

Big-data-homework2设计说明书

任务概述

目标

简单版Spark SQL查询分析器

运行环境

windows, jdk1.8, scala11.8, vue

需求概述

- 可以配置SparkSQL连接信息
- 列出当前连接下的库、表、字段，以方便编写SQL脚本
- 提供SQL编辑框，方便编写SQL和提交查询任务
- 以列表的形式，展示查询结果

总体设计

总体结构和模块外部设计

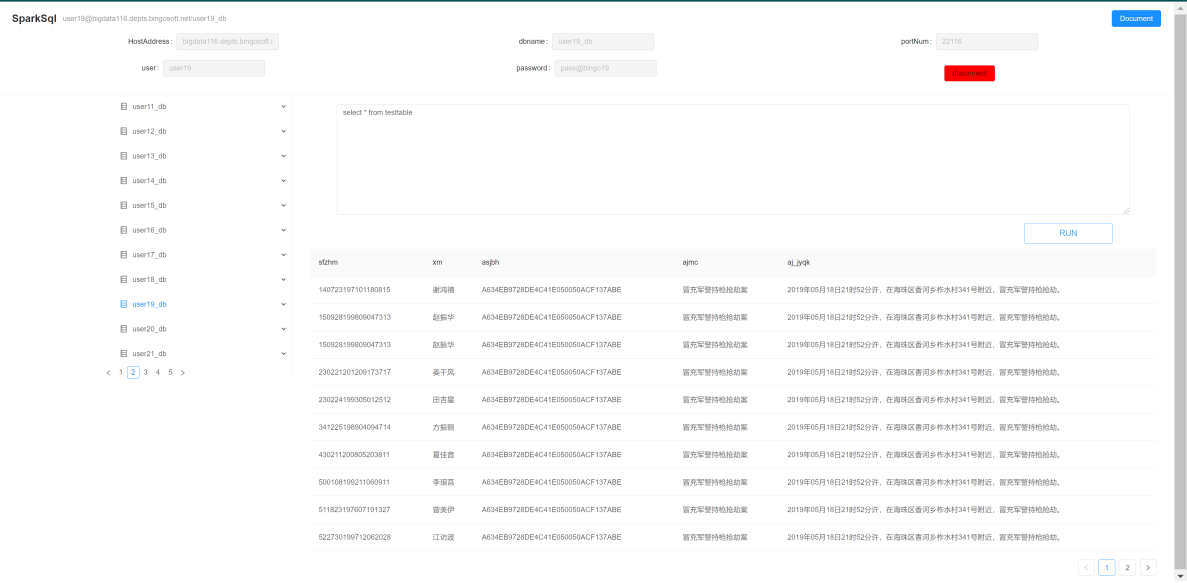
前端设计

Vue.js （ant design）

整体页面分为三个部分：

- 建立连接区
- 数据库信息区
- 查询及结果区

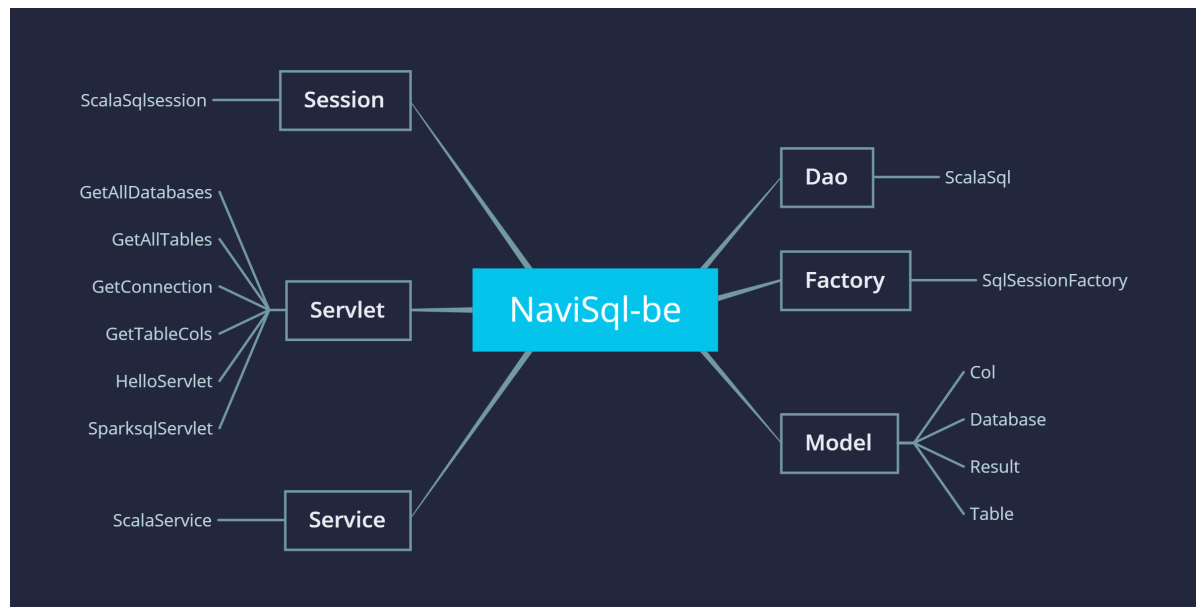
其中数据库信息区和查询及结果区只有处于连接成功状态才显示出来



后端设计

java1.8+scala11.8

利用scala与hive数据库进行sql操作，java servlet将程序封装对外开放接口



模块功能

Dao 定义sql语句

ScalaSql

ScalaSql负责dao层的操作，使用scala语言调用session执行sql的接口，该层主要负责具体sql语句的定义。包括获取所有表，所有数据库，所有字段数据，执行sql语句。

