CS590BD Big Data Analytics and Apps

**LAB - 3**

**Report on collecting data from sensor tags**

**and pushing the collected data into HBase**

**using Webservices and Android APP**

**By**

**Ponnam Balakrishna**

**16177831(BP8G6)**

**Collecting Data using Sensor Tag:**

To collect the data we have many devices and I have chosen sensor tag to collect the data. From this sensor tag I am collecting Longitude, Latitude, Humidity, Pressure and accelerometer values like X, Y and Z.

To collect this data we need to follows some steps those are:

**Step1:**

Install the **BLE Sensor tag** App in android mobile having the OS higher than 4.3 versions.

Then run the App and Switch ON the sensor button .The App shows the sensor devices which are available.

**Step2:**

Those list all the connected devices. If we select any connected device from the shown list it displays all the list of services available. It shows all the list of services provided by this sensor those some of the services we are using are GPS which gives the longitude, latitude, Pressure, Humidity and the accelerometer values which is indicated in the terms of X,Y and Z. By selecting any list of the service we get some sub list of services which we are required for collecting the values.

By selecting any of these devices it open some of the sub tasks and by selecting those it gives the values of the particular selected item.

For collecting the particular data we need to enable the service tags of the particular services in the Android application.

**Step3:**

Run the android application and enable those services and their UUID tags. So that it collects the data of the enable UUID tags.

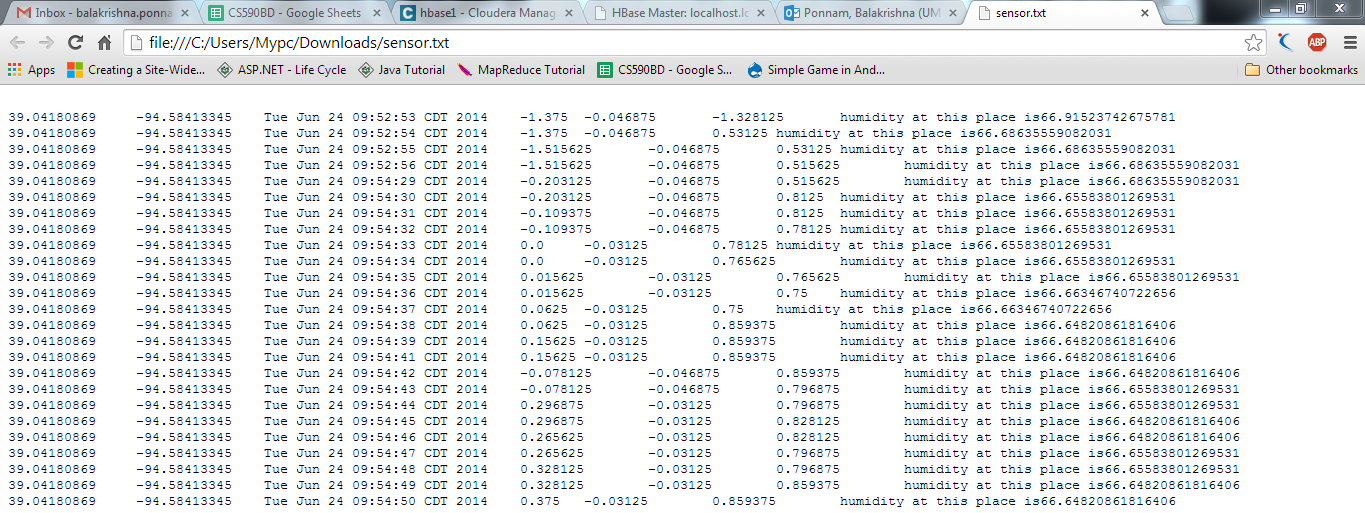
Those enabled and collected data is stored in a text document which can be getting by downloading **ASTRO FILE MANAGER APP** where the data is present in the Data folder of the ASTRO FILE MANAGER APP.

