

Data.sql

To initialize a table with some dummy values using Spring JDBC and a ``data.sql`` script, you can follow these steps:

1. ****Create the ``data.sql`` Script****: In your ``src/main/resources`` directory, create a file named ``data.sql``. This file will contain SQL statements to insert dummy data into your table. For example:

```
```sql
INSERT INTO your_table (column1, column2, column3) VALUES ('value1',
'value2', 'value3');
INSERT INTO your_table (column1, column2, column3) VALUES ('value4',
'value5', 'value6');
-- Add more INSERT statements as needed
```
```

2. ****Configure Spring JDBC****: Make sure you have configured Spring JDBC in your application context. This typically involves setting up a ``DataSource``, ``JdbcTemplate``, and any other necessary beans.

3. ****Enable SQL Script Execution****: By default, Spring Boot will automatically execute the ``data.sql`` script during application startup. However, if you have disabled this behavior or need more control over when the script is executed, you can explicitly enable it in your ``application.properties`` file:

```
```properties
spring.datasource.initialization-mode=always
```
```

This will ensure that the ``data.sql`` script is always executed when the application starts.

4. ****Run Your Application****: Simply run your Spring Boot application, and the ``data.sql`` script will be executed automatically, inserting dummy data into your table.

5. ****Verify the Data****: After starting your application, you can verify that the dummy data has been inserted into your table by querying the database.

By following these steps, you can easily initialize your table with dummy values using a ``data.sql`` script and Spring JDBC.