Solving Real Time Queries Using Java 8 Features - Employee Management System

BY PRAMODBABLAD (HTTPS://JAVACONCEPTOFTHEDAY.COM/AUTHOR/PRAMODBABLAD/)

JUNE 23, 2019 / JAVA 8 (HTTPS://JAVACONCEPTOFTHEDAY.COM/CATEGORY/JAVA-8/) / 28 COMMENTS (HTTPS://JAVACONCEPTOFTHEDAY.COM/SOLVING-REAL-TIME-QUERIES-USING-JAVA-8-FEATURES-EMPLOYEE-MANAGEMENT-SYSTEM/#COMMENTS)

Let's try to solve some of the real time queries faced in the Employee Management System using Java 8 features.

We will be using following *Employee* class and *employeeList* as example while solving the queries.

1) *Employee* Class :

```
class Employee
{
    int id;

    String name;

    int age;

    String gender;

    String department;

    int yearOfJoining;

    double salary;

    public Employee(int id, String name, int age, String gender, String department, int yearOfJoining, do
    {
        this.id = id;
        this.name = name;
        this.age = age;
        this.gender = gender;
        this.department = department;
        this.yearOfJoining = yearOfJoining;
        this.salary = salary;
    }

    public int getId()
    {
        return id;
    }

    public String getName()
    {
        return name;
    }

    public int getAge()
    {
        return age;
    }

    public String getGender()
    {
        return gender;
    }

    public String getDepartment()
    {
        return department;
    }

    public int getYearOfJoining()
    {
        return yearOfJoining;
    }

    public double getSalary()
    {
        return salary;
    }

    @Override
    public String toString()
    {
        return "Id : "+id
                +", Name : "+name
                +", age : "+age
                +", Gender : "+gender
                +", Department : "+department
                +", Year Of Joining : "+yearOfJoining
                +", Salary : "+salary;
    }
}
```

2) List Of Employees : *employeeList*

```
List<Employee> employeeList = new ArrayList<Employee>();

employeeList.add(new Employee(111, "Jiya Brein", 32, "Female", "HR", 2011, 25000.0));
employeeList.add(new Employee(122, "Paul Niksui", 25, "Male", "Sales And Marketing", 2015, 13500.0));
employeeList.add(new Employee(133, "Martin Theron", 29, "Male", "Infrastructure", 2012, 18000.0));
employeeList.add(new Employee(144, "Murali Gowda", 28, "Male", "Product Development", 2014, 32500.0));
employeeList.add(new Employee(155, "Nima Roy", 27, "Female", "HR", 2013, 22700.0));
employeeList.add(new Employee(166, "Iqbal Hussain", 43, "Male", "Security And Transport", 2016, 10500.0)
employeeList.add(new Employee(177, "Manu Sharma", 35, "Male", "Account And Finance", 2010, 27000.0));
employeeList.add(new Employee(188, "Wang Liu", 31, "Male", "Product Development", 2015, 34500.0));
employeeList.add(new Employee(199, "Amelia Zoe", 24, "Female", "Sales And Marketing", 2016, 11500.0));
employeeList.add(new Employee(200, "Jaden Dough", 38, "Male", "Security And Transport", 2015, 11000.5));
employeeList.add(new Employee(211, "Jasna Kaur", 27, "Female", "Infrastructure", 2014, 15700.0));
employeeList.add(new Employee(222, "Nitin Joshi", 25, "Male", "Product Development", 2016, 28200.0));
employeeList.add(new Employee(233, "Jyothi Reddy", 27, "Female", "Account And Finance", 2013, 21300.0));
employeeList.add(new Employee(244, "Nicolus Den", 24, "Male", "Sales And Marketing", 2017, 10700.5));
employeeList.add(new Employee(255, "Ali Baig", 23, "Male", "Infrastructure", 2018, 12700.0));
employeeList.add(new Employee(266, "Sanvi Pandey", 26, "Female", "Product Development", 2015, 28900.0));
employeeList.add(new Employee(277, "Anuj Chettiar", 31, "Male", "Product Development", 2012, 35700.0));
```

Also Read : Java 8 Lambda Expressions (https://javaconceptoftheday.com/java-8-lambda-expressions/)

