

Explanation of SalesMapper Class

In this section, we will understand the implementation of **SalesMapper** class.

1. We begin by specifying a name of package for our class. **SalesCountry** is a name of our package. Please note that output of compilation, **SalesMapper.class** will go into a directory named by this package name: **SalesCountry**.

Followed by this, we import library packages.

Below snapshot shows an implementation of **SalesMapper** class-

The screenshot shows an IDE with two tabs: `*SalesCountryDriver.java` and `*SalesMapper.java`. The `SalesMapper.java` file contains the following code:

```
package SalesCountry;

import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.*;

public class SalesMapper extends MapReduceBase implements Mapper<LongWritable, Text, Text, IntWritable> {
    private final static IntWritable one = new IntWritable(1);

    public void map(LongWritable key, Text value, OutputCollector<Text, IntWritable> output, Reporter reporter) throws IOException {
        String valueString = value.toString();
        String[] SingleCountryData = valueString.split(",");
        output.collect(new Text(SingleCountryData[7]), one);
    }
}
```

Annotations in the image:

- Package Name:** Points to `package SalesCountry;`
- Import Library Packages:** Points to the import statements for `java.io.IOException`, `org.apache.hadoop.io.IntWritable`, `org.apache.hadoop.io.LongWritable`, `org.apache.hadoop.io.Text`, and `org.apache.hadoop.mapred.*`.
- Every Map class must extend 'MapReduceBase' class and implement 'Mapper' interface:** Points to `extends MapReduceBase implements Mapper`.
- Every Mapper class must provide definition of 'map' function:** Points to the `map` method.
- Our mapper function maps every input record to '1':** Points to the `output.collect(new Text(SingleCountryData[7]), one);` line.

Sample Code Explanation:

1. SalesMapper Class Definition-

```
public class SalesMapper extends MapReduceBase implements
Mapper<LongWritable, Text, Text, IntWritable> {
```

Every mapper class must be extended from **MapReduceBase** class and it must implement **Mapper** interface.

2. Defining 'map' function-

```
public void map(LongWritable key,  
               Text value,  
               OutputCollector<Text, IntWritable> output,  
               Reporter reporter) throws IOException
```

The main part of Mapper class is a '**map()**' method which accepts four arguments.

At every call to '**map()**' method, a **key-value** pair ('**key**' and '**value**' in this code) is passed.

'**map()**' method begins by splitting input text which is received as an argument. It uses the tokenizer to split these lines into words.

```
String valueString = value.toString();  
String[] SingleCountryData = valueString.split(",");
```

Here, ',' is used as a delimiter.

After this, a pair is formed using a record at 7th index of array '**SingleCountryData**' and a value '**1**'.

```
output.collect(new Text(SingleCountryData[7]), one);
```

We are choosing record at 7th index because we need **Country** data and it is located at 7th index in array '**SingleCountryData**'.

Please note that our input data is in the below format (where **Country** is at 7th index, with 0 as a starting index)-

Transaction_date,Product,Price,Payment_Type,Name,City,State,**Country**,Account_Created,Last_Login,Latitude,Longitude

An output of mapper is again a **key-value** pair which is outputted using '**collect()**' method of '**OutputCollector**'.

Explanation of SalesCountryReducer Class

In this section, we will understand the implementation of **SalesCountryReducer** class.

1. We begin by specifying a name of the package for our class. **SalesCountry** is a name of our package. Please note that output of compilation, **SalesCountryReducer.class** will go into a directory named by this package name: **SalesCountry**.

Followed by this, we import library packages.

Below snapshot shows an implementation of **SalesCountryReducer** class-

```
*SalesCountryDriver.java x *SalesMapper.java x *SalesCountryReducer.java x

package SalesCountry;

import java.io.IOException;
import java.util.*;

import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.*;

public class SalesCountryReducer extends MapReduceBase implements Reducer<Text, IntWritable, Text, IntWritable> {

    public void reduce(Text t_key, Iterator<IntWritable> values,
        OutputCollector<Text,IntWritable> output,
        Reporter reporter) throws IOException {

        Text key = t_key;
        int frequencyForCountry = 0;

        while (values.hasNext()) {
            // replace type of value with the actual type of our value
            IntWritable value = (IntWritable) values.next();
            frequencyForCountry += value.get();
        }

        output.collect(key, new IntWritable(frequencyForCountry));
    }
}
```

Package Name

Import Library Packages

Every 'Reducer' class must extend 'MapReduceBase' class and implement 'Reducer' interface

Every 'Reducer' class must provide definition of 'reduce' function