|  |
| --- |
| **General Data Query Tool Design** |

Revise History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Description | Author |
| 2025/8/3 | 0.1 | Assignment | Jack-Zhang |

**Tables of Content**

[**1** **Summary** 3](#_Toc205153118)

[**2** **Installation Guide** 4](#_Toc205153119)

[**2.1** **Running environment** 4](#_Toc205153120)

[**2.1.1** **Install Java** 4](#_Toc205153121)

[**2.1.2** **Install NodeJS and Angular** 4](#_Toc205153122)

[**2.1.3** **Install Git** 5](#_Toc205153123)

[**2.1.4** **Install IntelliJ** 5](#_Toc205153124)

[**2.1.5** **Maven (IntelliJ built-in Maven plugin)** 5](#_Toc205153125)

[**2.2** **Steps to start the applications** 5](#_Toc205153126)

[**2.2.1** **Load Source Code from Git** 5](#_Toc205153127)

[**2.2.2** **Start UI** 5](#_Toc205153128)

[**2.2.3** **Start micro service** 6](#_Toc205153129)

[**2.2.4** **Access and check** 6](#_Toc205153130)

[**3** **Design** 7](#_Toc205153131)

[**3.1** **UI Design** 7](#_Toc205153132)

[**3.2** **Service Design** 8](#_Toc205153133)

[**3.2.1** **/general/data/query/datasource** 8](#_Toc205153134)

[**3.2.1.1** **JSON Payload** 8](#_Toc205153135)

[**3.2.1.2** **API Flow** 10](#_Toc205153136)

[**3.2.2** **/general/data/query/records** 10](#_Toc205153137)

[**3.2.2.1** **JSON Payload** 10](#_Toc205153138)

[**3.2.2.2** **API Flow** 12](#_Toc205153139)

[**3.3** **Database Design** 12](#_Toc205153140)

1. **Summary**

This tool built by Angular and Java spring-boot. Browser server architecture. Front-end and back-end separation mode. Development IDE is IntelliJ. Key deliverables: front-end source code, back-end source code, design document and an executable jar file.

1. **Installation Guide**
   1. **Running environment** 
      1. **Install Java**

JDK version as below, output of “java -version”:

openjdk version "18.0.1.1" 2022-04-22

OpenJDK Runtime Environment (build 18.0.1.1+2-6)

OpenJDK 64-Bit Server VM (build 18.0.1.1+2-6, mixed mode, sharing)