**Step4:**

The data is collected in this text document which needs to run on a java program to insert into the HBase.

The below screen shows the collected data from the sensor given with the tab spaces for each value

The below is the screen shot of collected data:

****

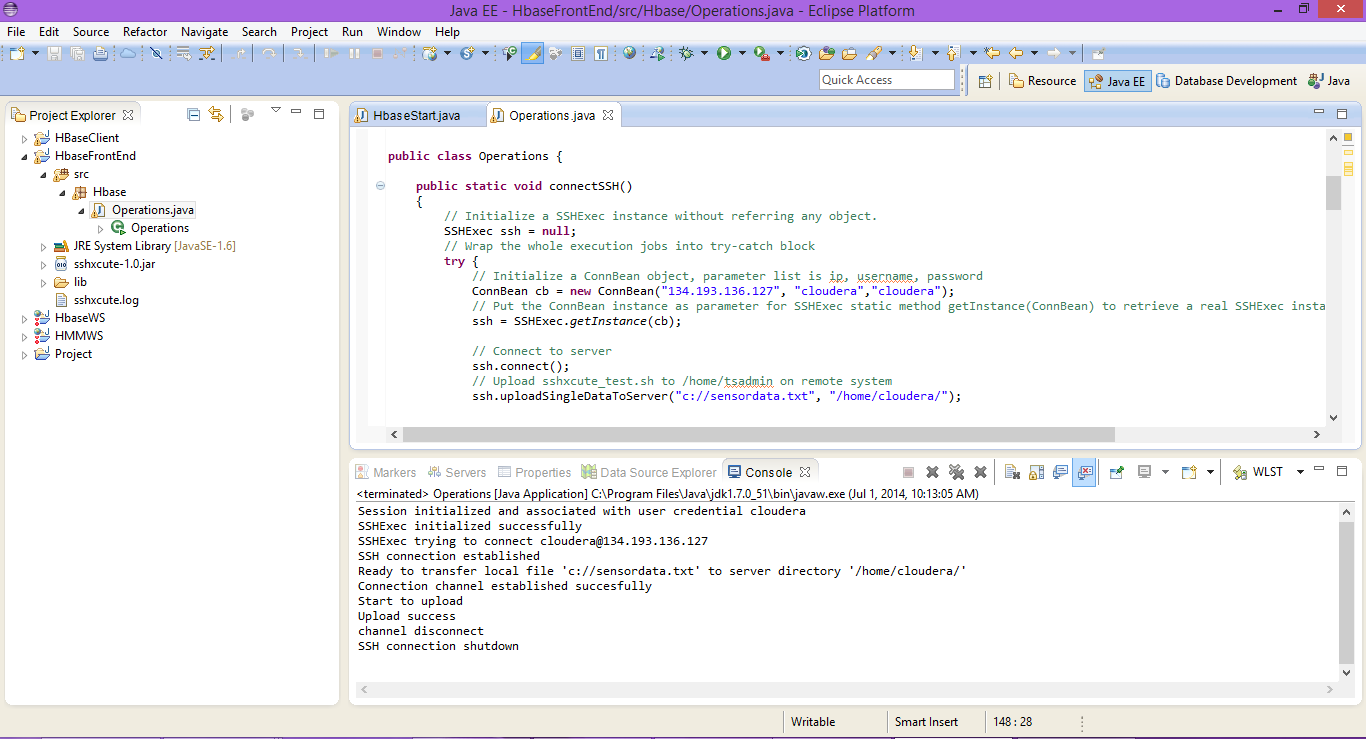
**File Transfer:**

For uploading the files into the hbase we need to mention the path. Our Local path of the system is transferred to the cloudera path.

We have written a java code for creating, updating and deleteing the path.

By running that code we can change the path and transfer the files.

The below is the screen shot Java code for the file transfer



Here we can see the method uploadingdataservice which changes our path and files can be exchanged.

**Pushing Data to Hbase Using WebService:**

In the lab2 we have seen the collection of sensor data and creating, inserting, retrieving and deleting of data from the HBase here the java code in written for those purpose.