Servlet 定义网络接口

GetAllDatabases

继承HttpServlet，负责开放获取所有数据库的接口，传入参数包括连接基本信息，返回在该链接下所有数据库的名称，数据封装成Col数据，再封装成Result返回

GetAllTables

继承HttpServlet，负责开放获取某数据库下所有表的接口，传入参数包括连接基本信息以及数据库名称，返回在该链接下改数据库的包含的所有表信息，数据封装成List

数据，再封装成Result返回

GetConnection

继承HttpServlet，负责开放测试连接的接口，传入参数包括连接基本信息，利用ScalaSqlSession的isVaild()函数执行"select 1"若执行成功则返回"success"，否则利用try catch获取错误信息，将错误信息封装成Result返回

GetTableCols

继承HttpServlet，负责开放获取某数据表下所有字段信息的接口，传入参数包括连接基本信息，返回在该链接下所有数据库的名称，数据封装成Table数据，再封装成Result返回

SparksqlServlet

继承HttpServlet，负责开放执行sql查询语句并返回结果的接口，传入参数包括连接基本信息以及sql语句，返回sql查询语句的结果，数据封装成Table数据，再封装成Result返回，若sql语句执行失败则用Result.fail()返回错误信息，前端根据返回结果的状态码判断sql语句是否执行成功

Factory 调度SqlSession

SqlSessionFactory

利用工厂+单例模式进行hive数据库连接的统一调度。SqlSessionFactory生成采用懒汉加载的单例模式，全局唯一。核心成员为一个全局hashmap，这个hashmap保存生成的SqlSession，key值为连接基本信息，当有相同的连接信息请求是，检查现有SqlSession是否已关闭，若无则直接使用相同SqlSession，减少数据库连接的压力，否则再创建新的SqlSession。

Model 结构化数据

Col

结构化字段信息，包括字段名称，字段类型，字段备注，字段数据列表

Table

结构化表数据，包括表名称，字段信息列表

Database

结构化数据库数据，包括数据库名称，表信息列表

Result

结构化结果信息，包括状态码（200成功，400失败）以及具体数据

Service 业务层，定义具体的业务逻辑

ScalaService

定义scala数据的具体业务逻辑，包括获取所有表名称，获取所有表字段信息，获取所有数据库信息，获取所有表信息，执行sql查询语句。ScalaService在调用dao层接口后将结果封装成对应的col，table或list，再用result封装返回。

Session 会话层，scala，连接请求及查询请求

ScalaSqlSession

提供连接数据库，关闭连接以及检测信息是否有效的接口，一个ScalaSqlSession对应一个jdbc连接

处理流程

1. 前端和后端部署好后，在前端页面填写连接信息，点击连接后前端通过axios发送请求到后端对应端口，后端接收到请求后从ScalaSqlSession的测试接口检查连接信息是否有效，若有效前端得到状态码为200的数据，并将连接信息存在localStorage，之后发送请求时都会利用该连接信息建立数据库连接；若无效则受到状态码为400的数据，并页面浮现通知，展示错误信息。
2. 建立连接成功后查询该链接下的所有数据库名称，前端得到所有数据库名称的Col，读取列表上的所有名称并以每页10条数据的分页格式展示在页面右方。
3. 当用户点击具体的数据库名称时前端向后端发送查询所有数据表的请求，前端接收到List

的数据，将该数据库的所有表列出，当用户点击数据表时popover浮现出该表的所有字段信息。

- 用户输入sql查询语句进行查询后，后端利用ScalaSession得到的jdbc连接执行executeQuery函数，将结果以List

的数据格式返回，前端接收到结果后将数据以每页十条的格式以表格的形式呈现，若sql执行出错则将错误信息通过popover浮现出来。

功能实现

- 配置SparkSQL连接：通过Vue.js编写前端代码，提供输入框填写连接信息以及连接按钮，当用户进行连接操作时后端利用"select 1"语句模拟查询，若成功则视为连接信息有效，存在localStorage中，之后每次连接请求都带上这个连接信息若测试连接失败则popover浮现出错误信息。
- 列出当前连接下的库，表，字段：通过Vue.js编写前端代码，提供展示数据库信息，表信息以及字段信息的视图；后端通过"show tables"获取数据表信息，"show databases"获取数据库信息，"desc formatted"获取字段信息，然后通过ResultSetMetadata获取结果表中的字段信息，然后从RsultSet中通过字段信息按行读取结果。
- 提供SQL编辑框：通过Vue.js编写前端代码，提供编写SQL语句的testarea。后端利用ScalaSession得到的jdbc连接执行excuteQuery函数，将结果以List

的数据格式返回。

- 以列表的形式展示查询结果：通过Vue.js编写前端代码，将接收到结果后将数据以每页十条的格式以表格的形式呈现，若sql执行出错则将错误信息通过popover浮现出来。