* + 1. **Install NodeJS and Angular**

**Node version as below, output of “node -v”：**

v22.17.0

**npm version as below, output of “npm -v”:**

10.9.2

**Ng version as below, output of “ng version”：**

Angular CLI: 20.1.4

Node: 22.17.0

Package Manager: npm 10.9.2

OS: win32 x64

Angular: 20.1.4

... build, cli, common, compiler, compiler-cli, core, forms

... platform-browser, platform-server, router, ssr

Package Version

------------------------------------------------------

@angular-devkit/architect 0.2001.4

@angular-devkit/core 20.1.4

@angular-devkit/schematics 20.1.4

@schematics/angular 20.1.4

rxjs 7.8.2

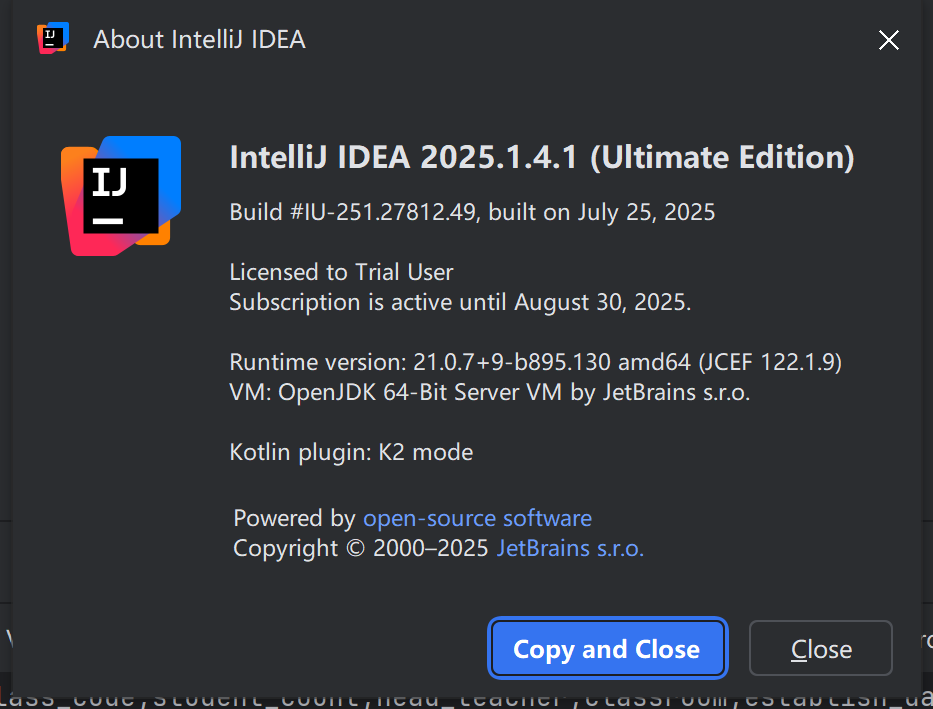
typescript 5.8.3

* + 1. **Install Git**

Git version as below, output of “git version”:

**git version 2.50.1.windows.1**

* + 1. **Install IntelliJ**