In order to access that data from the web now we are creating the restful web service which can be accessed from anywhere.

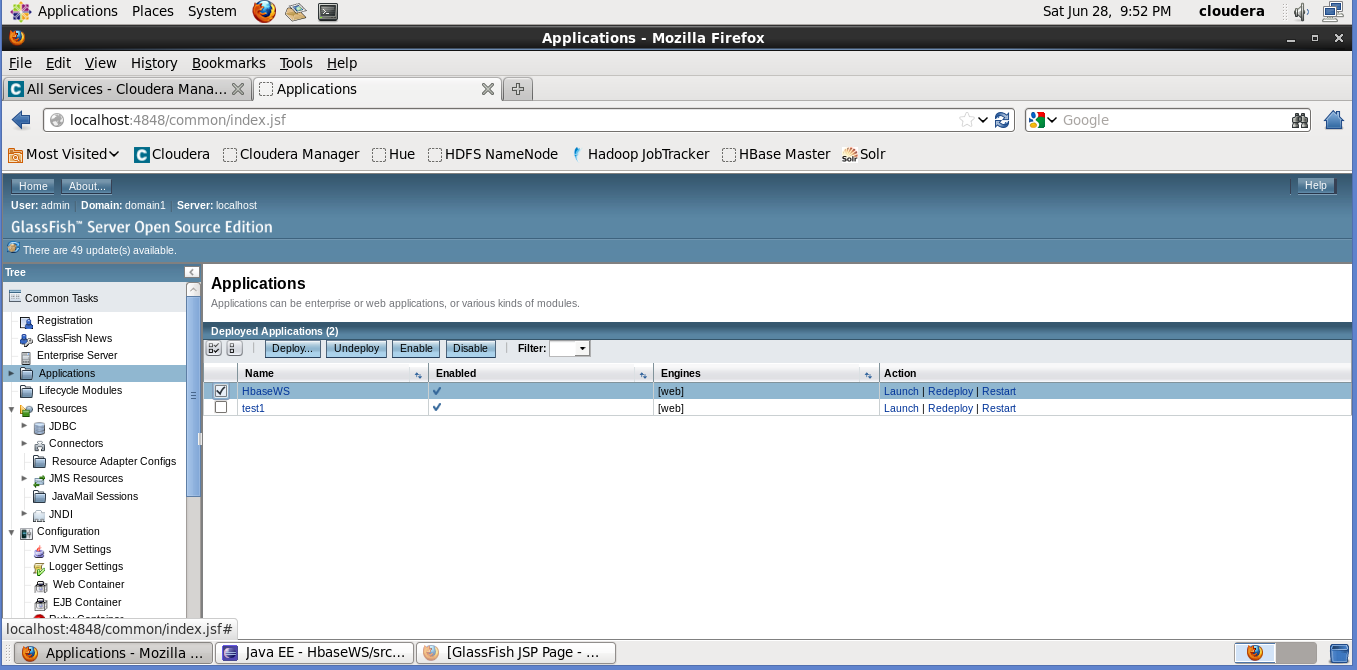
In order to create a restful service we need to install the glassfish server where we can deploy our projects. So those deployed data can be accessed from the web.

Installation of Glassfish Server:

To install the glass fish server need to download the Java EE eclipse from oracle. On installing the eclipse under the servers tab need to select new 🡪 servers. Then it asks the server which we need to select the “**Glass Fish Server from pattern 6**” Then Install that Glass Fish server.

After Completion of the installation type **localhost: 8080** if the Glass fish is running then it says that the glass fish is running. If we type localhost: 4848 we get the Console page of the admin UI. In the console of the glass fish page we can deploy the projects and can launch.

