

# Creating and Analyzing Histograms from Streams

---



**José Paumard**

PHD, JAVA CHAMPION, JAVA ROCK STAR

@JosePaumard <https://github.com/JosePaumard>



# Agenda



Let us create maps with collectors

Using `groupingBy()`

And see how to use downstream collectors



# Creating Maps with GroupingBy

---



# A List of Cities with Their State

Houston Texas

Salt Lake City Utah

Dallas Texas

Denver Colorado

Boulder Colorado

Austin Texas

Provo Utah

```
cities.stream()  
    .collect(  
        Collectors.groupingBy(  
            city -> city.getState()  
        )  
    );
```



# A List of Cities with Their State

Houston Texas

Salt Lake City Utah

Dallas Texas

Denver Colorado

Boulder Colorado

Austin Texas

Provo Utah

```
cities.stream()  
    .collect(  
        Collectors.groupingBy(  
            city -> city.getState()  
        )  
    );
```



```
Map<String, List<City>>
```



## The groupingBy collector

- groups data using a function
- by default gather the objects in lists

# A List of Cities with Their State

Houston Texas

Salt Lake City Utah

Dallas Texas

Denver Colorado

Boulder Colorado

Austin Texas

Provo Utah

Texas → Houston, Dallas, Austin

Utah → Salt Lake City, Provo

Colorado → Denver, Boulder



# GroupingBy with a Downstream Collector

---







A downstream collector  
Is a collector that is applied  
To the stream of the values



# A Count of the Cities per State

Texas	→	Houston, Dallas, Austin	<code>Collectors.counting()</code>
Utah	→	Salt Lake City, Provo	<code>{Houston, Dallas, Austin}</code> <code>.stream()</code> <code>.collect(     Collectors.counting() )</code>
Colorado	→	Denver, Boulder	



# A Count of the Cities per State

Texas → Houston, Dallas, Austin

Utah → Salt Lake City, Provo

Colorado → Denver, Boulder



Texas → 3L

Utah → 2L

Colorado → 2L



# Demo



Let us write some code!

Build maps, post-process their values

And extract information out of them



```
Map<City, Long> peopleByCity = ...;

City cityWithTheMostPeople =
    peopleByCity.entrySet().stream()
        .max(Map.Entry.comparingByValue())
        .orElseThrow()
        .getKey();
```

No `map.stream()` method

So to stream a map, the solution is to use its entry set



# Module Wrap Up



What did you learn?

How to create maps with  
the `groupingBy()` collector

Adding downstream collectors

To create complex maps and histograms

And stream the key value pairs

To analyze them



# Course Wrap Up



What did you learn?

The basics of the Stream API

As an implementation of the  
Map / Filter / Reduce pattern

A Stream does not hold any data

A Stream must be connected to a source

Reducing data with reduce and collect

Creating and analyzing maps with  
groupingBy and downstream collectors



# Course Wrap Up



# Thank you!

@JosePaumard

<https://github.com/JosePaumard>

<https://www.youtube.com/user/jpaumard>

