Setting up Databases for Grails Projects

Lamine Sissoko

Creating an External Database

To create MySQL databases on your computer, you first need to get the MySQL installer from http://dev.mysql.com/downloads/installer/ and run it. One of the last pages of the install will ask you if you want to set a root/admin user so choose a user name and password you will remember and create the root user. We are now ready to access the MySQL server.

Open the MySQL command line client (on Windows there will be a Start Menu icon under MySQL/MySQL Sever x.x). If it won't open then use the regular command line and type the following to reach the MySQL command line: ***mysql -uroot -p***. That command says that you want to connect to the root user. The following line will ask you to provide the password so type your root user password. Now each line should begin with ***mysql>*** instead of ***C:\Users\YourName>*** . Type the following commands to create a new database and a new user to manipulate it:

```
create database cs2340;
create user sqluser identified by "sqluserpw";
grant all privileges on cs2340.* to sqluser;
use cs2340;
```

We now have a local database, cs2340, that we can use for our web project.

Connecting to an External Database

Whenever you create a new Grails project, it uses the HSQL database by default for all of your data access/persistence needs. Pointing your Grails project to a different database requires little more than making a few changes to the following part of **DataSource.groovy** (located under **grails-app/conf**):

```
environments {
    development {
        dataSource {
            dbCreate = "create-drop" // one of 'create', 'create-drop','update'
            url = "jdbc:hsqldb:mem:devDB"
        }
    }
    test {
        dataSource {
            dbCreate = "update"
            url = "jdbc:hsqldb:mem:testDb"
        }
    }
    production {
        dataSource {
            dbCreate = "update"
            url = "jdbc:hsqldb:file:prodDb;shutdown=true"
        }
    }
}
```

First, you need to decide which environment(s) you want to start using a new database for. I'll focus on the **production** environment. In the section above, I created a regular MySQL database on my computer named cs2340 and granted all privileges on it to a defined user (name=”sqluser”, password=”sqluserpw”) to use for this Grails project. To switch from HSQL to this database, we need to add references to its **url** as well as the **username** and **password** for a user that has privileges on it :

```
production {
    dataSource {
         dbCreate = "update"
         url = "jdbc:mysql://localhost/cs2340”;
         username = "sqluser"
         password = "sqluserpw"
    }
}
```

Now when you run your program in production mode (by typing ***prod run-app*** in Eclipse/STS's Grails command window or ***grails prod run-app*** if you're working from the command line) your program will pull data from and store data in the MySQL database instead of the standard HSQL one.

Data Migration

You may have noticed that each environment in DataSource.groovy has a **dbCreate** attribute under dataSource. It can take one of three values, **'create', 'create-drop','update'**, that determine the state of the database on project launch and shutdown (if you pick no value then you must manage database data migration yourself).

Both **'create'** and **'create-drop'** will create tables matching your Domain classes when you run your program. They will also populate these tables with any data you add to BootStrap.groovy (also located under grails-app/conf). But your data will be lost when you exit the running program.

In **'update'** mode, data is preserved after shutdown. So when you start your program again, new tables matching your Domain classes are added if they do not already exist in the database, and the results of all operations on these tables are not lost after shutdown.

Useful Links

Grails Documentation
http://www.grails.org/doc/latest/guide/

Using STS Tool Suite in Eclipse (instead of standalone IDE)
http://blog.springsource.com/2011/03/25/early-access-springsource-tool-suite-for-eclipse-indigo-3-7/