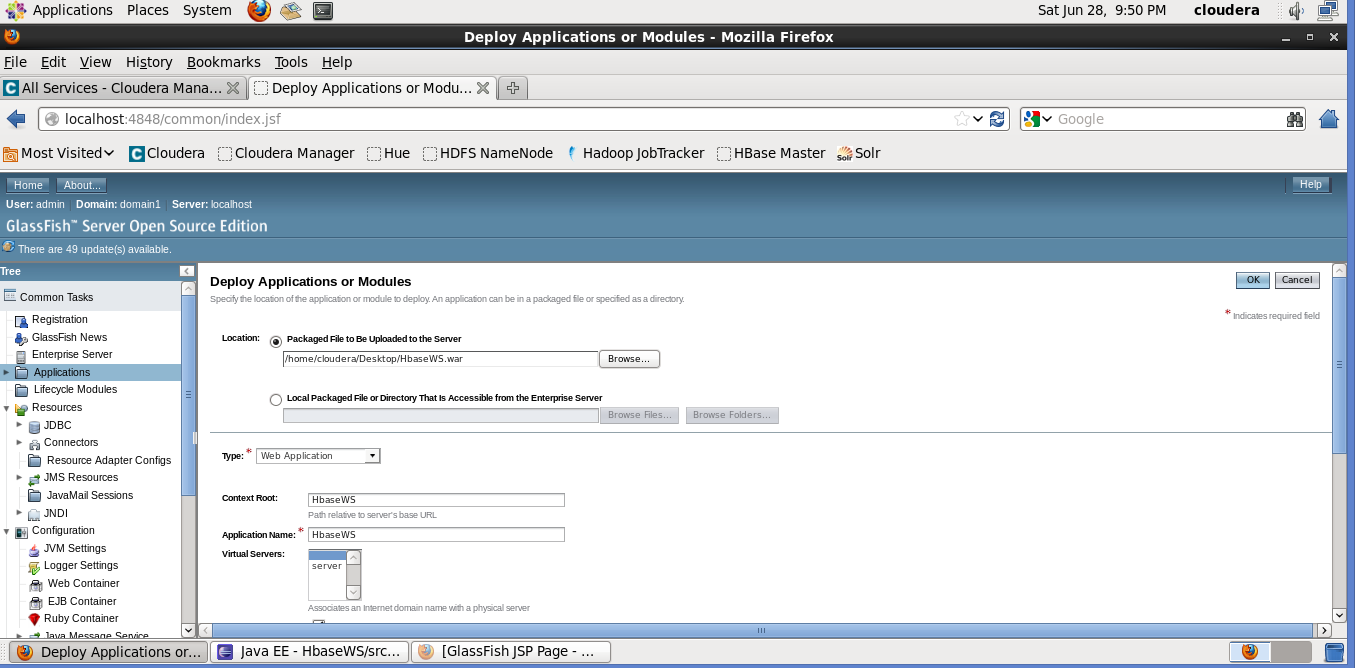
The below screen shows the console of the glassfish



In the above console we have the Deploy, UnDeploy, Enable, Disable options where Deploy is used to deploy an application and enable or disable says weather the device can be used or not. Enable indicates the Usage of device and the disable indicates not to use.

All the deployed applications will be shown as shown above. And if press the launch the program will be launched and shows the desired output of the program.

