

MavenPlugin

Mojo's Cassandra Maven Plugin

There is a Maven plugin for Cassandra. It is hosted at the mojo project: [Mojo's Cassandra Maven Plugin](#)

The plugin has the following goals.

- `cassandra:start` Starts up a test instance of Cassandra in the background.
- `cassandra:stop` Stops the test instance of Cassandra that was started using `cassandra:start`.
- `cassandra:run` Starts up a test instance of Cassandra in the foreground.
- `cassandra:load` Runs a `cassandra-cli` script against the test instance of Cassandra.
- `cassandra:repair` Runs `nodetool repair` against the test instance of Cassandra.
- `cassandra:flush` Runs `nodetool flush` against the test instance of Cassandra.
- `cassandra:compact` Runs `nodetool compact` against the test instance of Cassandra.
- `cassandra:cleanup` Runs `nodetool cleanup` against the test instance of Cassandra.
- `cassandra:delete` Deletes the the test instance of Cassandra.

Step-by-step develop a Cassandra backed web application in 2 minutes

Here is a from zero to web application step-by-step guide. We assume that you have Apache Maven 3.0.4 installed and configured with proxies correctly.

1. Verify that Maven is installed and that you are using Java 6.

```
$ mvn -version
Apache Maven 3.0.4 (r1232337; 2012-01-17 08:44:56+0000)
Java version: 1.6.0_20, vendor: Sun Microsystems Inc.
Java home: /usr/java/jdk1.6.0_20/jre
Default locale: en_US, platform encoding: UTF-8
OS name: "linux", version: "2.6.18-194.26.1.el5", arch: "amd64", family: "unix"
```

You are looking to verify the Apache Maven 3.0.4 and that the Java version starts with 1.6.

2. Create a basic web application

```
$ mvn archetype:generate
[INFO] Scanning for projects...
[INFO]
[INFO] -----
[INFO] Building Maven Stub Project (No POM) 1
[INFO] -----
...
Choose archetype:
...
102: remote -> maven-archetype-webapp (An archetype which contains a sample Maven Webapp project.)
...
Choose a number: 99:
```

Note that the numbers you will see are different every time. You are looking for the one that is `maven-archetype-webapp`

Type in the number of `maven-archetype-webapp` and press ENTER.

```
Choose a number: 99: 102
Choose version:
1: 1.0-alpha-1
2: 1.0-alpha-2
3: 1.0-alpha-3
4: 1.0-alpha-4
5: 1.0
Choose a number: 5:
```

For archetypes, you usually pick the latest version, if you want to follow exactly, pick version 1.0

```
Define value for property 'groupId': :
```

We'll use `org.apache.wiki.cassandra.mavenplugin` as the `groupId`. You should use a domain name that you control reversed as is the fashion for package names.

```
Define value for property 'groupId': : org.apache.wiki.cassandra.mavenplugin
Define value for property 'artifactId': : webapp
Define value for property 'version': : 1.0-SNAPSHOT
Define value for property 'package': : org.apache.wiki.cassandra.mavenplugin: :
Confirm properties configuration:
groupId: org.apache.wiki.cassandra.mavenplugin
artifactId: webapp
version: 1.0-SNAPSHOT
package: org.apache.wiki.cassandra.mavenplugin
Y: :
```

Press `Y` followed by `ENTER` to create the project.

```
[INFO] -----
[INFO] Using following parameters for creating project from Old (1.x) Archetype: maven-archetype-webapp:1.0
[INFO] -----
[INFO] Parameter: groupId, Value: org.apache.wiki.cassandra.mavenplugin
[INFO] Parameter: packageName, Value: org.apache.wiki.cassandra.mavenplugin
[INFO] Parameter: package, Value: org.apache.wiki.cassandra.mavenplugin
[INFO] Parameter: artifactId, Value: webapp
[INFO] Parameter: basedir, Value: /home/stephenc/src/cassandra-wiki
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] ***** End of debug info from resources from generated POM *****
[INFO] project created from Old (1.x) Archetype in dir: /home/stephenc/src/cassandra-wiki
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 4:57.118s
[INFO] Finished at: Wed Feb 09 11:04:37 GMT 2011
[INFO] Final Memory: 10M/106M
[INFO] -----
```

1. At this point we should have a basic web application created in the `webapp` (or whatever the `artifactId` you chose was) directory. The directory structure should look a little bit like

```
webapp
+-- pom.xml
|-- src
    |-- main
        +-- resources
        |-- webapp
            +-- WEB-INF
            |   |-- web.xml
            |-- index.jsp
```

2. We now need to add the client library that we will use to the list of dependencies. Open the `pom.xml` in your favourite editor. In the `<dependencies>` section add the hector dependency. (You could use any client library, we will use hector because it is available in the Maven Central repository which makes using it a lot easier) Your `pom.xml` should look a little something like:

