HowToDoInJava

Sorting a Stream by Multiple Fields in Java

Last Updated: March 10, 2022 | By: Lokesh Gupta | Java 8 | Java Compare, Java Sorting, Java Stream Basics

Learn to **sort the streams of objects by multiple fields** using Comparators and `Comparator.thenComparing()` method. This method returns a **lexicographic-order** comparator with another comparator. It gives the same effect as SQL **GROUP BY** clause.

Table Of Contents

1. Creating Comparators for Multiple Fields

To sort on multiple fields, we must first **create simple comparators** for each field on which we want to sort the stream items. Then we **chain these *Comparator* instances** in the desired order to give GROUP BY effect on complete sorting behavior.

Note that *Comparator* provides a few other methods that we can use if they fit in the requirements.

- *thenComparing(keyExtractor)* :
- *thenComparing(comparator)*
- *thenComparing(keyExtractor, comparator)*

- *thenComparingDouble(keyExtractor)*
- *thenComparingInt(keyExtractor)*
- *thenComparingLong(keyExtractor)*

Joining Multiple Comparators

```
//first name comparator
Comparator<Employee> compareByFirstName = Comparator.comparing( Emplo

//last name comparator
Comparator<Employee> compareByLastName = Comparator.comparing( Employ

//Compare by first name and then last name (multiple fields)
Comparator<Employee> compareByFullName = compareByFirstName.thenCompa

//Using Comparator - pseudo code
list.stream().sorted( comparator ).collect();
```

2. Sorting with Complex Comparator

Given below is an example of using `thenComparing()` to create `Comparator` which is capable of sorting the employees' list by their *first name* and *last name*.

Sort by first name and last name

```
import java.util.ArrayList;
import java.util.Comparator;
import java.util.List;
import java.util.stream.Collectors;

public class Main
{
  public static void main(String[] args)
  {
    ArrayList<Employee> employees = getUnsortedEmployeeList();
```

```
    //Compare by first name and then last name
    Comparator<Employee> compareByName = Comparator
                        .comparing(Employee::getFirstName)
                        .thenComparing(Employee::getLastName);

    List<Employee> sortedEmployees = employees.stream()
                    .sorted(compareByName)
                    .collect(Collectors.toList());

    System.out.println(sortedEmployees);
  }

  private static ArrayList<Employee> getUnsortedEmployeeList()
  {
    ArrayList<Employee> list = new ArrayList<>();
    list.add( new Employee(2l, "Lokesh", "Gupta") );
    list.add( new Employee(1l, "Alex", "Gussin") );
    list.add( new Employee(4l, "Brian", "Sux") );
    list.add( new Employee(5l, "Neon", "Piper") );
    list.add( new Employee(3l, "David", "Beckham") );
    list.add( new Employee(7l, "Alex", "Beckham") );
    list.add( new Employee(6l, "Brian", "Suxena") );
        return list;
  }
}
```

Program Output.

Output

```
[E[id=7, firstName=Alex,  lastName=Beckham],
E [id=1, firstName=Alex,  lastName=Gussin],
E [id=4, firstName=Brian,   lastName=Sux],
E [id=6, firstName=Brian,   lastName=Suxena],
E [id=3, firstName=David,   lastName=Beckham],
E [id=2, firstName=Lokesh,  lastName=Gupta],
E [id=5, firstName=Neon,  lastName=Piper]]
```

3. Conclusion

Similar to the chained predicates, we can combine any number of *Comparators* to create any complex sorting logic and sort the *Stream* items with it.

We can use other Comparator methods as well as documented in the official Java docs.

Happy Learning !!

Sourcecode on Github

Recommended Reading:

1. Getting Distinct Stream Items by Comparing Multiple Fields
2. Java Stream reuse – traverse stream multiple times?
3. Gson – Exclude or Ignore Fields
4. Sorting Streams in Java
5. Chaining Multiple Predicates in Java
6. Applying Multiple Conditions on Java Streams
7. Java Stream sorted()
8. Sorting Arrays in Java

9. Java – Sorting Array of Strings in Alphabetical Order

0. Guide to Sorting in Java

Leave a Comment

Name *

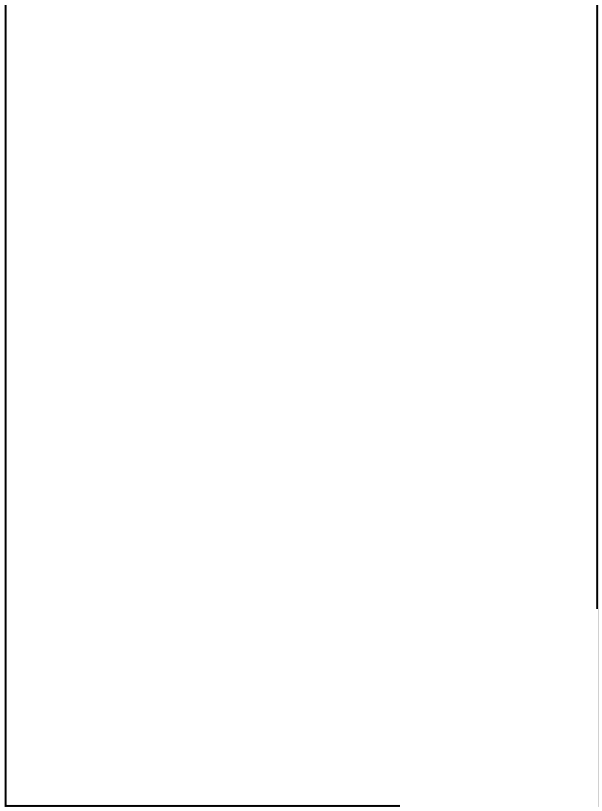
Email *

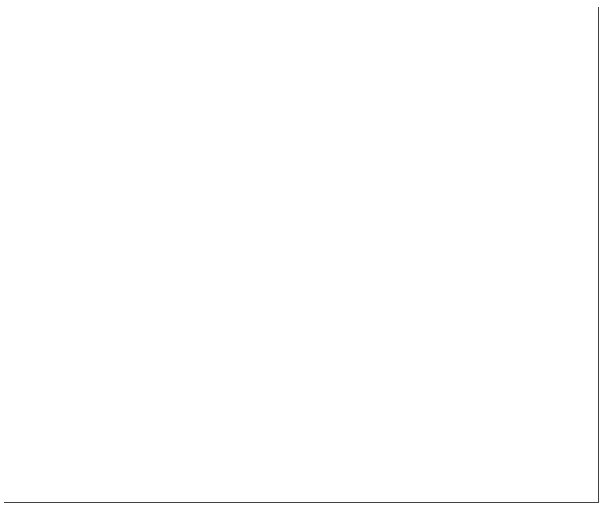
Website

Add me to your newsletter and keep me updated whenever you publish new blog posts

Post Comment

Search ...





HowToDoInJava

A blog about Java and related technologies, the best practices, algorithms, and interview questions.

Meta Links

- About Me
- Contact Us
- Privacy policy

- Advertise
- Guest Posts

Blogs

- REST API Tutorial