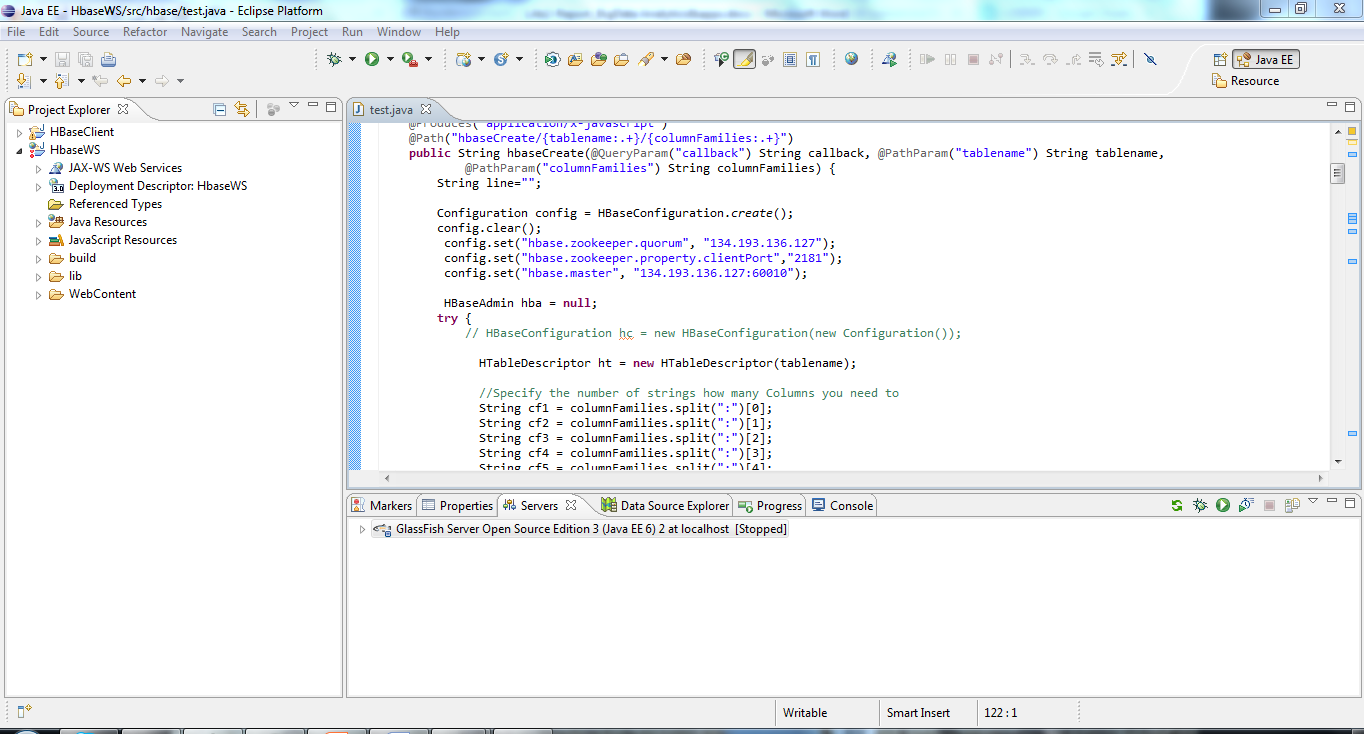
The below screen shows required fields:



For this project first we need to collect the data. The data is collected by using the senor. In lab2 we have already generated a file contain values of Gyroscope, Accelerometer, humidity and temperature.

Now by using that file we insert that file data into the hbase. We have written some code in the eclipse which create table and insert data into that table and retrieve the data.

The below screen shot shows the code to create the table:



By using the configuration settings we can specify the IP address of where to store the data weather in local cloudera or UMKC virtual Cloudera.

By using those written java code we can perform the CURD operations.

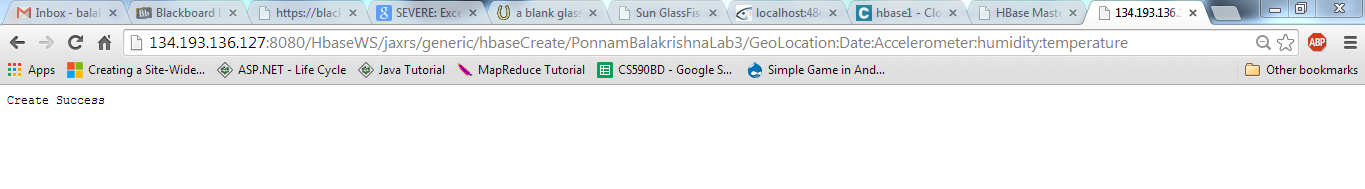
After deploying this code in the Glass Fish server we can access from the URL.

