

# Final Project

BigData



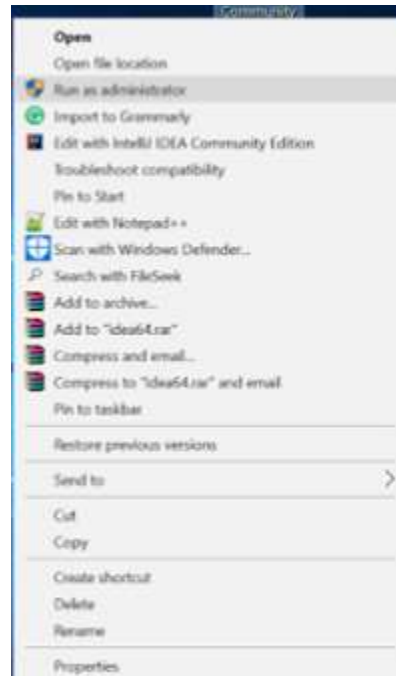
Baggio Deroger

1941720238 – TI 3H

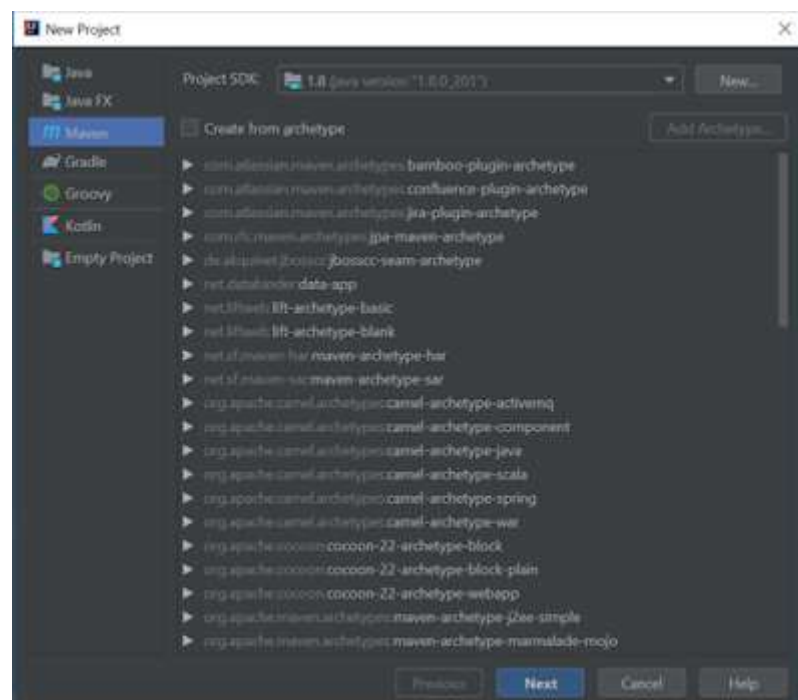
June 2022

# Making and Running Hadoop MapReduce Applications on Windows OS

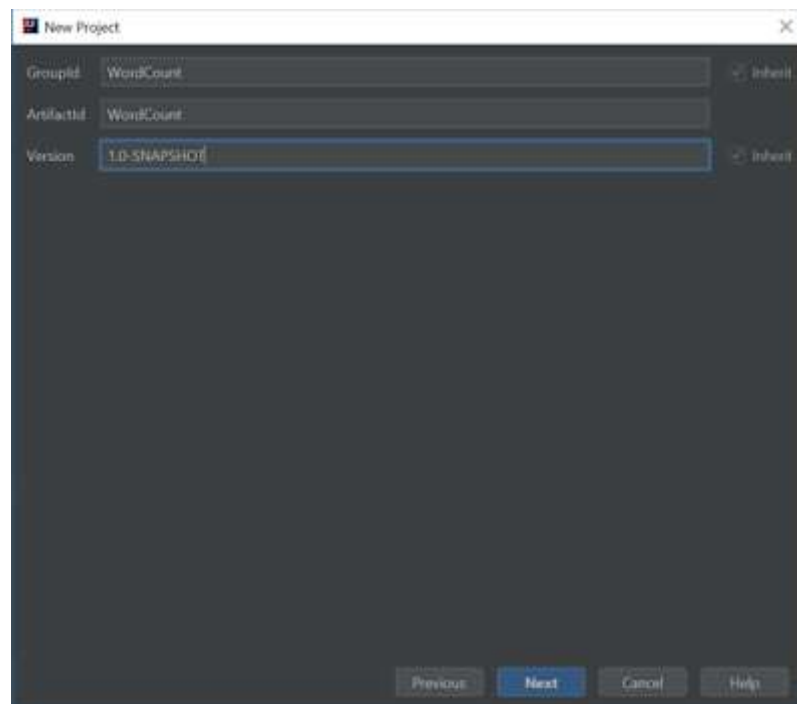