* + 1. **Maven (IntelliJ built-in Maven plugin)**
  1. **Steps to start the applications**
     1. **Load Source Code from Git**

[**https://github.com/Zhang-Albert/assignment/tree/main**](https://github.com/Zhang-Albert/assignment/tree/main)

**https://github.com/Zhang-Albert/assignment.git**

* + 1. **Start UI**

After source code downloaded, from command line go to directory：**general-data-query-ui,** then run following commands:

npm install

npm run start

* + 1. **Start micro service**

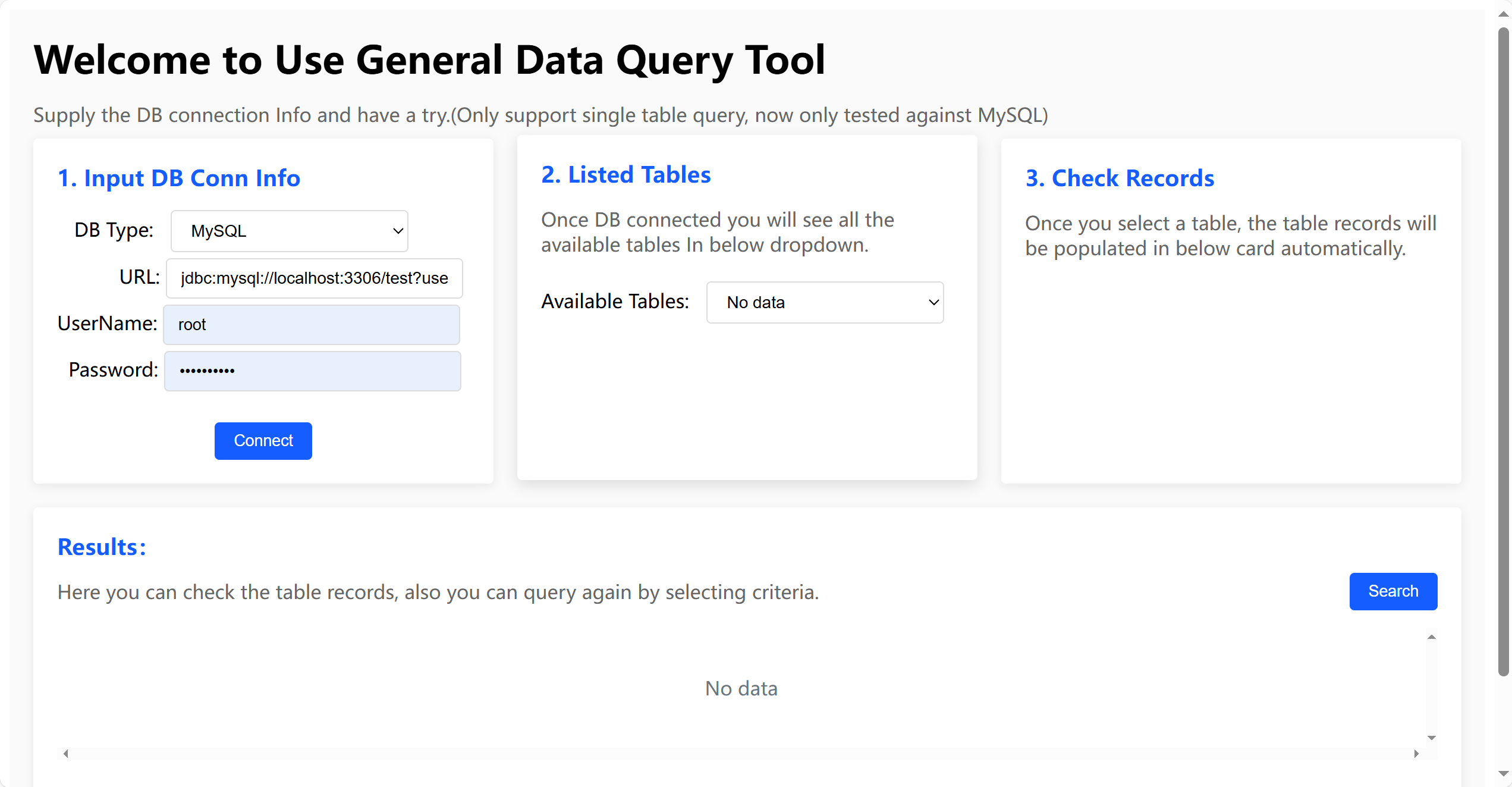
After source code downloaded, you can see a jar file: **query-0.0.1-SNAPSHOT.jar**, from command line just go to the directory that this jar reside in and start it by following command (alternatively you can start it from IDE):