**To Create a Table:**

<http://134.193.136.127:8080/HbaseWS/jaxrs/generic/hbaseCreate/PonnamBalakrishnaLab3/GeoLocation:Date:Accelerometer:humidity:temperature>

The above is the url used to create the table, where PonnamBalakrishnaLab3 denotes the table name and the remaining GeoLocation:Date:Accelerometer are the coloumn families of the table.

The below is the screen shot of creating table:



**To Insert Data:**

Similarly we insert data by using the url.

<http://134.193.136.127:8080/HbaseWS/jaxrs/generic/hbaseInsert/PonnamBalakrishnaLab3/C:-Users-Mypc-Desktop-bp8g6.txt/GeoLocation:Date:Accelerometer:humidity:temperature>

The above is the url used to insert the text file data into the table. Here we specify the table name, file address, and the columns in the text file we are uploading.

**To Retrieve Data:**

We can also retriew the inserted data

<http://134.193.136.127:8080/HbaseWS/jaxrs/generic/hbaseRetrieveAll/PonnamBalakrishnaLab3>

The above is the url used to retrieve data from the table.

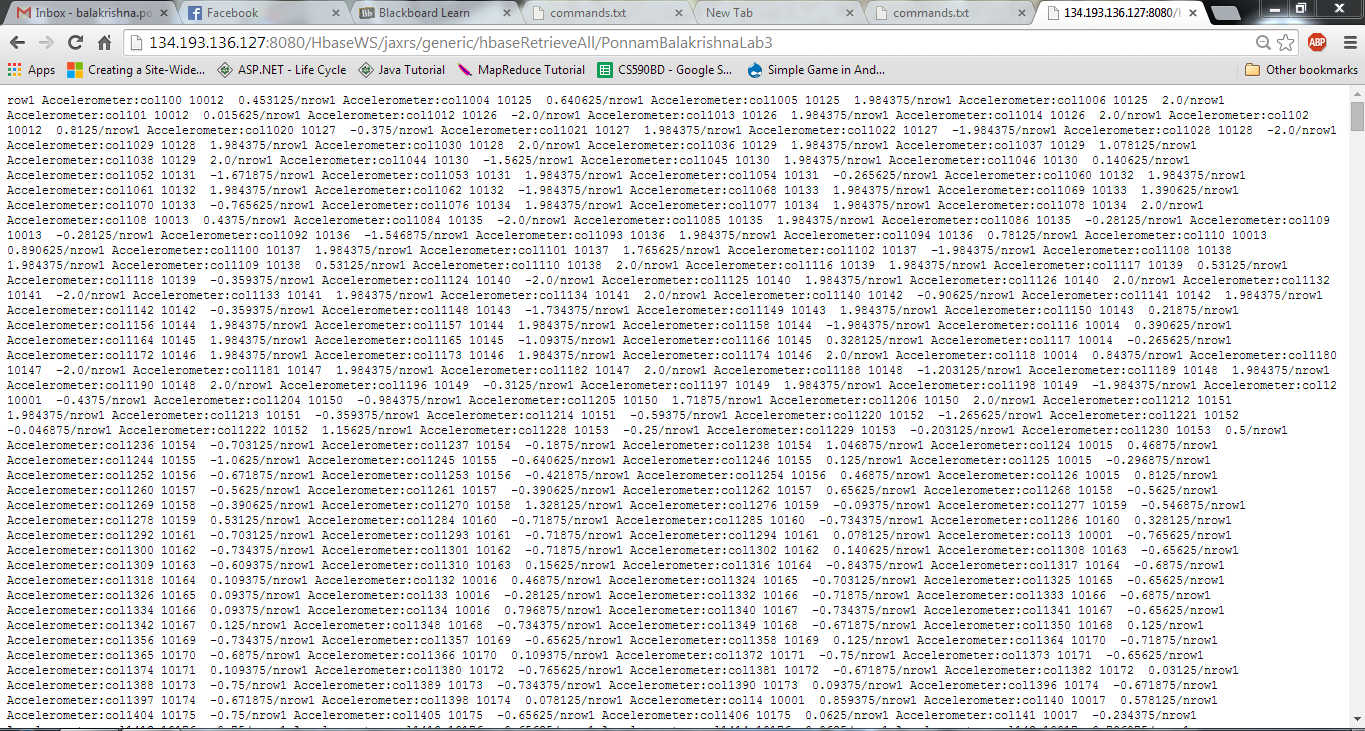
**To delete Table:**

The created table can also be deleted and it is deleted by using the following URL

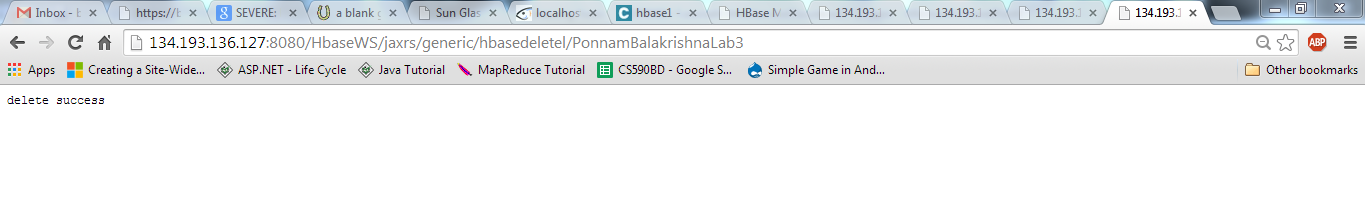
<http://134.193.136.127:8080/HbaseWS/jaxrs/generic/hbasedeletel/PonnamBalakrishnaLab3>

Those table can be deleted if the data is inserted also which deletes the data and the table.

The below screenshot shows the retrieve of the data

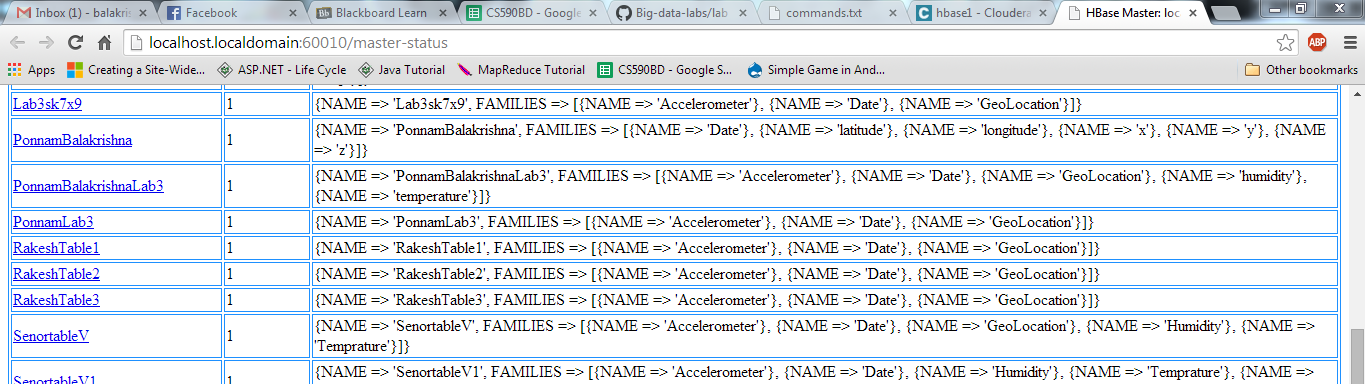


The below screen shot shows the deletion of the table



The created tables can see in the list of cloudera GUI page.

