

Java-Success.com

Prepare to fast-track, choose & go places with 800+ Java & Big Data Q&As with lots of code & diagrams.

[Home](#) [Why? ▾](#) [300+ Java FAQs ▾](#) [300+ Big Data FAQs ▾](#) [Courses ▾](#)

[👤 Membership ▾](#) [Your Career ▾](#)

[Home](#) > [bigdata-success.com](#) > [Tutorials - Big Data](#) > [TUT - Cloudera on Docker](#) > 25:

Docker Tutorial: HBase (i.e. NoSQL) Java API on Cloudera quickstart

25: Docker Tutorial: HBase (i.e. NoSQL) Java API on Cloudera quickstart

 Posted on [June 19, 2019](#)

This extends [Docker Tutorial: BigData on Cloudera quickstart via Docker](#).

Step 1: Run the container on a command line.

```
1 ~/projects/docker-hadoop]$ docker run --hostname=q
2 --privileged=true -t -i -v /Users/arulkumarankumar
3 --publish-all=true -p 8888:8888 -p 80:80 -p 7180:7
4 cloudera/quickstart /usr/bin/docker-quickstart
```

300+ Java Interview FAQs

300+ Java FAQs



16+ Java Key Areas Q&As



150+ Java Architect FAQs



80+ Java Code Quality Q&As



150+ Java Coding Q&As



300+ Big Data Interview FAQs

300+ Big Data FAQs



[Tutorials - Big Data](#)



[TUT - !\[\]\(8891837fe1b5b26680f2ee7b0ea5318e_img.jpg\) Starting Big Data](#)

[TUT - Starting Spark & Scala](#)

Install maven

Step 2: Download maven from

"http://mirror.ventraip.net.au/apache/maven/maven-3/3.6.1/binaries/apache-maven-3.6.1-bin.tar.gz"

```
1 [root@quickstart /]# sudo curl -L -O http://mirror
2
3
4
5
```

```
1 [root@quickstart /]# cp apache-maven-3.6.1-bin.tar
2 [root@quickstart /]# cd /opt
3 [root@quickstart /]# sudo tar xvfz apache-maven-3.6
4 [root@quickstart /]# sudo rm -f apache-maven-3.6.1
5
```

Step 3: Add it to your profile so that you can run the "mvn" command. Add it to ".bashrc" if you are a root user or to ".bash_profile" if you are a any other user.

```
1 [root@quickstart opt]# vi ~/.bash_profile
```

add the following lines at the end of "~/.bash_profile"

```
1 M3_HOME=/opt/apache-maven-3.6.1
2 export MAVEN_OPTS=-Dhttps.protocols=TLSv1,TLSv1.1,
3 export PATH=$PATH:$M3_HOME/bin
4
```

Apply the changes with the "source" command.

```
1 [root@quickstart opt]# source ~/.bash_profile
```

```
1 [root@quickstart opt]# mvn -version
2 Apache Maven 3.6.1 (d66c9c0b3152b2e69ee9bac180bb8f
3 Maven home: /opt/apache-maven-3.6.1
4 Java version: 1.7.0_67, vendor: Oracle Corporation
5 Default locale: en_US, platform encoding: UTF-8
6 OS name: "linux", version: "4.9.125-linuxkit", arch
```

TUT - Starting with Python

TUT - Kafka

TUT - Pig

TUT - Apache Storm

TUT - Spark Scala on Zeppelin

TUT - Cloudera

TUT - Cloudera on Docker

TUT - File Formats

TUT - Spark on Docker

TUT - Flume

TUT - Hadoop (HDFS)

TUT - HBase (NoSQL)

TUT - Hive (SQL)

TUT - Hadoop & Spark

TUT - MapReduce

TUT - Spark and Scala

TUT - Spark & Java

TUT - PySpark on Databricks

TUT - Zookeeper

800+ Java Interview Q&As

300+ Core Java Q&As



300+ Enterprise Java Q&As



150+ Java Frameworks Q&As



120+ Companion Tech Q&As



Tutorials - Enterprise Java



```
7 [root@quickstart opt]#  
8
```

Now Maven is installed.

