IBM Developer Recipes Home All recipes My recipes

Manage Db2 spatial data with open sourc Spatial

David Adler

Published on January 10, 2019 / *Updated on November 9, 2020*

👁 835 ☆ 0 0

Contents

Overview

Ingredients

Setup Db2 and sample data

Setup development environment

Run sample program

ⓘ Recipes are community-created content. They are neither monitored nor endorsed by IBM. If you find inappropriate co Abuse to let us know. For more information on community content, please refer to our Terms of Use.

Overview

Skill Level: Intermediate

Assumes familiarity with Db2 and Java application development using Maven

Hibernate is an open source Java persistence framework for relational databases with a database-a
Hibernate Spatial is a component which adds the capabilities to work with spatial data.

Ingredients

- Db2 client and server(Db2 client & server download)
- Maven download
- Hibernate web site and documentation
- Hibernate Spatial User Guide
- Db2 sample data and programs (download)
- Db2 spatial documentation as appropriate for your environment:
  - Db2 for z/OS Knowledge Center
  - Db2 LUW Knowledge Center

With this recipe you will populate Db2 tables with spatial data and run a sample Java program that que
the Hibernate Spatial API.

Step-by-step

1 Setup Db2 and sample data

1. Note: this recipe uses Hibernate on Windows connecting to a local Db2 database**.** The pro
   and MacOS environments. Change the database connection values in the examples to you
   userid.
2. Download and unzip the Db2 sample data and programs to a convenient directory. Change
3. Open a Db2 command window where you can execute SQL statements. If you are using a
   sure it is cataloged locally. You can use the scripts **catalog-luw.sql** or **catalog-zos.sql** as a
   changes for the remote server location. (Use **catalog-luw.sql** for DashDB)
   You can execute the script with a command like:
   db2 -tvf catalog-luw.sql

4. Import sample spatial for banks, customers and San Jose streets.
   – Modify the scripts **import-luw.sql** or **import-zos.sql** as appropriate to change the userid
   procedure. (The scripts contain descriptions of each of the steps and any modifications ne
   – Execute the script with a command like:
   ```
   db2 -tvf import-luw.sql
   ```
   – Check that there were no processing errors
5. Verify the imported data with the SQL statement (change "**osuser**" to your connection use
   ```
   db2 select name, street, varchar(db2gse.st_astext(geom),32) from osuser.banks
   ```
   which should return the name, street address and location of the **Meridian** and **San Carlos**

2 Setup development environment

- Hibernate and this tutorial are based on the Java language and require a Java 8 or a later S
- Due to the large number of dependencies, the Maven build tool is used to download pre-re
  build the recipe source code. Download Maven from the link listed above and install in you
- Change to the **Hibernate** directory of the downloaded Db2 sample data and programs. Bui
  command:
  ```
  mvn install
  ```

3 Run sample program

- From the same directory where the recipe was built, run the sample with a command like:
  ```
  mvn exec:java -Dexec.mainClass=”recipe.HSDB2″ -Dexec.args=”<operation>”
  ```
  where **<operation>** is one of:
  – listnear – lists customers within 2000 meters of the Meridian bank branch.
  – listwindow – lists customers withing a specified geographic window
  – listsome – selects all customers in the table and lists the first 10 customers

- Try changing the queries in **HSDB2.java** to use a different distance or query window. Make
  each time before executing the program.

by David Adler

Join The Discussion

Enter your comments...