开发前必读

开发文档阅读说明

1. API接口开放了丰富的接口说明，开发者可以借助接口能力，实现FiData的二次开发，通过目录导航可以快速预览，目录树按功能块聚合归类，如数据接入接口，API网关服务接口等。
2. 文档的阅读次序，建议先阅读一遍**开发指南**，并关注如何**获取token**，然后就可以独立查看各个功能块的**API接口文档**说明。

一、项目目录简介

注：仅对部分重点目录进行介绍。在本文档后续的介绍过程中，约定始终位于项目根目录(./)

./src——项目最重要的文件夹，代码及图片资源等都位于本文件夹

- ./src/api——存放不同功能模块的请求接口
- ./src/assets——静态资源文件夹，存放图片及字体资源
- ./src/components——跨模块组件的存放位置（如：数据接入及数仓建模同时使用了 Example.vue 组件，则将 Example.vue 文件存入本文件夹）
- ./src/icons——svg图标存放位置
- ./src/layout——项目布局组件的存放位置，例如侧边栏组件及导航条组件等
- ./src/mixin——跨模块组件的混入代码内容
- ./src/router——动态路由及静态路由的配置文件存放位置
- ./src/store——vuex状态管理文件的存放位置。注：本项目已对./src/modules文件夹做出批量化导入处理。
- ./src/styles——全局样式文件及跨模块样式文件的存放位置
- ./src/utils——工具类及对axios进行封装的文件的存放位置。注：不同模块的axios封装处理不同，因此产生了多个封装文件
- ./src/views——项目组件文件的存放位置

二、项目路由管理

1. **动态路由管理：**
   - 本项目的路由文件位于 ./src/router/index.js。在 index.js 文件中，componentMap 为动态路由对象，若需要在导航菜单中额外添加项目，则在 componentMap 对象中新增菜单属性，并懒加载导入菜单指向的文件地址即可。例如：example: () => import ('example.vue的地址')，另外，新增的菜单需要在“权限配置-服务授权”中设置角色权限方可访问。
2. **静态路由管理：**
   - 同样是 ./src/router/index.js 文件，该文件中的 constantRoutes 为静态路由数组，登陆页路由、首页路由、404页面路由等都位于本数组中，若需要在项目中添加额外的静态路由，仿照数组中已存在的元素对象进行新建即可。

三、接口调用流程

接口调用流程梗概图：

1. **请求方式及请求地址**
   a. 本项目的常用请求方式有4种：GET/POST/PUT/DELETE（http协议）；
   b. 除登录接口及获取项目标题的接口之外，本项目所有接口均需要token；
   c. 请求地址：`http://XXXXX:41004/` （注：各个环境的网关地址）
2. **封装axios**
   a. 对axios的封装文件位于 ./src/utils 目录下，一般的命名规则为 “xxxRequest.js” 。
   b. 不同模块有不同的封装文件。
3. **获取token**
   a. 本项目的前端采用了“js-cookie”库操作cookie，当登录成功后，后端会将token返回到前端，此时前端会将token传入cookie中。设置及获取token的代码详见 ./src/utils/auth.js 文件。
   b. 通过导入 ./src/utils/auth.js 文件，即可使用 getToken() 方法获取token
4. **请求拦截及响应拦截**
   a. 封装文件中编写了请求拦截及响应拦截代码段，可给每个请求附加 token，并对响应结果做出预先处理

5. **在具体的api文件中引入封装好的axios文件**
    a. api接口文件只需要引入该模块的封装好的axios文件，并以导出函数的形式对外暴露接口即可；
    b. api接口文件中的函数内容相对固定，函数内仅需要返回一个axios封装方法，并在方法中传入url、methods、data(可选)等配置项即可
6. **api文件导出api接口**
    a. 在组件文件中，只需要导入api文件中的函数，即可调用接口
7. **请求状态判断及响应数据获取**
    a. 请求状态码详情参考百度百科：https://baike.baidu.com/item/HTTP/243074?fr=aladdin
    b. 如果请求成功，则后端会返回固定格式的响应数据，其中：code 表示响应状态，code为0时表示响应正常，非0则表示响应异常，code可作为调用是否成功的判断依据；msg表示响应语句；data表示响应的数据
8. **具体的请求案例请查看"API接口文档"**

四、组件数据流转

本项目采用Vue框架进行开发，因此在项目中，组件间的数据传输方式大致有以下几种：

1. **父子组件间：props & emit**
    a. 父组件在子组件标签中添加要传输的数据属性，子组件采用 props 配置项进行接收；子组件采用 emit，触发父组件的自定义请求方法
2. **任意组件间：Bus**
    a. Bus是发布订阅模式的运用，利用vue的自定义事件机制，在触发的地方通过emit向外发布一个事件，在需要监听的页面，通过on监听事件
3. **任意组件间：Vuex**
    a. Vuex 是一个专为 Vue.js 应用程序开发的**状态管理模式 + 库**。它采用集中式存储管理应用的所有组件的状态，并以相应的规则保证状态以一种可预测的方式发生变化

五、基本调试方法

可以使用Postman等工具排查问题原因。

六、注意事项

更改页面时请勿在原文件内修改，而应新建一个同名文件，并附带版本号，版本号和文件名之间用短横线"-"分割。例如：Example-v2.0.vue