# 前端项目

## 1.运行前提

node.js 版本 v14.20.0 或者更高版本

这个项目里的配置文件是对应这套开源代码的管理端服务地址，需要先在本地环境中，将管理端的代码运行起来，才能运行起来看到页面效果。

## 2.本地运行

打开 cmd，或者使用开发工具 vscode

进入 vue\y9vue-dataFlowEngine 文件夹

执行 npm i 全局依赖。

依赖安装完成后，执行 npm run serve 运行程序。

前端工程默认使用 7070 端口。

如果需要部署到其他环境，可以执行 npm run build 或者 npm run build-test 打包部署至正式或者测试环境，需要修改.env.production 或者.env.test 中的配置

## 3.访问服务

运行成功后，访问前端页面 http://localhost:7070/y9vue-dataFlowEngine/

单机部署默认账号：

用户名：admin　　密码：admin

集成数字底座，账号使用底座的用户

# 后端项目

## 1.运行前提

安装 jdk11，开发工具：idea 或者 eclipse

## 2.本地运行

将项目导入到开发工具里

修改管理端工程的配置文件 application.yml，将 mysql 等服务端口和密码，改为自己的服务地址和密码。修改执行端配置文件 application 里的管理端运行服务地址

管理端默认使用 7055 端口

Maven 打包完成后，运行管理端和执行端工程：管理端默认采用外部 Tomcat 启动方式，执行端采用 springboot 内嵌 Tomcat 启动方式，可自行调整启动方式。

## 系统运行分为两种模式：

## 一、单机使用模式

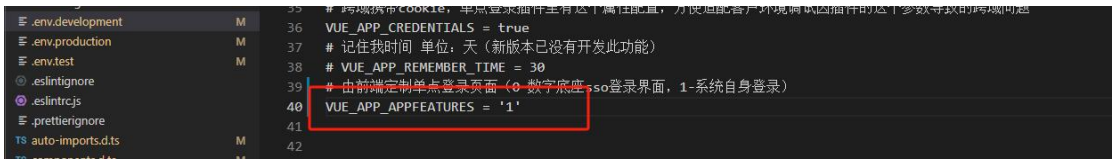
1. 后端工程 risenet-y9boot-support-risedata-scheduler 工程的 pom.xml 文件使用 risedata-security 包，如下图

```
<!-- 登录与权限：用数字底座的登录权限依赖y9kernel包，独立使用自身的功能依赖risedata包 -->
<dependency>
    <groupId>net.risesoft</groupId>
    <artifactId>risenet-y9boot-support-risedata-security</artifactId>
    <version>${project.version}</version>
</dependency>
<!-- <dependency>
    <groupId>net.risesoft</groupId>
    <artifactId>risenet-y9boot-support-y9kernel-security</artifactId>
    <version>${project.version}</version>
</dependency> -->
```

2.后端工程 risenet-y9boot-webapp-risedata-manager 工程的 application.yml 文件里将 y9.feature.oauth2.resource.enabled 属性设为 false，如下图

```yaml
    oauth2:
        resource:
            enabled: false
            saveLogMessage: false
            saveOnlineMessage: false
            allowBasicAuthentication: true
            allowFormEncodedBodyParameter: true
            allowUriQueryParameter: true
            protectedUrlPatterns: /api/*
            opaque:
                client-id: clientid
                client-secret: secret
```

3. 前端工程配置文件.env.development 里的 VUE_APP_APPFEATURES = '0'改为'1'，如下图




## 二、集成数字底座使用模式

1. 后端工程 risenet-y9boot-support-risedata-scheduler 工程的 pom.xml 文件使用 y9kernel-security 包，如下图

```xml
<!-- 登录与权限：用数字底座的登录权限依赖y9kernel包，独立使用自身的功能依赖risedata包 -->
<!-- <dependency>
    <groupId>net.risesoft</groupId>
    <artifactId>risenet-y9boot-support-risedata-security</artifactId>
    <version>${project.version}</version>
</dependency> -->
<dependency>
    <groupId>net.risesoft</groupId>
    <artifactId>risenet-y9boot-support-y9kernel-security</artifactId>
    <version>${project.version}</version>
</dependency>
```

2. 后端工程 risenet-y9boot-webapp-risedata-manager 工程的 application.yml 文件里修改 oauth2 属性，改成使用的数字底座的 sso 地址，如下图

```
oauth2:
    resource:
        enabled: true
        saveLogMessage: false
        saveOnlineMessage: false
        allowBasicAuthentication: true
        allowFormEncodedBodyParameter: true
        allowUriQueryParameter: true
        protectedUrlPatterns: /api/*
        opaque:
            client-id: clientid
            client-secret: secret
            introspection-uri: https://demo.youshengyun.com/sso/oauth2.0/introspect
            profile-uri: https://demo.youshengyun.com/sso/oauth2.0/profile
```

3. 将下图标红的地址改为数字底座接口调用的地址，具体地址按自己使用的来改

```
y9:
    systemName: dataflow
    internalIp: 192.168.x.x,10.0.x.x,172.20.x.x,10.192.x.x
    service:
        org:
            directUrl: https://t  un.com/platform
    common:
```

4. 前端工程配置文件.env.development 里的 VUE_APP_APPFEATURES = '0'，并且修改 sso 配置，与后端的数字底座的 sso 地址要保持一致，如下图

```
# sso接口
VUE_APP_SSO_DOMAINURL = 'https://demo.youshengyun.com/'
# sso接口上下文
VUE_APP_SSO_CONTEXT = 'sso'
# sso授权码接口（必须带上主域名，方便调试，避免拼接出现问题）
VUE_APP_SSO_AUTHORIZE_URL = 'https://demo.youshengyun.com/sso/oauth2.0/authorize'
# 退出URL，service的值，在这里可不带上（则由后端默认），也可带上（则由前端默认），可在退出时传入这个值给到单点登录插
VUE_APP_SSO_LOGOUT_URL = 'https://demo.youshengyun.com/sso/logout?service=http://localhost:7070/'
# 单点登录客户端唯一标示
VUE_APP_SSO_CLIENT_ID = 'clientid'
# 单点登录客户端密钥
VUE_APP_SSO_SECRET = 'secret'
# sso接口的固定字段
VUE_APP_SSO_GRANT_TYPE = 'authorization_code'
# token 缓存到前端的key属性
VUE_APP_SSO_SITETOKEN_KEY = 'y9AT'
```