```
package sparkAssignmentRDDfetch;


import java.util.List;
import java.util.Arrays;

import org.apache.hadoop.fs.Path;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.spark.SparkConf;
import org.apache.spark.api.java.JavaPairRDD;
import org.apache.spark.api.java.JavaRDD;
import org.apache.spark.api.java.JavaSparkContext;
import org.apache.spark.api.java.function.Function;

public class SparkRDD1 {
    public static void main (String args[])
    {
        //SparkConf conf = new
SparkConf().setAppName("Spark1").setMaster("local[*]");
        SparkConf conf = new SparkConf().setAppName("Spark1");
        JavaSparkContext sc = new JavaSparkContext(conf);

        //JavaRDD<String> lines = sc.textFile("data/yellow_tripdata_*.csv");

        //args[0] take the input file
        JavaRDD<String> dataset = sc.textFile(args[0]);

        //Single record lookup
        JavaRDD<String> Record= dataset.filter(
                records -> {
                    String[] vals = records.trim().split(",");
                    if (vals[0].contentEquals("2")
                            && vals[1].contentEquals("2017-10-01
00:15:30")
                            && vals[2].contentEquals("2017-10-01
00:25:11")
                            &&  vals[3].contentEquals("1")
                            && vals[4].contentEquals("2.17")
                            ) {
                        return true;
                    }
                    return false;
                }
                );
        System.out.println(Record);
        Record.foreach(x->System.out.println(x));

        //Consider we need 2 output files
        JavaRDD<String> newData = Record.coalesce(2);

        //args[1] stores the output in the location which the user will define
        newData.saveAsTextFile(args[1]);
    }
}
```