Chapter 1. Installing the GemFire labs Environment

1.1. Introduction

Welcome to the GemFire-Developer course. We have tried to make the lab setup as simple as possible. There are some installers that you can use to install everything you need for the course on your file system.

What You will do:

- Install and configure STS, GemFire and lab environment
- Start servers and verify correct configuration

Estimated completion time: 30 minutes

1.2. Instructions

Instructions for this lab are divided into specific sections. Each section describes the steps to perform specific tasks.

1.2.1. GemFire Installation Prerequisites

1. Make sure that your system meets the system requirements listed. Refer to the product documentation at http://docs.gopivotal.com/gemfire/.
2. Check that your Java version is appropriate and that Java is on your path. To check your current Java version, type java -version at a command-line. Your output should look something like this

   ```
   java version "1.8.0_66"
   Java(TM) SE Runtime Environment (build 1.8.0_66-b17)
   Java HotSpot(TM) 64-Bit Server VM (build 25.66-b17, mixed mode)
   ```

3. To install GemFire and the lab environment, you will need Java Developer Kit 1.7.0 or higher. If this command returned an error, that means Java is either not installed, or not on your `PATH` .

   You can download Sun/Oracle Java SE here

   Note

   Java 1.8 is officially supported as of GemFire version 8.2

1.2.2. Install Lab Environment

You will be using the Spring Tool Suite (STS) IDE for building and executing most of the exercises for this course. The lab installation package includes not only the STS but also GemFire and lab files. To install, perform the following steps.

1. Locate the appropriate installer for your target platform. Several are provided for Windows (32 and 64 bit), Linux (32 and 64 bit) and Mac. Once located, run the installer and generally take the recommended default values.

   Note

   Make sure you don't have any blank spaces in the path you choose for installation. For example, avoid choosing directories with path such as `C:\Document and Settings\gemfire`.

2. Once installation has completed, locate the installation folder and labs. For Windows, the path should be C:\GemFire-Developer-8.2.a.RELEASE. For Mac, the path will likely be /Applications/GemFire-Developer-8.2.a.RELEASE.

   Once there, note the folders that have been created.

   The lab files will be found in the GemFire-Developer-8.2.a.RELEASE folder (lab folder has the same name as the course installation folder). The GemFire installation will be found at `Pivotal_GemFire_800_b48319_Linux`

   Figure 1.1. Listing of folders created by course installer

   Change directories to GemFire-Developer-8.2.a.RELEASE/server-bootstrap. This folder contains the basic setup and initialization tooling for the server side of these labs.

1.2.3. Starting services

In this section, you will configure the environment based on your installation and start the services. To do so, perform the following steps.

1. Two files have been set up in the scripts folder to facilitate setting up the environment variables. Edit one of the configuration scripts (gf.confg for Mac and Linux) or setEnv.bat for Windows. The main thing you need to ensure is that you've properly set the JAVA_HOME location. In addition, verify that COURSE_HOME is correctly set. If you accepted the default installation settings, they should be correct. Your file will possibly look like one of the following:

   ```
   # JAVA_HOME may already be set
   export JAVA_HOME=/Library/Java/JavaVirtualMachines/jdk1.7.0_55.jdk/Contents/Home
   export COURSE_HOME=/Applications/GemFire-Developer-8.2.a.RELEASE
   export LAB_HOME=$COURSE_HOME/GemFire-Developer-8.2.a.RELEASE
   export GEMFIRE=$COURSE_HOME/Pivotal_GemFire_800_b48319_Linux
   export PATH=$PATH:$GEMFIRE/bin
   export GF_JAVA=$JAVA_HOME/bin/java
   export GF_SAMPLES=$GEMFIRE/SampleCode
   export CLASSPATH=../bin:./bin:./target/classes:$JAVA_HOME/lib/tools.jar:$GEMFIRE
   /lib/gemfire.jar:$GEMFIRE/lib/antlr.jar:$GEMFIRE/lib/gfSecurityImpl.jar:$GF_SAMP
   LES/helloworld/classes:$GF_SAMPLES/quickstart/classes:$CLASSPATH
   ```

   Figure 1.2. gf.config Example

   ```
   set JAVA_HOME=C:\Progra~1\Java\jdk1.6.0_18
   set COURSE_HOME=C:\GemFire-Developer-8.2.a.RELEASE
   set LAB_HOME=%COURSE_HOME%\GemFire-Developer-8.2.a.RELEASE
   set GEMFIRE=%COURSE_HOME%\Pivotal_GemFire_800_b48398_Windows
   set PATH="%PATH%";%GEMFIRE%\bin;%JAVA_HOME%\bin
   set GF_LIB=%GEMFIRE%\lib
   set GF_SAMPLES=%GEMFIRE%\SampleCode

   set CLASSPATH=.;../bin;./bin;%GF_LIB%\gemfire.jar;%GF_LIB%\antlr.jar;%GF_LIB%
   \gfSecurityImpl.jar;%GF_SAMPLES\helloworld\classes;%GF_SAMPLES%\quickstart\classes;
   %CLASSPATH%
   ```

   Figure 1.3. setEnv.bat example

2. Open terminal window or command window and change directories to `<LAB_HOME>/server_bootstrap/scripts` folder. The location of `LAB_HOME` will depend on where you installed the course on your system and should be the same as defined in either the `gf.config` file or `setEnv.bat` file above.

   Run the startServer.sh script or startServer.bat script to start the server processes and load the cache server with data that will be used for the client-side labs. You should see output similar to the following.

   ```
   Got the BookMaster Region: com.gemstone.gemfire.internal.cache.LocalRegion[path='/BookMaster';scope=LOCAL'
   ;dataPolicy=EMPTY; concurrencyChecksEnabled]
   Inserted a book: BookMaster [itemNumber=123, title=A Treatise of Treatises]
   Inserted a book: BookMaster [itemNumber=456, title=Clifford the Big Red Dog]
   Inserted a book: BookMaster [itemNumber=789, title=Operating Systems: An Introduction]
   Got the Customer Region: com.gemstone.gemfire.internal.cache.LocalRegion[path='/Customer';
   scope=LOCAL';dataPolicy=EMPTY; concurrencyChecksEnabled]
   Inserted a customer: Customer [customerNumber=5598, firstName=Kari, lastName=Powell, postalCode=44444]
   Inserted a customer: Customer [customerNumber=5543, firstName=Lula, lastName=Wax, postalCode=12345]
   Inserted a customer: Customer [customerNumber=6024, firstName=Trenton, lastName=Garcia, postalCode=88888]
   ```

   Figure 1.4. Output from running startServer.sh or startServer.bat

   > **Note**
   >
   > You can also stop the server processes by running stopServer.sh on Linux and Mac or stopServer.bat on Windows

1.2.4. Becoming familiar with the STS

In this section, you will start the STS, which is where most of the labs will be performed.

Begin by locating the STS icon, either by browsing to it (refer to section 1.2.2 above for the location) or by locating an icon, which may have been created on your desktop.

When running the STS for the first time, the tooling will perform some initial setup that will result in a number of projects being added and organized into working sets.

Figure 1.5. STS project structure

Expand the `01-server-bootstrap` working set and then the `server-bootstrap` project. Expand the `src/test/java` folder and then the `com.gopivotal.training` package. Locate the `VerifyServerTests.java` class file and execute as a JUnit test.

Tip

To run this program, you can select the file, right-mouse click and select `Run As -> JUnit Test`.

This test will verify that you have correctly configured the server by running a client test that accesses the distributed system, fetching an entry from the server. If you've configured everything correctly, you should see the following output as a result.

Congratulations!! You have completed this lab.