Splitter Class Part 2 MapSplitter

In previous posts we saw introduction to [Google Guava](http://data-structure-learning.blogspot.com/2015/05/google-guava-introduction.html), [Joiner class](http://data-structure-learning.blogspot.com/2015/05/joiner-class-part-1.html), [MapJoiner class](http://data-structure-learning.blogspot.com/2015/05/joiner-class-part-2-mapjoiner.html) and [Splitter Class](http://data-structure-learning.blogspot.com/2015/06/splitter-class-part-1.html).

In this post we will learn the MapSplitter class. It is a static class inside of Splitter class.

So let’s say that we have a set of key-value pairs separated by some separator and different entries separated by some separator.

Example:

String startString = "A=1#B=2#C=3";

So the key and value are separated by “=” and entries are separated by “#”.

To convert this String into the Map we can use MapSplitter class.

/\*\*

\* This method is used to split the string in key-value pair.

\* So the input is the string with key-value seperator

\* followed by Entry seperator.

\* \*/

**public** **static** **void** mapSplitter() {

String startString = "A=1#B=2#C=3";

System.***out***.println(startString);

Splitter.MapSplitter mapSplitter = Splitter.*on*("#").withKeyValueSeparator("=");

Map<String, String> map = mapSplitter.split(startString);

System.***out***.println(map);

}

Output

A=1#B=2#C=3

{A=1, B=2, C=3}

So this is how we can convert the String to map using Google Guava.