InputFileToMapper-Hadoop

The input to the mapper depends on what InputFormat is used. The InputFormat is responsible for reading the incoming data and shaping it into whatever format the Mapper expects.The default InputFormat is [TextInputFormat](http://hadoop.apache.org/docs/current/api/org/apache/hadoop/mapreduce/lib/input/TextInputFormat.html), which extends FileInputFormat<LongWritable, Text>.

<https://hadoop.apache.org/docs/r2.3.0/api/org/apache/hadoop/mapred/TextInputFormat.html>

## Class TextInputFormat (Default)

An [InputFormat](https://hadoop.apache.org/docs/r2.3.0/api/org/apache/hadoop/mapred/InputFormat.html" \o "interface in org.apache.hadoop.mapred) for plain text files. Files are broken into lines. Either linefeed or carriage-return are used to signal end of line. Keys are the position in the file, and values are the line of text.

**public** **class** MaxTemperatureMapper **extends**

Mapper<LongWritable, Text, Text, IntWritable> {

@Override

**public** **void** map(LongWritable key, Text value, Context context)

**throws** IOException, InterruptedException {

}

}

If you do not change the InputFormat, using a Mapper with different Key-Value type signature than <LongWritable, Text> will cause this error. If you expect <Text, Text> input, you will have to choose an appropiate InputFormat. You can set the InputFormat in Job setup:

job.setInputFormat(MyInputFormat.class);

jpb.setInputFormatClass(KeyValueTextInputFormat.class);

**Class KeyValueTextInputFormat**

An [InputFormat](https://hadoop.apache.org/docs/r2.3.0/api/org/apache/hadoop/mapred/InputFormat.html" \o "interface in org.apache.hadoop.mapred) for plain text files. Files are broken into lines. Either linefeed or carriage-return are used to signal end of line. Each line is divided into key and value parts by a separator byte. If no such a byte exists, the key will be the entire line and value will be empty. <https://hadoop.apache.org/docs/r2.3.0/api/org/apache/hadoop/mapred/KeyValueTextInputFormat.html>

**public** **static** **class** FileMapper

**extends** Mapper<Text, Text, IntWritable, Point> {

**public** **void** map(Text key, Text value, Context context)

**throws** IOException, InterruptedException

{

}

}