

可视化平台部署方式

一、简述

该平台采用前后端分离模式开发

- 前端采用vue + elementUl ,主要采用vue-admin-template的模版结构开发
- 后端依旧采用springboot开发

二、部署

由上可得出部署应当分前后端分别部署。

项目总体结构如下:



2.1 前端部署

前端部署因为采用vue,所以依赖于node.js环境。node.js环境安装请自行百度。以下是部署步骤:

• 进入前端工程所在路径运行:

```
npm install
```

• 依旧在同一个目录运行:

```
npm run dev
```

成功之后会有如下图的提示:

```
You may use special comments to disable some warnings.
Use // eslint-disable-next-line to ignore the next line.
Use /* eslint-disable */ to ignore all warnings in a file.

App running at:
- Local: http://localhost:9528/
- Network: http://172.18.91.193:9528/
```

2.2 后端部署

由于后端采用springboot开发,可直接打jar包运行,如果想用tomcat打war包部署,则需要自行解决。

由整体结构图可看出,后端主要代码在data-display-server工程中,该后端工程依赖同级工程 data-display-commons-util,因此打包可直接在server工程中运行。

如下所示:

第一步: 安装依赖。在主工程下(也就是整个代码的父目录),运行如下命令先把依赖安装到本地。

```
mvn install
```

• 第二步:编译打包。进入data-display-server文件夹,运行如下命令:

```
mvn package
```

• 第三步:运行jar包。运行成功之后,进入到data-display-server/target文件夹下,发现有一个datadisplay-server-0.0.1.jar,以java形式运行该jar包即可

```
nohup java -jar datadisplay-server-0.0.1.jar > log.file 2>&1 &
```

或者不进入target目录运行:

```
nohup java -jar target/datadisplay-server-0.0.1.jar > log.file 2>&1 &
```

两种命令等效。

• 第四步: 查看运行日志。

```
tail -200f log.file
```

备注:

后端代码部署时,如果想修改数据库地址,在application-*.properties中。或者自己新建额外的 application-*ABC.properties,那么修改application.properties中的active profile即可。

如下图所示:

```
≪ O Run
                     🗏 🕃 🔻 💴 🛘 🖟 MysqlOperati... 🖟 OracleOperat... 🖟 MongodbOper... 🎤 application... 😭 JdbcTemplate... 😭 DataSource.c... 🖟 DataDisplayS... 🗡 application... 🗵 🖟 logback.xml 🤻 🕏

    # com.tianque.datadisplay.serve

                                              server.port=
                                             spring.profiles.active
        ⊕ domain
                                            spring.datasource.password=Admin@1234
        # enums
        # exception
                                           6 spring.datasource.type=com.alibaba.druid.pool.DruidDataSource
                                            spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
                                           8 spring.datasource.max-idle=10
                                           9 spring.datasource.max-wait=10000
         DatabaseInfo.java
                                       a 10 spring.datasource.min-idle=5
a 11 spring.datasource.initial-size=5
         DatabaseType.java
        Databaserype:java

DataComingSource.java

GlobalParam.java

restconller.impl

restcontroller
                                          12 spring.datasource.druid.time-between-eviction-runs-millis=60000
                                          13 spring.datasource.druid.min-evictable-idle-time-millis=300000
                                           4 spring.datasource.druid.validation-query=SELECT 1 FROM DUAL
        # service
                                       15 spring.datasource.druid.test-while-idle=true
       16 spring.datasource.druid.test-on-borrow=false

MongodbOperationImpl.java
MySqlOperationImpl.java
CoracleOperationImpl.java
CoracleOperationImpl.java
                                       17 spring.datasource.druid.test-on-return=false
18 spring.datasource.druid.pool-prepared-statements=true
                                       § 19 spring.datasource.druid.max-pool-prepared-statement-per-connection-size=20
                                       20 spring.datasource.druid.filters=stat,wall,log4j,logbak
          > 

DatabaseOperationIntf.java
                                         21 spring.application.name
         ☑ DataDisplayServerApplication.java
                                        🔐 Problems 🍭 Javadoc 🗟 Declaration 🖋 Search 😉 Console 🕮 😇 Progress 🚳 Boot Dashboard 🦫 Call Hierarchy 🦃 P3C Results 🍣 Rule I

    ⇔ data

                                        DataDisplayServerApplication [Java Application] C\Program Files\Javajdk1.8.0 161\bin\javaw.exe [2019年11月14日上午9:30:43)
[2019-11-14 09:32] [DEBUG] springfox.documentation.spring.web.readers.parameter.ModelAttributeParameterExpander
                                        [2019-11-14 09:32][DEBUG] springfox.documentation.spring.web.readers.parameter.ModelAttributeParameterExpander
                                        [2019-11-14 09:32][DEBUG] springfox.documentation.spring.web.readers.parameter.ModelAttributeParameterExpander
      application-dev.properties
                                        [2019-11-14 09:32][DEBUG] springfox.documentation.spring.web.readers.parameter.ModelAttributeParameterExpander
      application-product.properties
                                        [2019-11-14 09:32] [DEBUG] springfox.documentation.spring.web.readers.parameter.ModelAttributeParameterExpander
       application-test.properties
application.properties
                                        [2019-11-14 09:32][DEBUG] springfox.documentation.spring.web.readers.parameter.ModelAttributeParameterExpander
                                        [2019-11-14 09:32][DEBUG] springfox.documentation.spring.web.readers.parameter.ModelAttributeParameterExpander
                                        [2019-11-14 09:32][DEBUG] springfox.documentation.spring.web.readers.parameter.ModelAttributeParameterExpander
      logback.xml
                                      [2019-11-14 09:32][DEBUG] springfox.documentation.spring.web.readers.parameter.ModelAttributeParameterExpander
```

2.3 前后端对接

既然采用前后端分离模式开发,那么前后端需要进行对接。交互模式采用http方式,因此需要前端知道后端代码的访问url。

后端部署之后,一般可通过日志查看运行端口,如下图所示:

```
[2019-11-14 09:32][DEBUG] springfox.documentation.swagger.readers.operation.VendorExtensionsReader - Extension count 1 for method update [2019-11-14 09:32][DEBUG] springfox.documentation.swagger.readers.operation.SwaggerOperationResponseClassReader - Setting response class [2019-11-14 09:32][DEBUG] springfox.documentation.swagger.readers.operation.OperationAuthReader - Authorization count 0 for method updat [2019-11-14 09:32][INFO] org.apache.coyote.http11.Http11NioProtocol - Starting ProtocolHandler ["http-njo-8030"] [2019-11-14 09:32][INFO] com.tianque.datadisplay.server.DataDisplayServerApplication - Started [ataDisplayServerApplication in 25.504 3]
```

可看出该后端应用启动在8030端口,那么本地访问路径为:

```
http://localhost:8030
```

外界访问路径为:

```
http://ip地址:8030
```

当然,纯后端没有做web,所以无法通过页面访问。

前端对接后端地址在前端工程的src/configs/global.js中控制:如下图所示,如果后端代码变更,直接修改此处即可。

