Chapter 11. Data Serialization

11.1. Introduction

In this lab, you will gain hands-on experience with...

Concepts you will gain experience with:

- Configuring your cache to perform PDX Auto Serialization
- Working with multiple version of domain objects in PDX
- Using PdxInstance get a single field

Estimated completion time: 40 minutes

11.2. Quick Instructions

Quick instructions for this exercise have been embedded within the lab materials in the form of TODO comments. To display them, open the `Tasks` view (Window -> Show view -> Tasks (*not Task List*)).

11.3. Detailed Instructions

Instructions for this lab are divided into specific sections. Each section describes the steps to perform specific tasks. Before beginning any of these tasks.

11.3.1. Configuring the Cache for PDX Auto Serialization

In this section, you will focus on configuring PDX Auto Serialization in both the client and server configurations.

1. (`TODO-01`) To begin, open the `Customer` class definition in the `com.gopivotal.bookshop.domain` package. Take a moment to examine how it's written. Note first of all that it does not implement `java.io.Serializable` at all. Also note that one of the field is an `Address` instance that represent various attributes related to a customer's address (ex. city, state, zipcode, etc). If you like, you can also open this `Address` class (in the same package) and note that it also does not implement `java.io.Serializable`. In this condition, the only way to ensure these objects are serialized is to use one of the PDX Serialization techniques. Note also that both the `Customer` and `Address` classes have default constructors, which are a requirement for PDX Serialization.
2. (`TODO-02`) Next, open the `xml/serverCache.xml` file. Your task is to add the appropriate configuration to enable PDX Serialization in the server cache. This includes configuring the `ReflectionBasedAutoSerializer` class as the serializer class.
3. (`TODO-03`) In addition, configure a parameter to the auto serializer called `classes` that registers the classes that should be auto serialized. You can use wild cards to be sure you get both the `Customer` class and `Address` class in the `com.gopivotal.bookshop.domain` package.
4. (`TODO-04`) Finally, add an attribute to the pdx element signaling that the server should NOT de-serialize objects.
5. Use gfsh to start the locator and two servers.

   Note

   Unlike prior exercises, you will NOT be using the `--classpath` argument when starting servers. It is important in observing the benefit of PDX Serialization that the domain classes not be on the classpath of the servers.

6. (TODO-05) Open the `xml/clientCache.xml` file and make a similar modification to the definition to support PDX Auto Serialization. Do NOT set the attribute to force client to read as a serialized object. We actually do want the PDX de-serialization to take place on the client.
7. (TODO-06) Locate the `CustomerLoader` class in the `com.gopivotal.bookshop.buslogic` package and run it. This will load 3 customers into the Customer region on the server.
8. Return to gfsh and execute a query to select values from the Customer region.

   You should see something like the following as output.

```
customerNumber | firstName | lastName |                    primaryAddress                    |
myBookOrders
-------------- | --------- | -------- | ---------------------------------------------------- |
-------------------------
5598           | Kari      | Powell   | class com.gemstone.gemfire.pdx.internal.PdxInstanceImpl |
class java.util.ArrayList
6024           | Trenton   | Garcia   | class com.gemstone.gemfire.pdx.internal.PdxInstanceImpl |
class java.util.ArrayList
5543           | Lula      | Wax      | class com.gemstone.gemfire.pdx.internal.PdxInstanceImpl |
class java.util.ArrayList
```

11.3.2. Working with PDX Domain Object Versions

In this section, you'll begin to understand the power of PDX as you modify the definition of the `Customer` class and see that multiple versions of the class definition can be used within the GemFire distributed system at the same time.

1. (`TODO-07`) To begin, open the `Customer` class again and add a `telephoneNumber` field of type `String`. Also add a getter and setter method. You might also want to go to the `toString()` method and add code to ensure the values of this new field get printed.
2. (`TODO-08`) Open the `NewCustomerClient` class in the `com.gopivotal.bookshop.buslogic` package and locate the `testCustomer()` method. Write the code to create a new `Customer` instance. Be sure to set the new `telephoneNumber` property. Save the entry with the key `9999`.
3. (TODO-09) Locate the `testGet()` method and add code to get the newly inserted `Customer` entry (key: `9999`) and print it out.
4. Run `NewCustomerClient` to insert this new record into GemFire.
5. Return to gfsh and re-issue the query command. This time, note that there is the additional entry for key `9999`. Notice also that there is a new field displayed for `telephoneNumber`. Note that the first three entries now show this field value as `null`, which is the expected behavior.

11.3.3. Using PdxInstance

In this last section, you will gain familiarity working with the PdxInstance object. This object will offer the ability to only de-serialize the fields that are required to perform necessary processing.

1. (`TODO-10`) Open the `PdxInstanceClient` class in the `com.gopivotal.bookshop.buslogic` package and locate the `testPdxGet()` method. Add the necessary code to fetch the entry for key `9999`, processing as a `PdxInstance`. Extract and print just the `telephoneNumber` field.

   

   Tip

   It may be a good idea to add a test to ensure the instance you got back is an instance of PdxInstance and print out a notice if it isn't.

2. (`TODO-11`) Return to the `clientCache.xml` file and add the appropriate attribute to the pdx element to tell the client cache NOT to de-serialize entries into `Customer` objects. This is important if you intend to process your entries as `PdxInstance` objects.
3. Run the `PdxInstanceClient` and verify that you were able to obtain the `telephoneNumber` field and that it is the correct value. It should be the value you set in the prior section when you created the new `Customer` entry.

Congratulations!! You have completed this lab.
