亮显示。



图5.1.1.1 路由配置示例

1. 路由高亮匹配规则

获取当前路由地址pathname

路由配置地址数组routesConfig

导航栏数组menuConfig（均包含子节点）

遍历routesConfig，利用path-to-regexp库的pathToRegexp方法获取配置路由地址的匹配正则，匹配与当前页面pathname匹配的地址

遍历menuConfig，通过test()匹配获取高亮id

匹配？

返回selectdKeys

移除一层route，即页面c(/test/b/c)的匹配地址为/test/b，再次遍历

是

否

图5.1.1.2 路由匹配规则逻辑图

## 通用组件

### http请求

此组件对axios进行了多一层的封装，暴露出http.interceptor()、http.GET()、http.POST()三个方式。

1. http. interceptor (request,responseError)

request:用于处理参数的方法，如统一传token或者用户id可在这里处理；

responseError:拦截请求状态码的方法，如401的情况下跳转会登录页面。

1. http.GET(url,params,options)

url:接口地址；

params:json对象，传递给后台的参数数据；

options:请求的其他设置，如请求头的设置。

示例：

http.GET('http://www.xxx.com',{id:1},{

headers: {

'Content-Type':'application/x-www-form-urlencoded;charset=utf-8'

}

})

.then(response=>response.json())

.then(json=>{

//handle things

})

1. http.POST(url,params,options)

url:接口地址；

params:json对象，传递给后台的参数数据；

options:请求的其他设置，如请求头的设置。

示例：

http.POST('http://www.xxx.com', { id: 1 }, {

headers: {

'Content-Type': 'application/x-www-form-urlencoded;charset=utf-8'

}

})

.then(response => response.json())

.then(json => {

//handle things

})

### 权限模块

用户登录的成功后，基于登录用户获取其所拥有的权限，存储于context 中，需要权限控制的组件通过判断其权限是否存在于登录成功后所获取的权限，来控制该组件的权限。

1. 在配置文件中传入获取权限的接口：

url：获取权限的接口

permission: {

url: '/permission.json' //获取权限接口

}

接口数据格式为：

{

"data": [

"/text/permission",

"/text/permission1"

]

}

1. 引入 withPermission ，通过高阶组件方式将权限存储于context 中

export default

withRouter(withPermission(config.permission)(BasicLayout));

1. 使用权限组件来控制权限

url: 权限标志（接口url）

<IsPermission url='/text/permission'>

<button>权限是否存在</button>

</IsPermission>

# 附录

（待补充）