**java -jar query-0.0.1-SNAPSHOT.jar**

* + 1. **Access and check**

[**http://localhost:4200/**](http://localhost:4200/)

**You will see below page, update the database connection info and try to connect to your MySQL and try it. (dummy\_data.sql is only for creating testing data in MySQL, if you have your own MySQL available you don’t need to execute the scripts)**

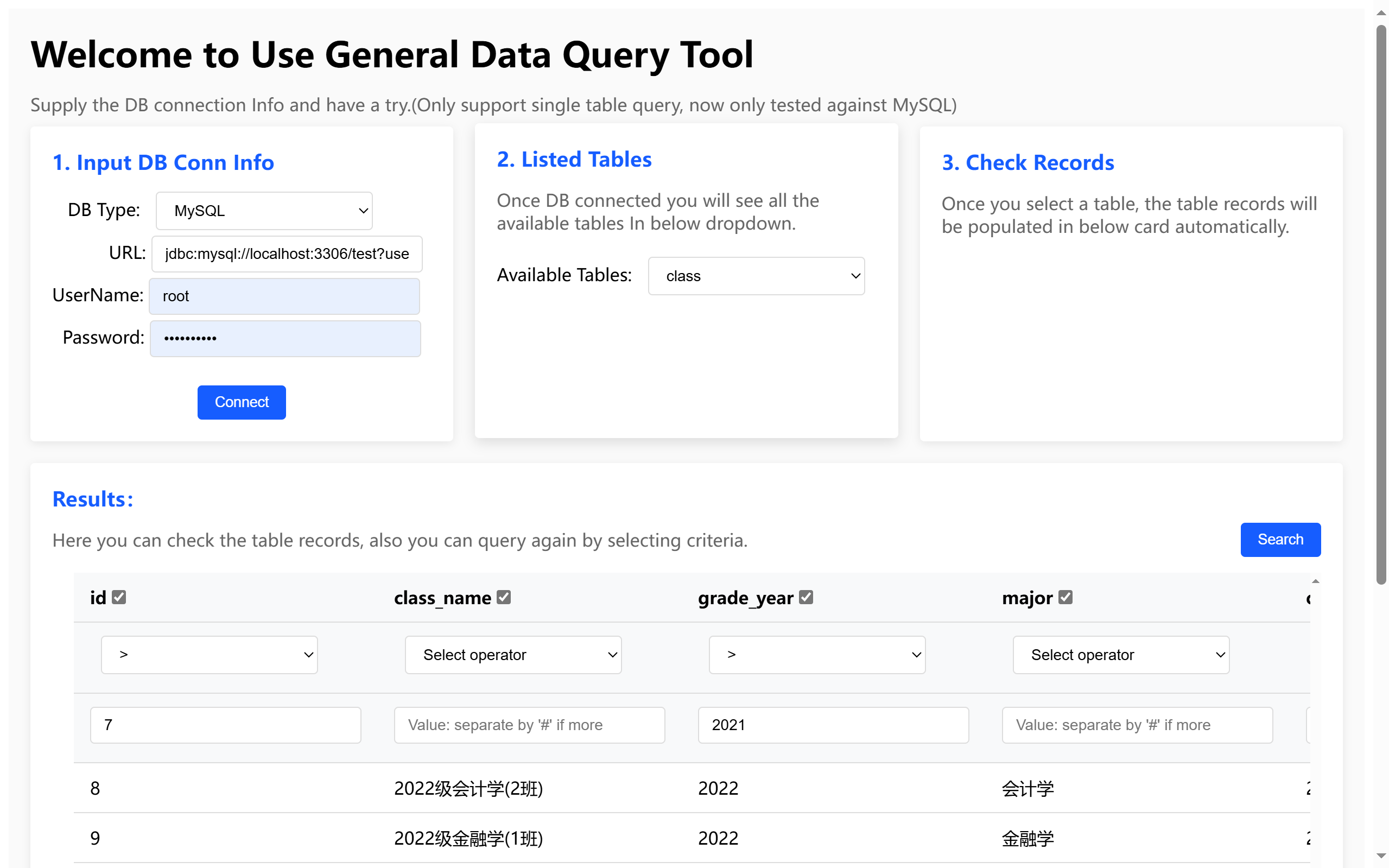


1. **Design**
   1. **UI Design**

User provide the database connection info, after click connect button, the available tables will be populated in card 2 and records also populated in below card.

You can switch table by re-select the dropdown in card2.

You can select which columns will be shown by clicking the checkbox close to the column title. You can also filter the records by select the operator and input the value. (the second and third line in the table)



* 1. **Service Design**

In order to support the UI, we designed two RESTful APIs that accept and produce JSON format data. If the service is started, we can also check the API specification by assessing swagger UI.

