HBASE - READ DATA

Reading Data using HBase Shell

The **get** command and the **get** method of **HTable** class are used to read data from a table in HBase. Using **get** command, you can get a single row of data at a time. Its syntax is as follows:

```
get '<table name>','row1'
```

Example

The following example shows how to use the get command. Let us scan the first row of the **emp** table.

```
hbase(main):012:0> get 'emp', '1'

   COLUMN                      CELL

personal : city timestamp = 1417521848375, value = hyderabad

personal : name timestamp = 1417521785385, value = ramu

professional: designation timestamp = 1417521885277, value = manager

professional: salary timestamp = 1417521903862, value = 50000

4 row(s) in 0.0270 seconds
```

Reading a Specific Column

Given below is the syntax to read a specific column using the **get** method.

```
hbase> get 'table name', 'rowid', {COLUMN ⇒ 'column family:column name '}
```

Example

Given below is the example to read a specific column in HBase table.

```
hbase(main):015:0> get 'emp', 'row1', {COLUMN ⇒ 'personal:name'}
  COLUMN                CELL
personal:name timestamp = 1418035791555, value = raju
1 row(s) in 0.0080 seconds
```

Reading Data Using Java API

To read data from an HBase table, use the **get** method of the HTable class. This method requires an instance of the **Get** class. Follow the steps given below to retrieve data from the HBase table.

Step 1: Instantiate the Configuration Class

Configuration class adds HBase configuration files to its object. You can create a configuration object using the **create** method of the **HbaseConfiguration** class as shown below.

```
Configuration conf = HbaseConfiguration.create();
```

Step 2: Instantiate the HTable Class

You have a class called **HTable**, an implementation of Table in HBase. This class is used to communicate with a single HBase table. While instantiating this class, it accepts the configuration

object and the table name as parameters. You can instantiate the HTable class as shown below.

```
HTable hTable = new HTable(conf, tableName);
```

Step 3: Instantiate the Get Class

You can retrieve data from the HBase table using the **get** method of the **HTable** class. This method extracts a cell from a given row. It requires a **Get** class object as parameter. Create it as shown below.

```
Get get = new Get(toBytes("row1"));
```

Step 4: Read the Data

While retrieving data, you can get a single row by id, or get a set of rows by a set of row ids, or scan an entire table or a subset of rows.

You can retrieve an HBase table data using the add method variants in **Get** class.

To get a specific column from a specific column family, use the following method.

```
get.addFamily(personal)
```

To get all the columns from a specific column family, use the following method.

```
get.addColumn(personal, name)
```

Step 5: Get the Result

Get the result by passing your **Get** class instance to the get method of the **HTable** class. This method returns the **Result** class object, which holds the requested result. Given below is the usage of **get** method.

```
Result result = table.get(g);
```

Step 6: Reading Values from the Result Instance

The **Result** class provides the **getValue** method to read the values from its instance. Use it as shown below to read the values from the **Result** instance.

```
byte [] value = result.getValue(Bytes.toBytes("personal"),Bytes.toBytes("name"));
byte [] value1 = result.getValue(Bytes.toBytes("personal"),Bytes.toBytes("city"));
```

Given below is the complete program to read values from an HBase table.

```
import java.io.IOException;

import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.hbase.HBaseConfiguration;
import org.apache.hadoop.hbase.client.Get;
import org.apache.hadoop.hbase.client.HTable;
import org.apache.hadoop.hbase.client.Result;
import org.apache.hadoop.hbase.util.Bytes;

public class RetriveData{

   public static void main(String[] args) throws IOException, Exception{

      // Instantiating Configuration class
      Configuration config = HBaseConfiguration.create();
```

```
      // Instantiating HTable class
      HTable table = new HTable(config, "emp");

      // Instantiating Get class
      Get g = new Get(Bytes.toBytes("row1"));

      // Reading the data
      Result result = table.get(g);

      // Reading values from Result class object
      byte [] value = result.getValue(Bytes.toBytes("personal"),Bytes.toBytes("name"));

      byte [] value1 = result.getValue(Bytes.toBytes("personal"),Bytes.toBytes("city"));

      // Printing the values
      String name = Bytes.toString(value);
      String city = Bytes.toString(value1);

      System.out.println("name: " + name + " city: " + city);
   }
}
```

Compile and execute the above program as shown below.

```
$javac RetriveData.java
$java RetriveData
```

The following should be the output:

```
name: Raju city: Delhi
```