Joiner class MapJoiner.

In previous post we saw why and how [Joiner](http://data-structure-learning.blogspot.com/2015/05/joiner-class-part-1.html) class of Google Guava is useful to us. In this post we will explore how to Joiner class will work for Map entries.

MapJoiner is a static class provided in Joiner class. Its job is to join map entries in same manner as Joiner used to join iterables and arrays. Also, like Joiner, MapJoiner is also Immutable.

Map<String, String> map=Maps.*newLinkedHashMap*();

map.put("Name", "John");

map.put("LastName", "Doe");

map.put("Address", "California");

Above Map would print like this using toString() method.

{Name=John, LastName=Doe, Address=California}

Maps.newLinkedHashMap() is a static factory that is used to return new LinkedHashMap<> every time the method is called. Remember this static factory is different from Factory method design pattern. Below is the code of static factory.

**public** **static** <K, V> LinkedHashMap<K, V> newLinkedHashMap() {

**return** **new** LinkedHashMap<K, V>();

}

Now let’s say we want to separate our key and value by “:” and multiple entries by “,” and then join all the entries.

So for this Guava provides an API to perform this operation.

Joiner.*on*(", ").withKeyValueSeparator(":").join(map);

We know about Joiner class, on([String|CharSequence]).

New thing over here is a method withKeyValueSeparator(String keyValueSeparator). Below is the code for this method.

**public** MapJoiner withKeyValueSeparator(String keyValueSeparator) {

**return** **new** MapJoiner(**this**, keyValueSeparator);

}

So this method returns the new instance of MapJoiner.

And then join method is used to join the entrySet by taking the iterator of the entrySet.

Below is the code for withKeyValueSeparator().

**package** com.google.guava.learning.chapter1;

**import** java.util.Map;

**import** com.google.common.base.Joiner;

**import** com.google.common.collect.Maps;

**public** **class** WithKeyValueSeperatorDemo {

**public** **static** **void** main(String[] args) {

*withKeyValueSperatorDemo*();

}

**public** **static** **void** withKeyValueSperatorDemo(){

Map<String, String> map=Maps.*newLinkedHashMap*();

map.put("Name", "John");

map.put("LastName", "Doe");

map.put("Address", "California");

System.***out***.println("Map toString() "+map);

// {Name=John, LastName=Doe, Address=California}

/\*\*

\* Here we add , as delimiter.

\* And then we add : as key value separator.

\* So output for above map will be

\* Name:John , LastName:Doe , Address:California

\* \*/

String str=Joiner.*on*(", ").withKeyValueSeparator(":").join(map);

System.***out***.println("Guava Output "+str);

}

}

Output:

Map toString() {Name=John, LastName=Doe, Address=California}

Guava Output Name:John, LastName:Doe, Address:California