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>org.apache.wiki.cassandra.mavenplugin</groupId>
  <artifactId>webapp</artifactId>
  <packaging>war</packaging>
  <version>1.0-SNAPSHOT</version>
  <name>webapp Maven Webapp</name>
  <url>http://maven.apache.org</url>
  <dependencies>
    <dependency>
      <groupId>me.prettyprint</groupId>
      <artifactId>hector-core</artifactId>
      <version>1.0-5</version>
    </dependency>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
  <build>
    <finalName>webapp</finalName>
  </build>
</project>
```

While we are at it, add a reference to the jetty-maven-plugin and change the java source level to 1.6

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>org.apache.wiki.cassandra.mavenplugin</groupId>
  <artifactId>webapp</artifactId>
  <packaging>war</packaging>
  <version>1.0-SNAPSHOT</version>
  <name>webapp Maven Webapp</name>
  <url>http://maven.apache.org</url>
  <dependencies>
    <dependency>
      <groupId>me.prettyprint</groupId>
      <artifactId>hector-core</artifactId>
      <version>1.0-5</version>
    </dependency>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
  <build>
    <finalName>webapp</finalName>
    <plugins>
      <plugin>
        <artifactId>maven-compiler-plugin</artifactId>
        <version>2.3.2</version>
        <configuration>
          <source>1.6</source>
          <target>1.6</target>
        </configuration>
      </plugin>
      <plugin>
        <groupId>org.mortbay.jetty</groupId>
        <artifactId>maven-jetty-plugin</artifactId>
        <version>6.1.24</version>
      </plugin>
    </plugins>
  </build>
</project>
```

3. Now create a script to set-up our cassandra instance. First create the following directory structure:

```
webapp
+-- pom.xml
|-- src
    +-- cassandra
        |   |-- cli
    |-- main
        +-- resources
            |-- webapp
                +-- WEB-INF
                    |   |-- web.xml
                    |-- index.jsp
```

and then using your favourite editor, create a file called `load.script` in the `webapp/src/cassandra/cli` directory.

```
create keyspace WebappKeyspace
  with placement_strategy = 'org.apache.cassandra.locator.SimpleStrategy'
  and strategy_options = {replication_factor:1};
use WebappKeyspace;
create column family Example with column_type='Standard' and comparator='UTF8Type';
```

4. Now start development

```
$ mvn cassandra:start jetty:run -Dcassandra.jmxPort=7199
[INFO] Scanning for projects...
[INFO]
[INFO] -----
[INFO] Building webapp Maven Webapp 1.0-SNAPSHOT
[INFO] -----
[INFO]
[INFO] --- cassandra-maven-plugin:1.0.0-1:start (default-cli) @ webapp ---
[INFO] Waiting for Cassandra to start...
...
[INFO] [WARN] 11:32:02,516 Generated random token 11852913165738683554068538477956203649. Random tokens
will result in an unbalanced ring; see http://wiki.apache.org/cassandra/Operations
[INFO] Cassandra cluster "Test Cluster" started.
[INFO] Running /home/stephenc/src/cassandra-wiki/webapp/src/cassandra/cli/load.script...
...
[INFO] Connected to: "Test Cluster" on 127.0.0.1/9160
[INFO] 3757b11f-3440-11e0-8318-e700f669bcfc
[INFO] Authenticated to keyspace: WebappKeyspace
[INFO] 37c0fa40-3440-11e0-8318-e700f669bcfc
[INFO] Finished /home/stephenc/src/cassandra-wiki/webapp/src/cassandra/cli/load.script.
[INFO] Cassandra started in 2.5s
[INFO]
[INFO] >>> maven-jetty-plugin:6.1.24:run (default-cli) @ webapp >>>
...
[INFO] Tmp directory = determined at runtime
[INFO] Web defaults = org/mortbay/jetty/webapp/webdefault.xml
[INFO] Web overrides = none
[INFO] web.xml file = /home/stephenc/src/cassandra-wiki/webapp/src/main/webapp/WEB-INF/web.xml
[INFO] Webapp directory = /home/stephenc/src/cassandra-wiki/webapp/src/main/webapp
[INFO] Starting jetty 6.1.24 ...
2011-02-09 11:32:04.604:INFO::jetty-6.1.24
2011-02-09 11:32:04.726:INFO::No Transaction manager found - if your webapp requires one, please
configure one.
2011-02-09 11:32:04.916:INFO::Started SelectChannelConnector@0.0.0.0:8080
[INFO] Started Jetty Server
```

At this point you have a running web application at <http://localhost:8080/webapp/> and a cassandra instance started on localhost. Jetty will automatically re-load the webapp if you modify it, so you can now just start developing the web application (assuming you are a masochist who likes writing web applications in *jsp* - gasp horror!)

<https://c.statcounter.com/9397521/0/fe557aad/1/> | stats