**Creating an HBase table using Java API on Cloudera Quickstart VM**

1. Start the Cloudera Quickstart VM on Virtualbox. Open terminal and proceed as follows:
2. Create a new directory using the mkdir command and make it your working directory. We have named the directory HBase.

[cloudera@quickstart ~]$ **mkdir HBase**

[cloudera@quickstart ~]$ **cd HBase**

3. Copy “MyFirstHBaseTable.java” program from the provided s3 link to your virtual machine using command wget inside the HBase directory.

[cloudera@quickstart HBase]$ **wget https://s3.amazonaws.com/hbasejavaapi/MyFirstHBaseTable.java**

4. Verify with ls command.

[cloudera@quickstart HBase]$ **ls**

5. Compile and execute the above program as shown below-

* Set the environment Variables.

[cloudera@quickstart HBase]$ **export HADOOP\_CLASSPATH=$HADOOP\_PATH:`hbase classpath`**

* Now compile:-

[cloudera@quickstart HBase]$ **javac -cp `hbase classpath` MyFirstHBaseTable.java**

* Verify if the class file is created or not by entering the **ls** command.
* Now execute:-

[cloudera@quickstart HBase]$ **java -cp `hbase classpath` MyFirstHBaseTable**

* The following should be the output:

*Table created*

**Checking with HBase Shell**

We can check our hbase table using following HBase shell commands:

1. [cloudera@quickstart HBase]$ **hbase shell**
2. Run list to see the table.

hbase(main):001:0> **list**

1. You may exit the shell by entering exit.

hbase(main):002:0> **exit**

**Inserting Data in the table Using Java API**

Now we created TableDemo4 table in Hbase with columns ‘Id’ and ‘Name’. You can insert data into Hbase using the add() method of the Put class. You can save it using the put() method of the HTable class. The following code inserts data into the TableDemo4 table:

1. Download “PopulateHBaseTable.java “ from the s3 link using the wget command and place it inside the Hbase directory.

[cloudera@quickstart HBase]$ **wget** [**https://s3.amazonaws.com/hbasejavaapi/PopulateHBaseTable.java**](https://s3.amazonaws.com/hbasejavaapi/PopulateHBaseTable.java)

1. [cloudera@quickstart HBase]$ **javac -cp `hbase classpath` PopulateHBaseTable.java**
2. [cloudera@quickstart HBase]$ **ls**

*MyFirstHBaseTable.class MyFirstHBaseTable.java PopulateHBaseTable.class PopulateHBaseTable.java*

1. [cloudera@quickstart HBase]$ **java -cp `hbase classpath` PopulateHBaseTable**
2. Output:

*GET: Id: AAAName: BBB*

*Found row : keyvalues={row1/Id:col1/1520256266586/Put/vlen=3/seqid=0, row1/Name:col2/1520256266586/Put/vlen=3/seqid=0}*

1. Verify-login hbase shell

* [cloudera@quickstart HBase]$ **hbase shell**

*18/02/27 22:08:44 INFO Configuration.deprecation: hadoop.native.lib is deprecated. Instead, use io.native.lib.available*

*HBase Shell; enter 'help<RETURN>' for list of supported commands.*

*Type "exit<RETURN>" to leave the HBase Shell*

*Version 1.2.0-cdh5.14.0, rUnknown, Sat Jan 6 13:47:53 PST 2018*

* hbase(main):001:0> **list**
* hbase(main):002:0> **scan "TableDemo4"**

*ROW COLUMN+CELL*

*row1 column=Id:col1, timestamp=1519769114329, value=AAA*

*row1 column=Name:col2, timestamp=1519769114329, value=BBB*

*1 row(s) in 0.1770 seconds*