

TutorialOneCompleteSourceListing

```
package org.apache.nutch.examples

import java.io.IOException;
import java.util.ArrayList;
import java.util.Iterator;
import java.util.List;

import org.apache.hadoop.conf.*;
import org.apache.hadoop.fs.*;
import org.apache.hadoop.io.*;
import org.apache.hadoop.mapred.*;

import org.apache.nutch.parse.*;
import org.apache.nutch.util.*;

public class LinkCounter {

    public static class CounterMapper extends MapReduceBase implements Mapper
    {
        public void map(WritableComparable key, Writable value, OutputCollector collector, Reporter
reporter) throws IOException {
            // TODO Auto-generated method stub
            ParseData data = (ParseData)value;

            IntWritable outboundLinkCount = new IntWritable(data.getOutlinks().
length);

            collector.collect(key, outboundLinkCount);
        }

        public void close() throws IOException {
            // TODO Auto-generated method stub
            super.close();
        }

        public void configure(JobConf arg0) {
            // TODO Auto-generated method stub
            super.configure(arg0);
        }
    }

    public static class CounterReducer extends MapReduceBase implements Reducer
    {
        public void reduce(WritableComparable url, Iterator iterator, OutputCollector output, Reporter
reporter) throws IOException {
            IntWritable linkCount = (IntWritable)iterator.next();
            output.collect(url, linkCount);
        }

        public void close() throws IOException {
            // TODO Auto-generated method stub
            super.close();
        }

        public void configure(JobConf arg0) {
            // TODO Auto-generated method stub
            super.configure(arg0);
        }
    }

    public static void main(String[] args) throws IOException{
        Configuration config = NutchConfiguration.create();
```

```
JobConf jobConfig = new NutchJob(config);
jobConfig.setJobName("countlinks");

jobConfig.setInputFormat(SequenceFileInputFormat.class);

jobConfig.setOutputFormat(MapFileOutputFormat.class);

// the keys are words (strings)
jobConfig.setOutputKeyClass(Text.class);
// the values are counts (ints)
jobConfig.setOutputValueClass(IntWritable.class);

jobConfig.setMapperClass(CounterMapper.class);
jobConfig.setCombinerClass(CounterReducer.class);
jobConfig.setReducerClass(CounterReducer.class);

jobConfig.setInputPath(new Path((String) args[0], ParseData.DIR_NAME));
jobConfig.setOutputPath(new Path((String) args[1]));

JobClient.runJob(jobConfig);
}
}
```