Create a simple maven project structure

Step 4: Create “**projects**” folder and create a maven project structure.

```
1 [root@quickstart ~]# mkdir /projects  
2 [root@quickstart ~]# cd !$  
3 cd projects  
4 [root@quickstart projects]#  
5
```

Create a maven project structure.

```
1 [root@quickstart projects]# mvn archetype:generate  
2 -DartifactId=simple-hbase \  
3 -DarchetypeArtifactId=maven-archetype-quickstart \  
4 -DinteractiveMode=false  
5
```

Give it a few minutes.

```
1 [root@quickstart projects]# tree  
2 .  
3 └─ simple-hbase  
4   │   pom.xml  
5   └─ src  
6       │   main  
7       │   │   java  
8       │   │   │   com  
9       │   │   │   │   mycompany  
10       │   │   │   │   │   app  
11       │   │   │   │   │   App.java  
12       │   └─ test  
13       │       │   java  
14       │       │   │   com  
15       │       │   │   │   mycompany  
16       │       │   │   │   │   app  
17       │       │   │   │   │   AppTest.java
```

```
18  
19 12 directories, 3 files  
20 [root@quickstart projects]#  
21
```

pom.xml add hbase-client

```
1 [root@quickstart projects]# vi simple-hbase/pom.xml  
2
```

```
1 <project xmlns="http://maven.apache.org/POM/4.0.0"  
2   xsi:schemaLocation="http://maven.apache.org/POM/  
3   <modelVersion>4.0.0</modelVersion>  
4   <groupId>com.mycompany.app</groupId>  
5   <artifactId>simple-hbase</artifactId>  
6   <packaging>jar</packaging>  
7   <version>1.0-SNAPSHOT</version>  
8   <name>simple-hbase</name>  
9   <url>http://maven.apache.org</url>  
10  <dependencies>  
11    <dependency>  
12      <groupId>junit</groupId>  
13      <artifactId>junit</artifactId>  
14      <version>3.8.1</version>  
15      <scope>test</scope>  
16    </dependency>  
17  
18    <dependency>  
19      <groupId>org.apache.hbase</groupId>  
20      <artifactId>hbase-client</artifactId>  
21      <version>1.1.3</version>  
22    </dependency>  
23  </dependencies>  
24  
25  <!-- Build uber jar -->  
26  <build>  
27    <plugins>  
28      <plugin>  
29        <groupId>org.apache.maven.plugins</gro  
30        <artifactId>maven-shade-plugin</artifa  
31        <executions>  
32          <execution>  
33            <phase>package</phase>  
34            <goals>  
35              <goal>shade</goal>  
36            </goals>  
37          </execution>  
38        </executions>  
39      </plugin>  
40    </plugins>  
41  </build>  
42
```

```
43 </project>
44
```

Create a HBase table via Java API

```
1 [root@quickstart projects]# vi simple-hbase/src/main
2
```

```
1 package com.mycompany.app;
2
3 import java.io.IOException;
4 import org.apache.hadoop.hbase.HTableDescriptor;
5 import org.apache.hadoop.hbase.client.HBaseAdmin;
6 import org.apache.hadoop.hbase.HColumnDescriptor;
7 import org.apache.hadoop.hbase.TableName;
8 import org.apache.hadoop.conf.Configuration;
9 import org.apache.hadoop.hbase.HBaseConfiguration;
10 import org.apache.hadoop.hbase.client.Table;
11 import org.apache.hadoop.hbase.client.Put;
12 import org.apache.hadoop.hbase.util.Bytes;
13 import org.apache.hadoop.hbase.client.Connection;
14 import org.apache.hadoop.hbase.client.ConnectionFactory;
15
16 public class CreateAnHbaseTable
17 {
18     public static void main(String[] args) throws IOException
19     {
20
21         String TABLE_NAME = "employees";
22
23         Configuration config = HBaseConfiguration.create();
24         config.setInt("timeout", 120000);
25
26         //zookeeper quorum, basic info needed to process requests
27         config.set("hbase.zookeeper.quorum", "quickstart1:2181,quickstart2:2181,quickstart3:2181");
28         config.set("hbase.zookeeper.property.clientPort", "2181");
29
30         HTableDescriptor htable = new HTableDescriptor(TableName.valueOf(TABLE_NAME));
31         htable.addFamily(new HColumnDescriptor("personal"));
32         htable.addFamily(new HColumnDescriptor("professional"));
33         System.out.println("Connecting...");
34         HBaseAdmin hbaseAdmin = new HBaseAdmin(config);
35         System.out.println("Creating Table...");
36         hbaseAdmin.createTable(htable);
37         System.out.println("Done!");
38
39
40         Connection connection = ConnectionFactory.createConnection(config);
41         Table table = connection.getTable(TableName.valueOf(TABLE_NAME));
42
```

```
43     try {
44         Put put = new Put(Bytes.toBytes("999"));
45         put.add(Bytes.toBytes("personal"), Bytes.toBytes("999"));
46         table.put(put);
47         System.out.println("done");
48     } catch (Exception e) {
49         e.printStackTrace();
50     } finally {
51         table.close();
52         connection.close();
53     }
54 }
55 }
56 }
57 }
58 }
59 }
60 }
```

Note: Some API could be deprecated, and replaced with newer APIs.

