

Chapter 4. gfsh practice

4.1. Introduction

This lab is intended to increase your familiarity with `gfsh`, the gemfire shell. Unlike other labs, this lab has no code component: all tasks will be performed inside the shell.

In this lab, you will learn:

- how to launch and exit the shell
- starting and stopping locators, servers
- creating a region, inserting data, querying regions
- obtain information about cluster members, regions



Note

Full documentation for `gfsh` is available at: http://gemfire.docs.pivotal.io/docs-gemfire/latest/tools_modules/gfsh/chapter_overview.html

Estimated completion time: 20 minutes

4.2. Overview

This session is intended to be a brief tour of working within `gfsh`. Among other things, you will be starting a locator, servers, a region, inserting and querying data, stop and restart a server, and validating the expected behavior of the system. Let's get started.

4.3. Instructions

4.3.1. Launching and exiting `gfsh`

Gemfire bundles the command line tool `gfsh`. You will find it located in `$GEMFIRE/bin`, which should be in your `$PATH`, so that, launching the shell is simply a matter of invoking the command: **`gfsh`**.

To quit `gfsh`, simply type **`exit`**.

4.3.2. Stopping existing servers and locators

Go ahead and launch `gfsh`, and make sure that prior running servers and locators are stopped. To do that, perform the following:

- connect to the running locator with **`connect --locator=localhost[41111]`** (if you get a message saying you could not connect, it's likely you have no locator running, in which case proceed to the next section).
- invoke the command **`shutdown`** to stop the servers,
- stop the locator with the command **`stop locator`** (you'll have to supply the name of the locator you wish to stop (it should simply be 'locator')).

4.3.3. Built-in help

`gfsh` comes with built-in help at multiple levels.

In its most general form, simply invoking **`help`** will list all available commands.

Notice that many of the commands begin with the same verb, such as `start`, `stop`, `status`, and `list`. For example, to learn more about the `start` command, type **`help start`**.

To obtain feedback on a specific command, such as "start server," type **`help start server`**. Notice the output will provide information about all options that can be supplied along with this command. The list of options for starting a server is rather lengthy.

4.3.4. Start a locator

Let's begin by starting a locator. Try to type just **`start locator`** and see what happens. What feedback do you get?



Note

If no port number is specified, the locator will start on its default port number, 10334

4.3.5. Start a server

Start a server, make sure to give it the name "server1."



Tip

`gfsh` sports command completion. Many commands, such as **`start server`** come with many options. As you construct your command, use the `tab` key to obtain a list

of options to append to your command.

In prior labs, we usually referenced a `gemfire.properties-file`. In this case, do not supply this option as we have no need to reference, for example, the location of a `server.cache.xml` file.

Also, rather than manage the specific port our server will listen on, let gemfire select the port number. To do this, make sure to supply the option **server-port** with a value of 0.

4.3.6. Validate running locator and server

In gfsH parlance, the locator and server we just started are known as members. To verify that we have one of each running, invoke the command **list members**. Inspect the output.

To gain more information about these two running members, explore the **describe member** command. Use the built-in help to help you figure out how to invoke the command for each your locator and server1.

Inspect the output. For the server, can you tell whether it's running, what the running port is, and the locator it's associated with?

4.3.7. Regions

Let's create a replicated region named "people." Again use the help command to assist you in figuring out the syntax. You'll need to use the **type** option to specify that the type of region you want to create is a replicated region (the value you supply here is case sensitive).

Just like we can list members, we can also list regions. Go ahead and use the **list regions** command to verify that the region named "people" was properly created.

We can also describe regions. Run the **describe region** command on the people region and inspect the output. Verify the region type.

Our region is empty for now. Let's verify this by querying it. Use the **query** command for this. If you haven't yet gone over the lab about OQL and the gemfire query language, you can use the following simple query: **select * from /people**.

4.3.8. Inserting data

Insert a simple record into the people region. Supply a key and a value and make sure you specify what region you want to "put" (hint) this record into.

Run your query one more time to verify that the record was inserted.

Insert a second record into the people region.

4.3.9. Start a second server

Start a second server named "server2." Once more, let gemfire select what port the server should listen on.

Run **list members** once more to verify that we now have two running servers.

Run the query once more against the people region to verify that it still functions properly.

Describe the people region once more. Notice that both server1 and server2 are listed as "hosting members." This is a validation that the data is being replicated across both servers.

4.3.10. Stop server1

Use the **stop server** command to stop server1. Inspect the output of **list members** to validate that server1 is no longer running. Run the query once more. Notice that it is still working because we had redundancy/replication.

Start server1 back up.

Explore the **status server** command on either of the running servers. What additional information can we glean about the server compared to the **describe member** command?

4.3.11. Exit, then return to the shell

At this point, issue the **exit** command to exit the shell. Now re-invoke gfsH and re-connect to your locator (port number should be 10334). Notice that your servers should still be running. So, exiting merely detaches you from the shell but has no impact on the running servers.

4.3.12. Shutting down

Issue the command **shutdown** to stop your servers. Using the built-in help, issue the **stop locator** command to stop your locator.

In this session, we've covered a lot of ground. Although we haven't explored the myriad of options that many of the commands sport, you should now be familiar and comfortable enough with gfsH to explore further. Congratulations!! You have completed this lab.