1. Start IntelliJ IDEA as Administrator



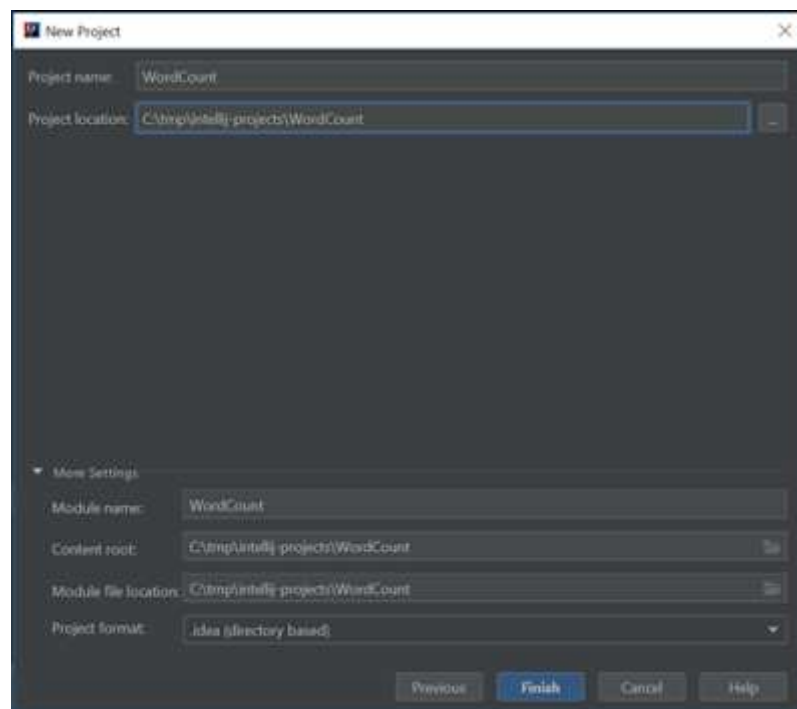
2. Creating a Maven Project from the File menu > New > Project > Maven (please follow the picture, then finally click Finish):  
New Project -> Maven -> Next