3) Real Time Queries On *employeeList*

- How many male and female employees are there in the organization?
- Print the name of all departments in the organization?
- What is the average age of male and female employees?
- Get the details of highest paid employee in the organization?
- Get the names of all employees who have joined after 2015?
- Count the number of employees in each department?
- What is the average salary of each department?
- Get the details of youngest male employee in the product development department?
- Who has the most working experience in the organization?
- How many male and female employees are there in the sales and marketing team?
- What is the average salary of male and female employees?
- List down the names of all employees in each department?
- What is the average salary and total salary of the whole organization?
- Separate the employees who are younger or equal to 25 years from those employees who are older than 25 years?
- Who is the oldest employee in the organization? What is his age and which department he belongs to?

Query 3.1 : How many male and female employees are there in the organization?

For queries such as above where you need to group the input elements, use the *Collectors.groupingBy()* method. In this query, we use *Collectors.groupingBy()* method which takes two arguments. We pass *Employee::getGender* as first argument which groups the input elements based on *gender* and *Collectors.counting()* as second argument which counts the number of entries in each group.

```
Map<String, Long> noOfMaleAndFemaleEmployees=
employeeList.stream().collect(Collectors.groupingBy(Employee::getGender, Collectors.counting()));

System.out.println(noOfMaleAndFemaleEmployees);
```

Output :

{Male=11, Female=6}

Type your email... Subscribe

Interview Questions

60+ Java 8 Interview Questions (https://javaconceptoftheday.com/java-8-interview-questions-and-answers/)

50+ Java Threads Interview Questions (https://javaconceptoftheday.com/java-threads-interview-questions-and-answers/)

30+ Java Exception Handling Interview Questions (https://javaconceptoftheday.com/java-exception-handling-interview-questions-and-answers/)

50+ Java String Interview Questions (https://javaconceptoftheday.com/java-string-interview-questions-and-answers/)

25 Java Array Interview Questions (https://javaconceptoftheday.com/java-array-interview-questions-and-answers/)

Top 25 Simple Basic Java Interview Questions For Freshers (https://javaconceptoftheday.com/top-25-simple-basic-java-interview-questions-for-freshers/)

Java Interview Questions On Nested Classes (https://javaconceptoftheday.com/java-interview-questions-on-nested-classes/)

Java Interview Questions On final Keyword (https://javaconceptoftheday.com/java-interview-questions-on-final-keyword/)

10 Tricky Core Java Interview Coding Questions (https://javaconceptoftheday.com/tricky-core-java-interview-coding-questions/)

Java Interview Questions On Interfaces (https://javaconceptoftheday.com/java-interview-questions-on-interfaces/)

Java Interview Questions On Abstract Class (https://javaconceptoftheday.com/java-interview-questions-on-abstract-class/)

15 Simple But Confusing Java Interview Questions (https://javaconceptoftheday.com/15-simple-but-confusing-java-interview-questions/)

Java Interview Questions On Method Overloading (https://javaconceptoftheday.com/important-java-interview-questions-on-method-overloading/)

Java Interview Questions On Method Overriding (https://javaconceptoftheday.com/java-interview-questions-on-method-overriding/)

Java Interview Questions On Type Casting (https://javaconceptoftheday.com/java-interview-questions-on-type-casting/)

Java Interview Questions On Modifiers (https://javaconceptoftheday.com/java-interview-questions-on-modifiers/)

Java Interview Questions On Inheritance (https://javaconceptoftheday.com/java-interview-questions-on-inheritance/)

Java Interview Questions On Constructors (https://javaconceptoftheday.com/java-interview-questions-on-constructors/)

Java Interview Questions On main() Method (https://javaconceptoftheday.com/java-interview-questions-on-main-method/)

400+ Java Interview Questions (https://javaconceptoftheday.com/java-interview-questions-and-answers/)

Quiz

Java Strings Quiz (https://javaconceptoftheday.com/java-strings-quiz-questions-and-answers/)

++ And -- Quiz (https://javaconceptoftheday.com/quiz-on-increment-and-decrement-operators/)

Java Arrays Quiz (https://javaconceptoftheday.com/10-java-interview-sample-coding-questions-on-arrays/)

Java Enums Quiz (https://javaconceptoftheday.com/java-practice-coding-questions-on-enum-types/)

Nested Classes Quiz (https://javaconceptoftheday.com/java-practice-coding-questions-on-nested-classes/)

Java Modifiers Quiz (https://javaconceptoftheday.com/java-practice-questions-on-access-modifiers/)

Java Interfaces Quiz (https://javaconceptoftheday.com/java-

Query 3.2 : Print the name of all departments in the organization?

