

```

package com.analyticsanvil.custominputformat;

import java.io.IOException;
import org.apache.hadoop.fs.FileSystem;
import org.apache.hadoop.fs.Path;

import org.apache.hadoop.io.BytesWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.mapred.FileSplit;
import org.apache.hadoop.mapred.InputSplit;
import org.apache.hadoop.mapred.RecordReader;
import org.apache.hadoop.mapred.FixedLengthInputFormat;
import org.apache.hadoop.mapred.JobConf;
import org.apache.hadoop.mapred.Reporter;

public class CustomTextInputFormat extends FixedLengthInputFormat {

    @Override
    public RecordReader<LongWritable, BytesWritable> getRecordReader(InputSplit genericSplit, JobConf job, Reporter reporter) throws IOException {
        reporter.setStatus(genericSplit.toString());

        long recordLength;

        FileSplit split = (FileSplit) genericSplit;
        Path path = split.getPath();
        FileSystem fs = path.getFileSystem(job);
        recordLength = fs.getFileStatus(path).getLen();

        // Set record length to the size of the fixed file currently being read
        FixedLengthInputFormat.setRecordLength(job, (int) recordLength);

        recordLength = getRecordLength(job);

        if (recordLength <= 0) {
            throw new IOException(
                "Fixed record length "
                + recordLength
                + " is invalid. It should be set to a value greater than zero");
        }

        return new CustomTextRecordReader(genericSplit, job, reporter);
    }

    @Override
    protected boolean isSplittable(FileSystem fs, Path file) {
        // Ensure mappers only process individual files (grid files should be read fully before being transformed)
        return false;
    }
}

```