

```

package stubs;

import org.apache.hadoop.fs .Path;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.conf.Configured;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;

//importing relevant tools custom search in CLI
import org.apache.hadoop.conf.Configured;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.util.Tool;
import org.apache.hadoop.util.ToolRunner;
import org.apache.hadoop.mapreduce.lib.reduce.IntSumReducer;

public class SearchVideoDriver extends Configured implements Tool{

    public static void main(String[] args) throws Exception {
        Configuration conf = new Configuration();
        conf.set("SearchWord", "");
        conf.setInt("MinimumLikes", 0);
        int exitCode = ToolRunner.run(conf, new SearchVideoDriver(), args);
        System.exit(exitCode);
    }

    public int run(String[] args) throws Exception {

        if (args.length != 2) {
            System.out.printf("Usage: SearchVideo <input dir> <output dir>\n");
            System.exit(-1);
        }

        //create new job and assign jar class and job name
        Job job = new Job(getConf());
        job.setJarByClass(SearchVideoDriver.class);
        job.setJobName("Search Video");

        //input loc is the first arg, output loc is second
        FileInputFormat.addInputPath(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));

        //output relevant to <searched_title, search_min_likes>, <Text, IntWritable>
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(IntWritable.class);

        //TASK 1. Using the IntSumReducer, since the output mapper value is IntWritable.
        job.setReducerClass(IntSumReducer.class);
        job.setMapperClass(SearchVideoMapper.class);

        boolean success = job.waitForCompletion(true);
        return (success ? 0 : 1);
    }
}

```