



Tables in Guava

Frank Walsh

Guava Reminder

- *“Guava is the open-sourced version of Google's core Java libraries: the core utilities that Googlers use every day in their code”*
 - Can't be bad!
- *“the guava-tests package includes over 286,000 individual test cases”*
 - Well tested!

*"Null sucks" - Doug
Lea*

```
Optional<String> value =  
Optional.fromNullable(str);  
if (value.isPresent()) {  
    // Do stuff with str  
}
```

CharMatcher

```
CharMatcher matcher =
```

```
    CharMatcher.DIGIT.or(inRange('a',  
'z').or(inRange('A', 'Z')));
```

```
if (matcher.matchesAllOf("this1will2match3"))  
{
```

```
    // ...
```

```
}
```

```
String input = " secret passcode ";
```

```
String result =
```

```
CharMatcher.WHITESPACE.trimFrom(input);
```

Multimap

```
Multimap<String, String> mm =  
    ArrayListMultimap.create();
```

```
Collection<String> smiths = mm.get("Smith");  
// empty collection (never null)
```

```
mm.put("Smith", "John");  
mm.put("Smith", "Alice");  
mm.put("Smith", "Diane");
```

```
smiths = mm.get("Smith");  
// [ "John", "Alice", "Diane" ]
```



Table

- `Table<R, C, V>`
 - Think spreadsheet table (row, column, value)

HashBasedTable implements this interface:

```
Table<String, String, Integer>  
iotCoursePlaceTable  
= HashBasedTable.create();
```

Table Example

```
Table<String, String, Integer> iotCoursePlaceTable =  
HashBasedTable.create();
```

```
iotCoursePlaceTable.put("Waterford", "Chemical",  
120);
```

```
iotCoursePlaceTable.put("Waterford", "IT", 60);
```

```
iotCoursePlaceTable.put("Carlow", "Electrical", 60);
```

```
iotCoursePlaceTable.put("Carlow", "IT", 120);
```

```
int seatCountWaterfordIt =  
iotCoursePlaceTable.get("Waterford", "IT");
```

```
Integer seatCountForNoEntry =  
iotCoursePlaceTable.get("Oxford", "IT");
```

```
assertEquals(seatCountWaterfordIt, 60);
```

```
assertNull(seatCountForNoEntry);
```

Table Example

```
Map<String, Integer> itPlaces =  
    iotCoursePlaceTable.column("IT");
```

```
assertEquals(itPlaces.get("Waterford").i  
ntValue(), 60);
```

```
int totalItPlaces = 0;
```

```
for (Integer places : itPlaces.values())  
{  
    totalItPlaces += places;  
}
```


Idea: Movie Ratings

```
Table<Long, Long, Rating>  
movieRatingsTable =  
HashBasedTable.create();
```

```
movieRatingsTable.put(user.id, movie.  
id, rating);
```

```
//map of all ratings for a movie:
```

```
Map<Long, Rating> movieRatings =  
movieRatingsTable.column(movie.id)
```

```
//map of all ratings for a user:
```

```
Map<Long, Rating> userRatings =  
movieRatingsTable.row(user.id)
```