

# Word Count with MapReduce\*

Montasser AKERMI

Last Update: March, 2025

---

\*The course material is hosted at <https://akermi.org/c2/>.

## Learning objectives

Write a MapReduce application that counts the total number of each word in a file or a folder containing multiple files.

## Input data

my dreams my dreams  
what has become of their sweetness  
what indeed has become of my youth

## Output data

sweetness 1  
become 2  
dreams 2  
indeed 1  
youth 1  
their 1  
what 2  
has 2  
of 2  
my 3

## Solution

Listing 1: WordCount Class

```
1 import java.io.IOException;
2 import java.util.StringTokenizer;
3
4 import org.apache.hadoop.conf.Configuration;
5 import org.apache.hadoop.fs.Path;
6 import org.apache.hadoop.io.IntWritable;
7 import org.apache.hadoop.io.Text;
8 import org.apache.hadoop.mapreduce.Job;
9 import org.apache.hadoop.mapreduce.Mapper;
10 import org.apache.hadoop.mapreduce.Reducer;
11 import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
12 import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
13
14 public class WordCount {
15
16     public static class TokenizerMapper
17         extends Mapper<Object, Text, Text, IntWritable> {
18
19         private final static IntWritable one = new IntWritable(1);
20         private Text word = new Text();
21
22         public void map(Object key, Text value, Context context
23             ) throws IOException, InterruptedException {
24             StringTokenizer itr = new StringTokenizer(value.toString());
25             while (itr.hasMoreTokens()) {
26                 word.set(itr.nextToken());
```

```

27         context.write(word, one);
28     }
29 }
30 }
31
32 public static class IntSumReducer
33     extends Reducer<Text, IntWritable, Text, IntWritable> {
34     private IntWritable result = new IntWritable();
35
36     public void reduce(Text key, Iterable<IntWritable> values,
37         Context context
38     ) throws IOException, InterruptedException {
39         int sum = 0;
40         for (IntWritable val : values) {
41             sum += val.get();
42         }
43         result.set(sum);
44         context.write(key, result);
45     }
46 }
47
48 public static void main(String[] args) throws Exception {
49     Configuration conf = new Configuration();
50     Job job = Job.getInstance(conf, "word count");
51     job.setJarByClass(WordCount.class);
52     job.setMapperClass(TokenizerMapper.class);
53     job.setCombinerClass(IntSumReducer.class);
54     job.setReducerClass(IntSumReducer.class);
55     job.setOutputKeyClass(Text.class);
56     job.setOutputValueClass(IntWritable.class);
57     FileInputFormat.addInputPath(job, new Path(args[0]));
58     FileOutputFormat.setOutputPath(job, new Path(args[1]));
59     System.exit(job.waitForCompletion(true) ? 0 : 1);
60 }
61 }

```

*Note: It is highly recommended to manually write the code instead of copy/pasting.*