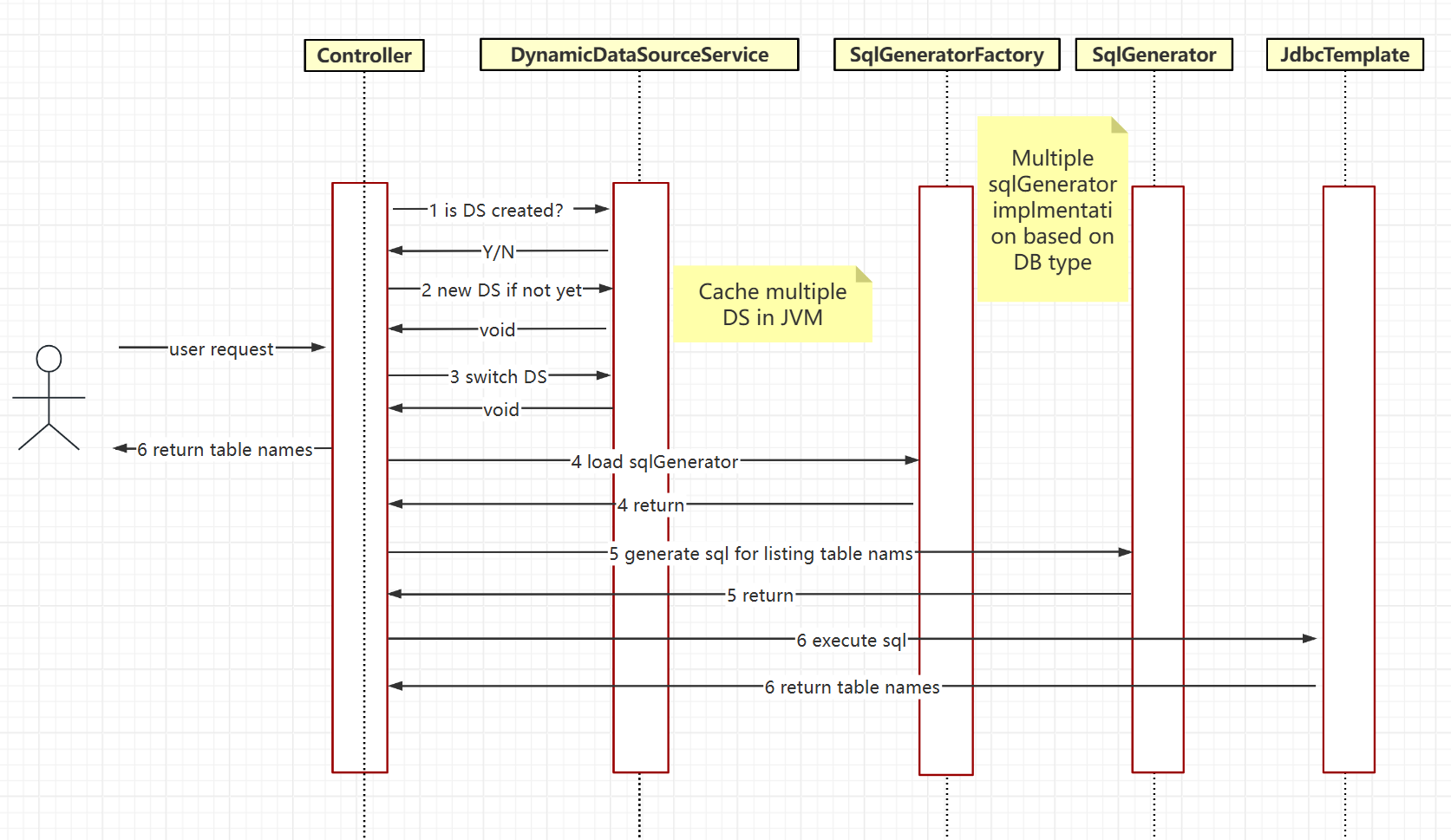
<http://localhost:8080/swagger-ui/index.html#/>

| Method | Path | Description |
| --- | --- | --- |
| POST | /general/data/query/datasource | Register new Data Source and Load all available tables. |
| POST | /general/data/query/tables/records | Search table records by criteria. |

* + 1. **/general/data/query/datasource**
       1. **JSON Payload**

| Request Payload | Response Payload |
| --- | --- |
| **{**  **"url": "string",**  **"userName": "string",**  **"password": "string",**  **"type": "string"**  **}** | **{**  **"retCode": "200",**  **"msg": "success",**  **"result": [**  **"table1",**  **“table2”**  **]**  **}** |

