Quick Start

Table of contents

1 Introduction

This is a quick start guide to help you setup the project.

2 Download

First you need to check out the most stable Gora release through the official Apache Gora release page here.

For those who would like to use a development version Gora or simply wish to work with the bleeding edge, instructions for how to check out the source code using svn or git can be found here.

3 Compiling the project (Maven users)

Once you have the source code for Gora, you can compile the project using

```
$ cd gora
$ mvn clean compile
```

You can also compile individual modules by cd'ing to the module directory and running `$ mvn clean compile` there.

4 Setting up your project

More recently Gora began using Maven to manage it's dependencies and build lifecycle. Stable Gora releases are available on the central maven repository or ivy repositories and Gora-SNAPSHOT OSGi bundle artifacts are now pushed to the Apache Nexus here.

You can manage Gora dependencies in a few ways.

4.1 Using ivy to manage gora

If your project already uses ivy, then you can include gora dependencies to your ivy by adding the following lines to your `ivy.xml` file:

```
<dependency org="org.apache.gora" name="gora-hbase"
rev="${version}" conf="*->compile" changing="true">
<dependency org="org.apache.gora" name="gora-cassandra"
rev="${version}" conf="*->compile" changing="true">
<dependency org="org.apache.gora" name="gora-sql"
rev="${version}" conf="*->compile" changing="true">
```

N.B. The `${version}` variable should be replaced by the most stable Gora release.

Only add the modules `gora-hbase, gora-cassandra, gora-sql` that you will use, and set the `conf` to point to the configurations (of your project) that you want to depend on gora. The `changing="true"` attribute states that, gora artifacts should not be cached, which is required if you want to change gora's source and use the recompiled version.

Add the following to your `ivysettings.xml`

```
   <resolvers>
     ...
     <chain name="internal">
       <resolver ref="local"/>
     </chain>
     ...
   </resolvers>
   <modules>
     ...
     <module organisation="org.apache.gora" name=".*" resolver="internal"/>
     ...
   </modules>
```

This forces gora to be built locally rather than look for it in other repositories.

4.2 Using Maven to manage Gora

If your project however uses maven, then you can include gora dependencies to your project by adding the following lines to your `pom.xml` file:

```
<dependency>
   <groupId>org.apache.gora</groupId>
   <artifactId>gora-hbase</artifactId>
   <version>${version}</version>
</dependency>



<dependency>
   <groupId>org.apache.gora</groupId>
   <artifactId>gora-cassandra</artifactId>
   <version>${version}</version>
</dependency>



<dependency>
   <groupId>org.apache.gora</groupId>
   <artifactId>gora-sql</artifactId>
   <version>${version}</version>
</dependency>
```

N.B. The `${version}` variable should be replaced by the most stable Gora release.

Only add the modules `gora-hbase, gora-cassandra, gora-sql` that you will use.

4.3 Managing gora jars manually

You can include gora jars manually, if you prefer so. After compiling gora first copy all the jars in `gora-[modulename]/lib/` dir. Then copy all the jars in `gora-core/lib/` since all of the modules depend on `gora-core`. Last, copy the actual gora-jars in `gora-core/build/gora-core-x.x.jar` and the jars of all the other modules that you want to use (for example `gora-hbase/build/gora-hbase-x.x.jar`)

5 What's next

After setting up gora, you might want to check out the documentation. Most of the documentation can be find at the project web site or at the wiki.

5.1 Gora Modules

Gora source code is organized in a modular architecture. The gora-core module is the main module which contains the core of the code. All other modules depend on the gora-core module. Each data store backend in Gora resides in it's own module. The documentation for the specific module can be found at the module's documentation directory.

It is wise so start with going over the documentation for the gora-core module and then the specific data store module(s) you want to use. Below are the modules in gora.

- gora-core: Module containing core functionality
- gora-cassandra: Module for Apache Cassandra backend
- gora-hbase: Module for Apache HBase backend
- gora-sql: Module for SQL database backends