**WCMapper.java**

import java.io.IOException;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Mapper;

public class WCMapper extends Mapper<Object, Text, Text, IntWritable> {

private final static IntWritable one = new IntWritable(1);

private Text word = new Text();

public void map(Object key, Text value, Context context) throws IOException, InterruptedException {

String line = value.toString();

String user = line.split(",")[0];

if(user.length()>0) {

word.set(user);

context.write(word, one);

}

}

}

**WCReducer.java**

import java.io.IOException;

import java.util.HashSet;

import java.util.Set;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Reducer;

public class WCReducer extends Reducer<Text, IntWritable, Text, IntWritable> {

private Set<String> uniqueWords = new HashSet<>();

private IntWritable result = new IntWritable();

public void reduce(Text key, Iterable<IntWritable> values, Context context) throws IOException, InterruptedException {

String word = key.toString();

uniqueWords.add(word);

}

protected void cleanup(Context context) throws IOException, InterruptedException {

result.set(uniqueWords.size());

context.write(new Text("Unique Listeners Count: "), result);

}

}

**WCDriver.java**

import org.apache.hadoop.conf.Configuration;

import org.apache.hadoop.fs.Path;

import org.apache.hadoop.io.IntWritable;

import org.apache.hadoop.io.Text;

import org.apache.hadoop.mapreduce.Job;

import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;

import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class WCDriver{

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

Configuration conf = new Configuration();

Job job = Job.getInstance(conf, "word count distinct");

job.setJarByClass(WCDriver.class);

job.setMapperClass(WCMapper.class);

job.setReducerClass(WCReducer.class);

job.setOutputKeyClass(Text.class);

job.setOutputValueClass(IntWritable.class);

FileInputFormat.addInputPath(job, new Path(args[0]));

FileOutputFormat.setOutputPath(job, new Path(args[1]));

System.exit(job.waitForCompletion(true) ? 0 : 1);

}

}