GroupId, ArtifactId, Version -> Next

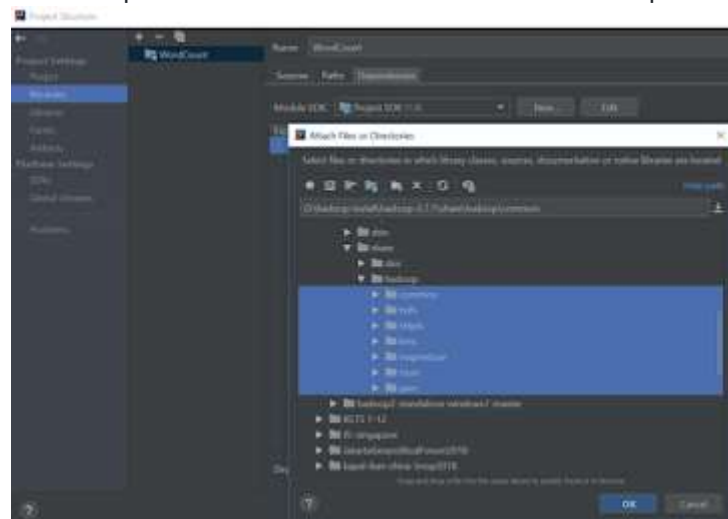


Project Name, Project location > Finish

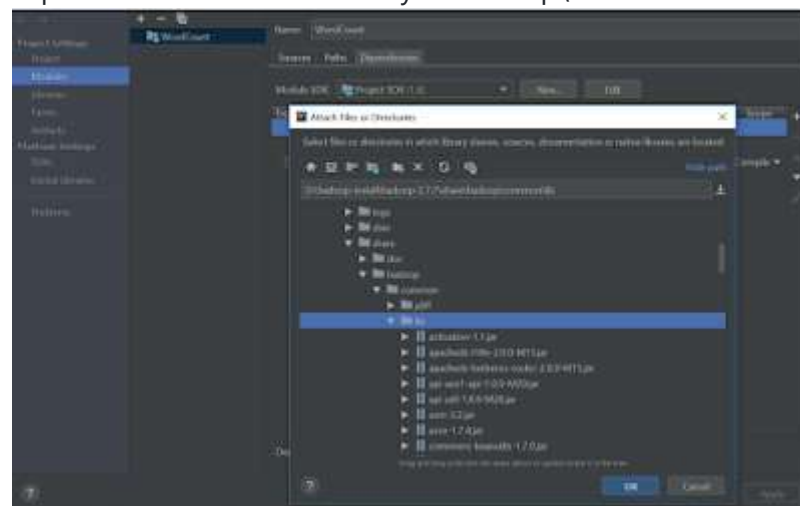


3. Load Hadoop Library from File -> Project Structure -> Modules -> Dependencies -> + -> 1 JARs or directories...

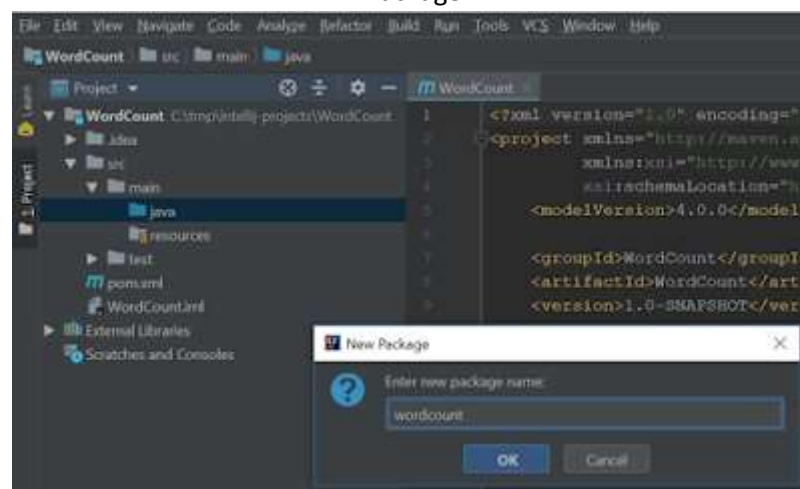
Dependencies: all sub-directories on hadoop



