

后端接口说明

一、概要说明

原有的TwoRavens后端是由庞大的javaee组件以及复杂的数据库表组成的。在此基础上 我们抽离出相关计算的主要逻辑,改用现在比较流行且易于维护的Spring Boot作为后端框架。另外,基于TwoRavens已经是比较完善的独立的前端界面,后端会采用RESTful设计规约保证前后端分离。

文件主要分为两个目录: java目录和resources目录。

- java目录下存储的是java代码 (java.com.monetware.demo) ;
 - controller: 提供前端请求的接口
 - prep: prep文件生成代码
 - xml: xml文件生成代码
 - variable: 全局变量的存储位置
- resources文件夹下存储的是prep,tab和xml文件, 以及R语言脚本。
 - tab: 存放预先的tab文件
 - xml: 存放生成的xml文件
 - prep: 存放生成的prep文件
 - application.properties: spring boot配置文件
 - 注意: 生成的tab, xml和prep文件都是名字都是一一对应的, 例如 abcd.tab文件所对应是abcd-ddi.xml和abcd.prep文件

二、环境配置

- Java SE Development Kit 8
- R version 3.3.2 或更高版本
 - Rserve 1.7.3 或更高版本D
 - 其他依赖包可参考TwoRavens的说明, 如下:
 - <https://github.com/IQSS/TwoRavens/blob/master/r-setup/package-versions.txt>
- IntelliJ Idea 或 eclipse IDE工具 (暂时需要)

三、运行说明

- 使用命令行输入 R CMD Rserve开启 Rserve
- 使用IntelliJ或者eclipse打开demo文件夹下的pom.xml文件, 该项目所需依赖会自动导入到本机中
- 需要更改文件一些参数: (提交的虚拟机里的参数不用改)
 - a. 更改 java.com.monetware.demo.variable.root.java里的**FileRoot** (根目录), 改为resources.static.files文件夹的 **绝对路径**
 - b. 更改 java.com.monetware.demo.prep.RemoteDataFrameService.java里的**RSERVE_TMP_DIR**变量为服务器上任意文件夹, 该文件夹下存放生成.prep文件时的中间文件

- 点击运行按钮，运行DemoApplication类中的main()函数。运行后Spring Boot内置的Tomcat会自动运行（内置Tomcat配置文件是resources文件目录下的application.properties）
- 至此，后端便已开启

四、接口说明

url: **create/xml?name={test}**

接口描述：
此接口通过get方法请求生成名字为 **test** 的xml文件： **test.xml**

Request Method	GET
----------------	-----

请求参数

名字	位置	描述	是否必须	类型
name	url的参数	放在GET请求中最为参数传入后端，生成对应的{test}-ddi.xml文件	否（默认名字为 test）	String

返回结果

返回值	描述	类型
success	表示生成文件成功	String
file already exist	文件已经存在	String
no such a file	对应的.tab文件不存在	String

实例请求
localhost:8080/create/xml?name=hello

示例返回
no such a file! (因为并没有hello.tab文件)

url: **create/prop?name={test}**

接口描述：
此接口通过get方法请求生成名字为 **test** 的prop文件： **test.prop**

请求参数

名字	位置	描述	是否必须	类型
name	url的参数	放在GET请求中最为参数传入后端，生成对应的{test}.prop文件	否（默认名字为 test）	String

返回结果

返回值	描述	类型
unexpected error	如果解析tab文件出现问题（没有对应文件或者文件内容有误）	String
success	创建成功	String

实例请求

localhost:8080/create/prep?name=test

示例返回

success

url: **download/xml?name={test}**

接口描述:

此接口通过get方法下载名字为 **test** 的xml文件: **test-ddi.xml**。若没有此文件则返回文件大小为0

请求参数

名字	位置	描述	是否必须	类型
name	url的参数	放在GET请求中最为参数传入后端，下载对应的xml文件	否（默认名字为 test）	String

返回结果

返回值	描述	类型
下载窗口	此动作描述了一次完整的文件下载动作，既可以单独输入url进行文件下载，其本身也作为tworavens运行时的get请求被写在了app_ddijs里	null

url: **download/prep?name={test}**

接口描述:

此接口通过get方法下载名字为 **test** 的prep文件: **test.prep**。若没有此文件则返回文件大小为0

请求参数

名字	位置	描述	是否必须	类型
name	url的参数	放在GET请求中最为参数传入后端，生成对应的test.xml文件	否（默认名字为 test）	String

返回结果

返回值	描述	类型
下载窗口	此动作描述了一次完整的文件下载动作，既可以单独输入url进行文件下载，其本身也作为tworavens运行时的get请求被写在了app_ddijs里	null

url: **download/tab?name={test}**

接口描述:

此接口通过get方法下载名字为 **test** 的tab文件: **test.tab**。若没有此文件则返回文件大小为0

请求参数

名字	位置	描述	是否必须	类型
name	url的参数	放在GET请求中最为参数传入后端，生成对应的{test}-ddi.xml文件	否（默认名字为 test）	String

返回结果

返回值	描述	类型
下载窗口	此动作描述了一次完整的文件下载动作，既可以单独输入url进行文件下载，其本身也作为tworavens运行时的get请求被写在了app_ddi.js里	String

五、注意事项和问题

- 由于spring boot自身的目录结构和我们单独完成的java代码目录结构有所差异，某些静态资源的路径在项目迁移后需要更改。具体说明已经写到了 概要说明里
- 在app-ddi.js里更改了 xml、prep和tab文件的下载路径，还没有写前端用户输入文件名字的接口，默认为"test":
 - 221行: dataurl = "http://localhost:8080/download/tab?name=test"
 - 210行: metadataurl = "http://localhost:8080/download/xml?name=test"
 - 232行: pURL = "http://localhost:8080/download/prep?name=test"
- 不太确定虚拟机中的 TwoRavens 的 app-ddi.js 在原有基础上做过哪些改动？
 - a. 已经发现了在解析.prep时去掉了private:和variables字段并改回原有版本
 - b. 在点击右上角变量计算时，无法正常加载出新的变量