Use *distinct()* method after calling *map(Employee::getDepartment)* on the stream. It will return unique departments.

```
employeeList.stream()
            .map(Employee::getDepartment)
            .distinct()
            .forEach(System.out::println);
```

Output :

```
HR
Sales And Marketing
Infrastructure
Product Development
Security And Transport
Account And Finance
```

Query 3.3 : What is the average age of male and female employees?

Use same method as query 3.1 but pass *Collectors.averagingInt(Employee::getAge)* as the second argument to *Collectors.groupingBy()*.

```
Map<String, Double> avgAgeOfMaleAndFemaleEmployees=
employeeList.stream().collect(Collectors.groupingBy(Employee::getGender, Collectors.averagingInt(Employee:

System.out.println(avgAgeOfMaleAndFemaleEmployees);
```

Output :

```
{Male=30.181818181818183, Female=27.166666666666668}
```

Also Read : Java 8 Collectors (https://javaconceptoftheday.com/java-8-collectors-tutorial/)

Query 3.4 : Get the details of highest paid employee in the organization?

Use *Collectors.maxBy()* method which returns maximum element wrapped in an *Optional* object based on supplied *Comparator*.

```
Optional<Employee> highestPaidEmployeeWrapper=
employeeList.stream().collect(Collectors.maxBy(Comparator.comparingDouble(Employee::getSalary)));

Employee highestPaidEmployee = highestPaidEmployeeWrapper.get();

System.out.println("Details Of Highest Paid Employee : ");

System.out.println("================================");

System.out.println("ID : "+highestPaidEmployee.getId());

System.out.println("Name : "+highestPaidEmployee.getName());

System.out.println("Age : "+highestPaidEmployee.getAge());

System.out.println("Gender : "+highestPaidEmployee.getGender());

System.out.println("Department : "+highestPaidEmployee.getDepartment());

System.out.println("Year Of Joining : "+highestPaidEmployee.getYearOfJoining());

System.out.println("Salary : "+highestPaidEmployee.getSalary());
```

Output :

```
Details Of Highest Paid Employee :
================================
ID : 277
Name : Anuj Chettiar
Age : 31
Gender : Male
Department : Product Development
Year Of Joining : 2012
Salary : 35700.0
```

Query 3.5 : Get the names of all employees who have joined after 2015?

For such queries which require filtering of input elements, use *Stream.filter()* method which filters input elements according to supplied *Predicate*.

```
employeeList.stream()
            .filter(e -> e.getYearOfJoining() > 2015)
            .map(Employee::getName)
            .forEach(System.out::println);
```

Output :

```
Iqbal Hussain
Amelia Zoe
Nitin Joshi
Nicolus Den
Ali Baig
```

Query 3.6 : Count the number of employees in each department?

This query is same as query 3.1 but here we are grouping the elements by *department*.

```
Map<String, Long> employeeCountByDepartment=
employeeList.stream().collect(Collectors.groupingBy(Employee::getDepartment, Collectors.counting()));

Set<Entry<String, Long>> entrySet = employeeCountByDepartment.entrySet();

for (Entry<String, Long> entry : entrySet)
{
   System.out.println(entry.getKey()+" : "+entry.getValue());
}
```

Output :

```
Product Development : 5
Security And Transport : 2
Sales And Marketing : 3
Infrastructure : 3
HR : 2
Account And Finance : 2
```

Also Read : Java 8 Streams (https://javaconceptoftheday.com/java-8-streams-beginners-guide/)

Query 3.7 : What is the average salary of each department?