Build package with maven

```
1 [root@quickstart projects]# cd simple-hbase/
2 [root@quickstart simple-hbase]# mvn package
3 .....
4
```

```
1 [root@quickstart projects]# tree
2 .
3 ├── dependency-reduced-pom.xml
4 ├── pom.xml
5 ├── src
6 │   ├── main
7 │   │   ├── java
8 │   │   │   ├── com
9 │   │   │   │   ├── mycompany
10 │   │   │   │   │   ├── app
11 │   │   │   │   │   │   ├── App.java
12 │   │   │   │   │   │   └── CreateAnHbaseTable.java
13 │   └── test
14 │   │   ├── java
15 │   │   │   ├── com
16 │   │   │   │   ├── mycompany
17 │   │   │   │   │   ├── app
18 │   │   │   │   │   │   └── AppTest.java
19 └── target
```

```
20 | classes
21 |   | com
22 |   |   | mycompany
23 |   |   |   | app
24 |   |   |   |   | App.class
25 |   |   |   |   | CreateAnHbaseTable.class
26 |   | maven-archiver
27 |   |   | pom.properties
28 |   | maven-status
29 |   |   | maven-compiler-plugin
30 |   |   |   | compile
31 |   |   |   |   | default-compile
32 |   |   |   |   |   | createdFiles.lst
33 |   |   |   |   |   | inputFiles.lst
34 |   |   |   | testCompile
35 |   |   |   |   | default-testCompile
36 |   |   |   |   |   | createdFiles.lst
37 |   |   |   |   |   | inputFiles.lst
38 |   | original-simple-hbase-1.0-SNAPSHOT.jar
39 |   | simple-hbase-1.0-SNAPSHOT.jar
40 |   | surefire-reports
41 |   |   | com.mycompany.app.AppTest.txt
42 |   |   | TEST-com.mycompany.app.AppTest.xml
43 |   | test-classes
44 |   |   | com
45 |   |   |   | mycompany
46 |   |   |   |   | app
47 |   |   |   |   |   | AppTest.class
48 |
```

Run

```
1 [root@quickstart simple-hbase]# java -cp target/simple-hbase-1.0-SNAPSHOT.jar com.mycompany.app.AppTest
2 Connecting...
3 log4j:WARN No appenders could be found for logger com.mycompany.app.AppTest.
4 log4j:WARN Please initialize the log4j system properly.
5 log4j:WARN See http://logging.apache.org/log4j/1.2.12faq.html#noconfig for more details.
6 Creating Table...
7 Done!
8 done
9
```

Verify via hbase shell

```
1 [root@quickstart simple-hbase]# hbase shell
2 2019-06-19 12:57:32,277 INFO [main] Configuration: hbase.rootdir=hdfs://quickstart:8020/
3 HBase Shell; enter 'help<RETURN>' for list of supported commands.
4 Type "exit<RETURN>" to leave the HBase Shell
5 Version 1.2.0-cdh5.7.0, rUnknown, Wed Mar 23 11:39:00 UTC 2016
```

```
6  
7 hbase(main):001:0>  
8
```

list

```
1 hbase(main):002:0> list  
2 TABLE  
3 employees  
4 1 row(s) in 0.1800 seconds  
5  
6 => ["employees"]  
7 hbase(main):003:0>  
8
```

scan

```
1 hbase(main):003:0> scan 'employees'  
2 ROW COLUMN+CELL  
3 999 column=personal:  
4 1 row(s) in 0.1230 seconds  
5  
6 hbase(main):004:0>  
7
```

◀ 24: Docker Tutorial: HBase (i.e. NoSQL DB) Shell on Cloudera quickstart

26: Docker Tutorial: Apache Kafka install, create topic & publish message
on Cloudera quickstart ▶

Disclaimer

The contents in this Java-Success are copyrighted and from EmpoweringTech pty ltd. The EmpoweringTech pty ltd has the right to correct or enhance the current content without any prior notice. These are general advice only, and one needs to take his/her own circumstances into consideration. The EmpoweringTech pty ltd will not be held liable for any damages caused or alleged to be caused either directly or indirectly by these materials and resources. Any trademarked names or labels used in this blog remain the property of their respective trademark owners. Links to external sites do not imply endorsement of the linked-to sites. [Privacy Policy](#)