Dependencies: lib sub-directory of hadoop\common . directory

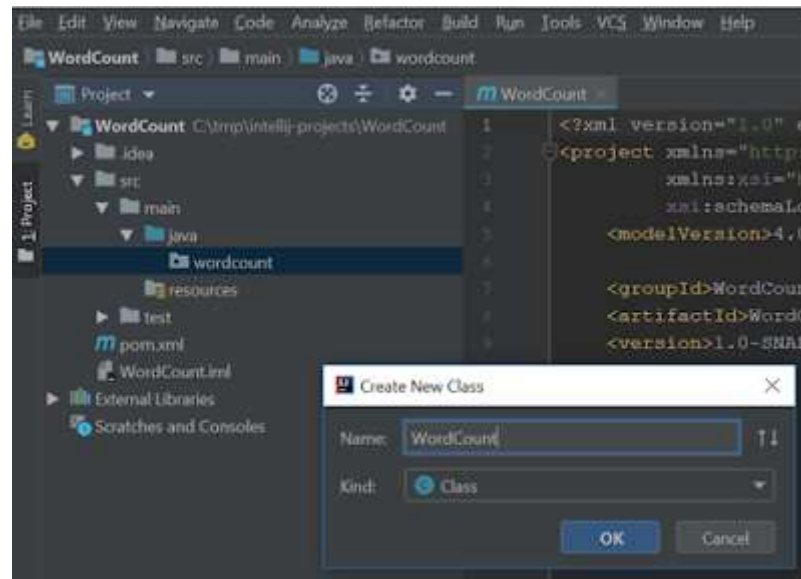


4. Create Java Package "wordcount" by right click WordCount -> src -> main -> java -> New -> Package



5. Create Java Class "WordCount.java" by right click WordCount -> src -> main -> java -> wordcount-> New -> Java Class

New Class > WordCount



6. Start coding

```
package  
wordcount;
```

```
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.Mapper;  
import org.apache.hadoop.mapreduce.Reducer;  
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;  
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;  
  
import java.io.IOException;  
import java.util.StringTokenizer;  
  
public class WordCount {  
    public static class TokenizerMapper  
        extends Mapper<Object, Text, Text, IntWritable>{  
  
        private final static IntWritable ONE = new IntWritable(1);  
        private final Text word = new Text();
```

```

@Override
public void map(Object key, Text value, Context context
) throws IOException, InterruptedException {
    StringTokenizer itr = new StringTokenizer(value.toString());
    while (itr.hasMoreTokens()) {
        word.set(itr.nextToken());
        context.write(word, ONE);
    }
}

public static class IntSumReducer
    extends Reducer<Text,IntWritable,Text,IntWritable> {
    private final IntWritable result = new IntWritable();

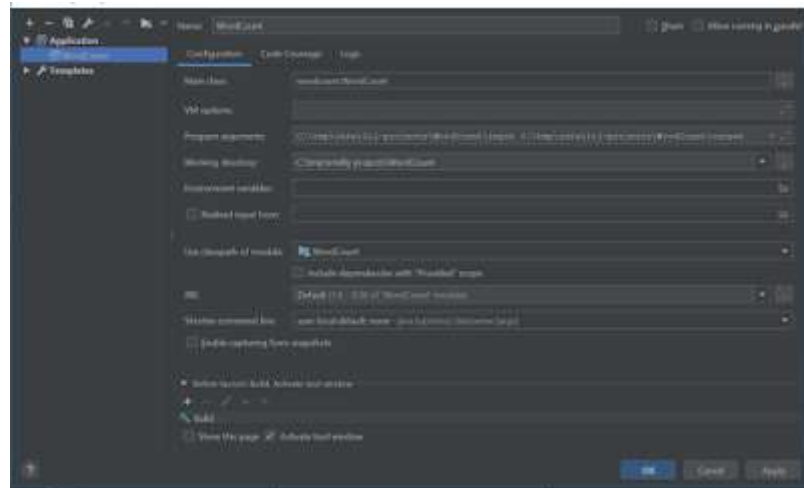
    @Override
    public void reduce(Text key, Iterable<IntWritable> values,
        Context context
    ) throws IOException, InterruptedException {
        int sum = 0;
        for (IntWritable val : values) {
            sum += val.get();
        }
        result.set(sum);
        context.write(key, result);
    }
}

public static void main(String[] args) throws Exception {
    Configuration conf = new Configuration();
    Job job = Job.getInstance(conf, "word count");
    job.setJarByClass(WordCount.class);
    job.setMapperClass(TokenizerMapper.class);
    job.setCombinerClass(IntSumReducer.class);
    job.setReducerClass(IntSumReducer.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);
}

```

7. Edit the configuration to run the WordCount program from the Run menu
  - > Edit Configuration... > + > Application (Remember! create an input directory, but DO NOT create an output directory. Then put any text file in the input directory)

## Run Configuration



- Run the application from the Run menu > Run 'WordCount' (The result is as shown)