Use the same method as in the above query 3.6, but here pass *Collectors.averagingDouble(Employee::getSalary)* as second argument to *Collectors.groupingBy()* method.

```
Map<String, Double> avgSalaryOfDepartments=
employeeList.stream().collect(Collectors.groupingBy(Employee::getDepartment, Collectors.averagingDouble(Em

Set<Entry<String, Double>> entrySet = avgSalaryOfDepartments.entrySet();

for (Entry<String, Double> entry : entrySet)
{
   System.out.println(entry.getKey()+" : "+entry.getValue());
}
```

Output :

```
Product Development : 31960.0
Security And Transport : 10750.25
Sales And Marketing : 11900.166666666666
Infrastructure : 15466.666666666666
HR : 23850.0
Account And Finance : 24150.0
```

Query 3.8 : Get the details of youngest male employee in the product development department?

For this query, use *Stream.filter()* method to filter male employees in product development department and to find youngest among them, use *Stream.min()* method.

practice-coding-questions-on-interfaces/)

Java Abstract Classes Quiz (https://javaconceptoftheday.com/java-practice-coding-questions-on-abstract-classes/)

Java Polymorphism Quiz (https://javaconceptoftheday.com/java-practice-questions-on-method-overloading-and-overriding/)

Java Inheritance Quiz (https://javaconceptoftheday.com/java-inheritance-practice-coding-questions/)

Java Classes & Objects Quiz (https://javaconceptoftheday.com/java-practice-questions-on-classes-and-objects/)

Tutorials

Java 11 (https://javaconceptoftheday.com/category/java-11/)

Java 10 (https://javaconceptoftheday.com/category/java-10/)

Java 9 (https://javaconceptoftheday.com/category/java-9/)

Java 8 (https://javaconceptoftheday.com/category/java-8/)

Interview Questions (https://javaconceptoftheday.com/java-interview-questions-and-answers/)

Quiz (https://javaconceptoftheday.com/category/java-practice-coding-questions/)

X Vs Y (https://javaconceptoftheday.com/concepts-comparison/)

Java Collections (https://javaconceptoftheday.com/collection-framework-class-hierarchy/)

Java Threads (https://javaconceptoftheday.com/category/threads-java-tutorial/)

Exception Handling (https://javaconceptoftheday.com/category/exception-handling/)

Java Strings (https://javaconceptoftheday.com/category/strings/)

Java Arrays (https://javaconceptoftheday.com/category/arrays/)

Java Generics (https://javaconceptoftheday.com/category/generics/)

JDBC (https://javaconceptoftheday.com/category/jdbc/)

Latest Posts

Tight Coupling And Loose Coupling In Java (https://javaconceptoftheday.com/tight-coupling-and-loose-coupling-in-java/)

Java JDBC Cheat Sheet (https://javaconceptoftheday.com/java-jdbc-cheat-sheet/)

Java Collections Cheat Sheet (https://javaconceptoftheday.com/java-collections-cheat-sheet/)

Java Exception Handling Quiz (https://javaconceptoftheday.com/java-exception-handling-quiz/)

Java Exception Handling Cheat Sheet (https://javaconceptoftheday.com/java-exception-handling-cheat-sheet/)

Java Threads Quiz (https://javaconceptoftheday.com/java-threads-quiz/)

Java I/O Cheat Sheet (https://javaconceptoftheday.com/java-io-cheat-sheet/)

HTTP Status Codes Cheat Sheet (https://javaconceptoftheday.com/http-status-codes-cheat-sheet/)

Java Threads Cheat Sheet (https://javaconceptoftheday.com/java-threads-cheat-sheet/)

Java 17 Sealed Classes And Interfaces – Short Notes (https://javaconceptoftheday.com/java-17-sealed-classes-and-interfaces/)

Most Viewed

Java JDBC Cheat Sheet (https://javaconceptoftheday.com/java-jdbc-cheat-sheet/)