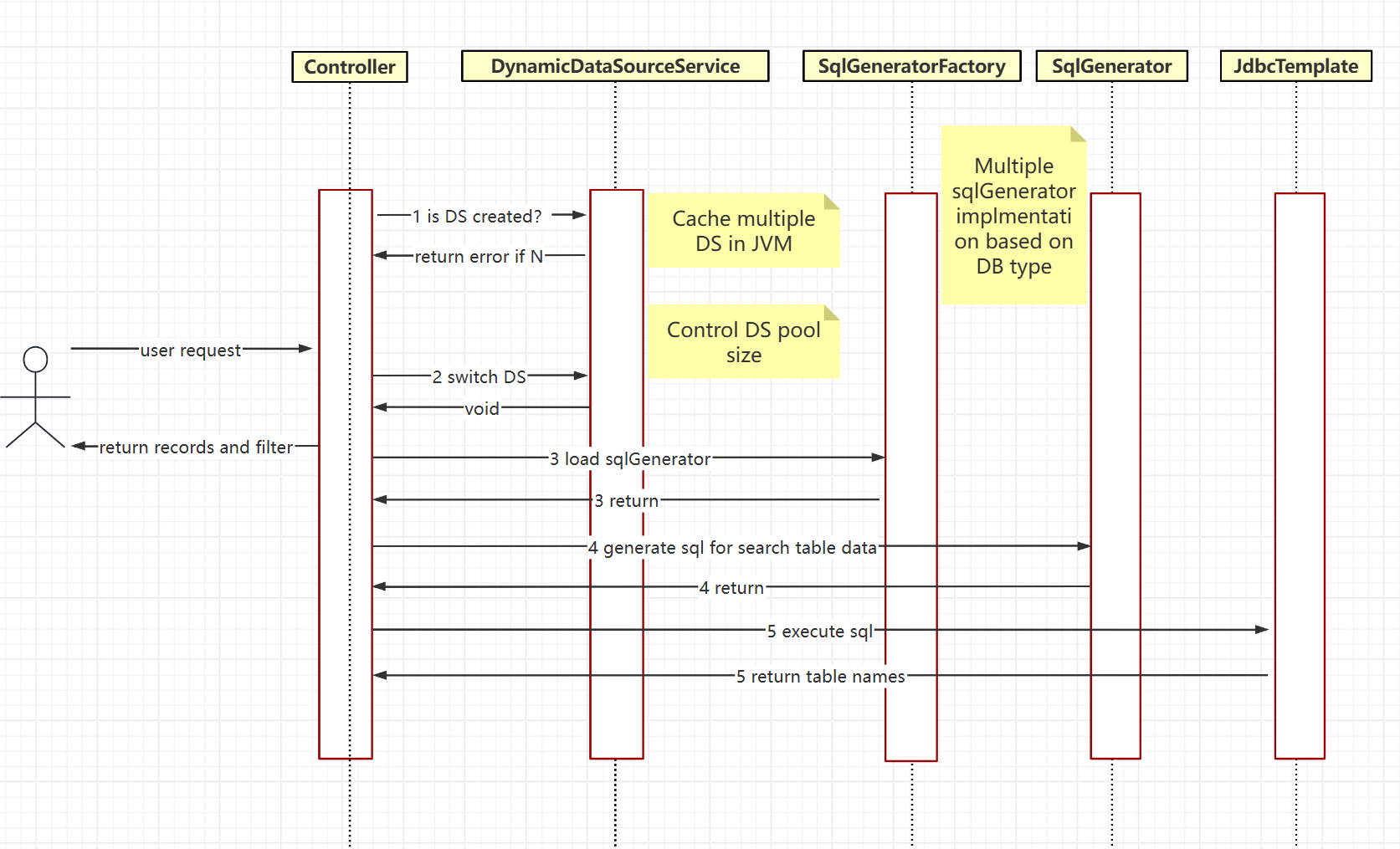
* + - 1. **API Flow**



* + 1. **/general/data/query/records**
       1. **JSON Payload**

| Request Payload | Response Payload |
| --- | --- |
| **{**  **"dbConnInfoDTO": {**  **"url": "string",**  **"userName": "string",**  **"password": "string",**  **"type": "string",**  **},**  **"pageSize": 0,**  **"pageNum": 0,**  **"tableName": "string",**  **"selectedColumnCriteriaValue": {**  **"additionalProp1": "string",**  **"additionalProp2": "string",**  **"additionalProp3": "string"**  **},**  **"selectedColumnOperation": {**  **"additionalProp1": "string",**  **"additionalProp2": "string",**  **"additionalProp3": "string"**  **},**  **"selectedColumnNames": [**  **"string"**  **]**  **}** | **{**  **"retCode": "200",**  **"msg": "success",**  **"result": {**  **"columns": [**  **"cols1",**  **“cols2”,**  **“cols3”**  **],**  **"records": [**  **{**  **"cols1": 1,**  **"cols2": “zhangsan”,**  **"cols3":” wangwu”**  **}**  **],**  **"totalNum": 0,**  **"totalPage": 0,**  **"pageNum": 0,**  **"pageSize": 0**  **}**  **}** |

* + - 1. **API Flow**



* 1. **Database Design**

This tool itself actually doesn’t need underlying database support. It only connects to specified database as needed. Scripts in the demo is only for initialize the sample testing data.