The below screen shot shows the list of all the created tables:



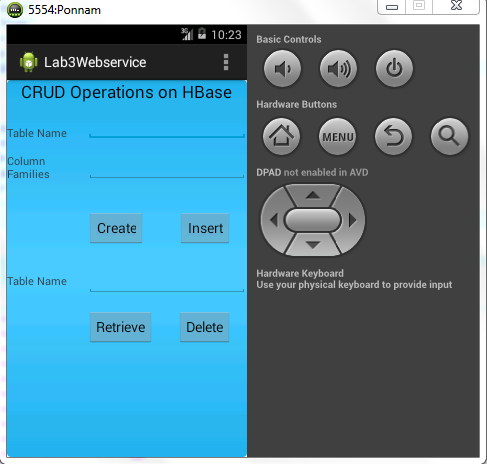
Those Table names are displayed with the column familes with the column names

**Developing Android Application for Performing CURD operations on HBase:**

The android is developed to perform CURD operations. By using this app we can create the tables with column families.

In this app we specified the Textbox for table name and another textbox for column families.

The below is the screen shot for the application:



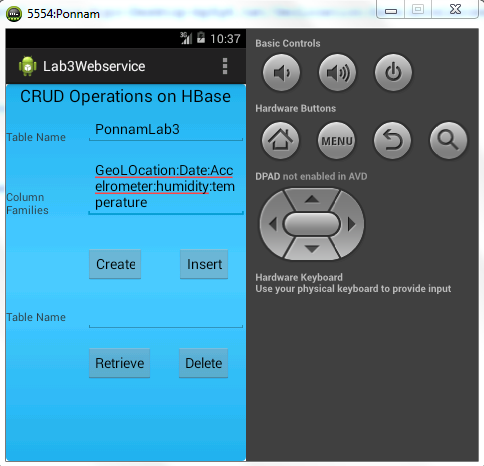
For Creating the table we need to specify the table name and the column families and click on the create button. Thus it gets connect to the hbase and creates the table. And we get the messga of sucessful insertion.

For Inserting the Data in the code it is mentioned the path from where the file need to be taken to upload.

For Retrieve and Delete we need to mention the table name and click on the particular button. Then it gets connected to the hbase and perform the particular operation.

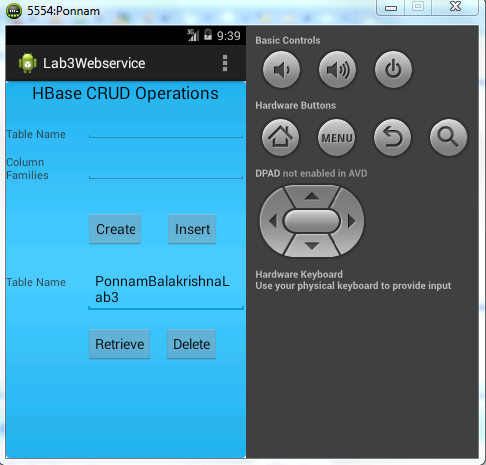
The below are the screen shots of the operations:

For Create and Insert operations:



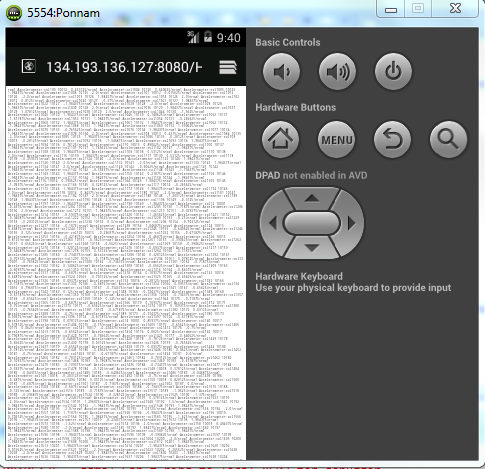
On creating or inserting it displays a message create success/insert success

For Retrieve and Delete the screen shot is shown as below



On entering the table name and clicking the Retrieve button the Data uploaded to hbase and is shown in the screen.

The below screen shot shows the retrieve of data from the hbase



On clicking the delete button the table will be deleted from the HBase and shows the message with delete success.