Solving Real Time Queries Using Java 8 Features -Employee Management System (https://javaconceptoftheday.com/solving-

```
Optional<Employee> youngestMaleEmployeeInProductDevelopmentWrapper=
employeeList.stream()
            .filter(e -> e.getGender()=="Male" && e.getDepartment()=="Product Development")
            .min(Comparator.comparingInt(Employee::getAge));

Employee youngestMaleEmployeeInProductDevelopment = youngestMaleEmployeeInProductDevelopmentWrapper.get()

System.out.println("Details Of Youngest Male Employee In Product Development");

System.out.println("-----------------------------------------------");

System.out.println("ID : "+youngestMaleEmployeeInProductDevelopment.getId());

System.out.println("Name : "+youngestMaleEmployeeInProductDevelopment.getName());

System.out.println("Age : "+youngestMaleEmployeeInProductDevelopment.getAge());

System.out.println("Year Of Joinging : "+youngestMaleEmployeeInProductDevelopment.getYearOfJoining());

System.out.println("Salary : "+youngestMaleEmployeeInProductDevelopment.getSalary());
```

Output :

```
Details Of Youngest Male Employee In Product Development :
-----------------------------------------------
ID : 222
Name : Nitin Joshi
Age : 25
Year Of Joinging : 2016
Salary : 28200.0
```

Query 3.9 : Who has the most working experience in the organization?

For this query, sort *employeeList* by *yearOfJoining* in natural order and first employee will have most working experience in the organization. To solve this query, we will be using *sorted()* and *findFirst()* methods of *Stream*.

```
Optional<Employee> seniorMostEmployeeWrapper=
employeeList.stream().sorted(Comparator.comparingInt(Employee::getYearOfJoining)).findFirst();

Employee seniorMostEmployee = seniorMostEmployeeWrapper.get();

System.out.println("Senior Most Employee Details :");

System.out.println("---------------------------");

System.out.println("ID : "+seniorMostEmployee.getId());

System.out.println("Name : "+seniorMostEmployee.getName());

System.out.println("Age : "+seniorMostEmployee.getAge());

System.out.println("Gender : "+seniorMostEmployee.getGender());

System.out.println("Age : "+seniorMostEmployee.getDepartment());

System.out.println("Year Of Joinging : "+seniorMostEmployee.getYearOfJoining());

System.out.println("Salary : "+seniorMostEmployee.getSalary());
```

Output :

```
Senior Most Employee Details :
---------------------------
ID : 177
Name : Manu Sharma
Age : 35
Gender : Male
Age : Account And Finance
Year Of Joinging : 2010
Salary : 27000.0
```

Also Read : Java 8 Optional Class (https://javaconceptoftheday.com/java-8-optional-class/)

Query 3.10 : How many male and female employees are there in the sales and marketing team?

This query is same as query 3.1, but here use *filter()* method to filter sales and marketing employees.

```
Map<String, Long> countMaleFemaleEmployeesInSalesMarketing=
employeeList.stream()
            .filter(e -> e.getDepartment()=="Sales And Marketing")
            .collect(Collectors.groupingBy(Employee::getGender, Collectors.counting()));

System.out.println(countMaleFemaleEmployeesInSalesMarketing);
```

Output :

```
{Female=1, Male=2}
```

Query 3.11 : What is the average salary of male and female employees?

This query is same as query 3.3 where you have found average age of male and female employees. Here, we will be finding average salary of male and female employees.

```
Map<String, Double> avgSalaryOfMaleAndFemaleEmployees=
employeeList.stream().collect(Collectors.groupingBy(Employee::getGender, Collectors.averagingDouble(Employ

System.out.println(avgSalaryOfMaleAndFemaleEmployees);
```

Output :

```
{Male=21300.090909090908, Female=20850.0}
```

Query 3.12 : List down the names of all employees in each department?

For this query, we will be using *Collectors.groupingBy()* method by passing *Employee::getDepartment* as an argument.

```
Map<String, List<Employee>> employeeListByDepartment=
employeeList.stream().collect(Collectors.groupingBy(Employee::getDepartment));

Set<Entry<String, List<Employee>>> entrySet = employeeListByDepartment.entrySet();

for (Entry<String, List<Employee>> entry : entrySet)
{
    System.out.println("---------------------------------------");

    System.out.println("Employees In "+entry.getKey() + " : ");

    System.out.println("---------------------------------------");

    List<Employee> list = entry.getValue();

    for (Employee e : list)
    {
        System.out.println(e.getName());
    }
}
```

Output :

```
_____
Employees In Product Development :
_____
Murali Gowda
Wang Liu
Nitin Joshi
Sanvi Pandey
Anuj Chettiar
_____
Employees In Security And Transport :
_____
Iqbal Hussain
Jaden Dough
_____
Employees In Sales And Marketing :
_____
Paul Niksui
Amelia Zoe
Nicolus Den
_____
Employees In Infrastructure :
_____
Martin Theron
Jasna Kaur
Ali Baig
_____
Employees In HR :
_____
```

Jiya Brein
Nima Roy

Employees In Account And Finance :

Manu Sharma
Jyothi Reddy

Query 3.13 : What is the average salary and total salary of the whole organization?

For this query, we use *Collectors.summarizingDouble()* on *Employee*::*getSalary* which will return statistics of the employee salary like max, min, average and total.

```
DoubleSummaryStatistics employeeSalaryStatistics=
employeeList.stream().collect(Collectors.summarizingDouble(Employee::getSalary));

System.out.println("Average Salary = "+employeeSalaryStatistics.getAverage());

System.out.println("Total Salary = "+employeeSalaryStatistics.getSum());
```

Output :

Average Salary = 21141.235294117647
Total Salary = 359401.0

Query 3.14 : Separate the employees who are younger or equal to 25 years from those employees who are older than 25 years.

For this query, we will be using *Collectors.partitioningBy()* method which separates input elements based on supplied *Predicate*.

```
Map<Boolean, List<Employee>> partitionEmployeesByAge=
employeeList.stream().collect(Collectors.partitioningBy(e -> e.getAge() > 25));

Set<Entry<Boolean, List<Employee>>> entrySet = partitionEmployeesByAge.entrySet();

for (Entry<Boolean, List<Employee>> entry : entrySet)
{
    System.out.println("----------------------------");

    if (entry.getKey())
    {
        System.out.println("Employees older than 25 years :");
    }
    else
    {
        System.out.println("Employees younger than or equal to 25 years :");
    }

    System.out.println("----------------------------");

    List<Employee> list = entry.getValue();

    for (Employee e : list)
    {
        System.out.println(e.getName());
    }
}
```

Output :

Employees younger than or equal to 25 years :

Paul Niksui
Amelia Zoe
Nitin Joshi
Nicolus Den
Ali Baig

Employees older than 25 years :

Jiya Brein
Martin Theron
Murali Gowda
Nima Roy
Iqbal Hussain
Manu Sharma
Wang Liu
Jaden Dough
Jasna Kaur
Jyothi Reddy
Sanvi Pandey
Anuj Chettiar

Query 3.15 : Who is the oldest employee in the organization? What is his age and which department he belongs to?

```
Optional<Employee> oldestEmployeeWrapper = employeeList.stream().max(Comparator.comparingInt(Employee::get

Employee oldestEmployee = oldestEmployeeWrapper.get();

System.out.println("Name : "+oldestEmployee.getName());

System.out.println("Age : "+oldestEmployee.getAge());

System.out.println("Department : "+oldestEmployee.getDepartment());
```

Output :

Name : Iqbal Hussain
Age : 43
Department : Security And Transport

References :

- Java 8 Spliterator (https://javaconceptoftheday.com/java-8-spliterator/)
- Java 8 map() Vs flatMap() (https://javaconceptoftheday.com/differences-between-java-8-map-and-flatmap/)
- Java 8 Stream Intermediate & Terminal Operation (https://javaconceptoftheday.com/java-8-stream-intermediate-and-terminal-operations/)
- Java 8 Collections Vs Streams (https://javaconceptoftheday.com/collections-and-streams-in-java/)
- Java 8 Functional Interfaces (https://javaconceptoftheday.com/java-8-functional-interfaces/)
- Java 8 merge two maps (https://javaconceptoftheday.com/java-8-merge-two-maps-with-same-keys/)
- Java 8 Sort HashMap By Keys (https://javaconceptoftheday.com/java-8-sort-hashmap-by-keys/)
- Java 8 Sort HashMap By Values (https://javaconceptoftheday.com/java-8-sort-hashmap-by-values/)
- Java 8 StringJoiner (https://javaconceptoftheday.com/java-8-stringjoiner-string-join-collectors-joining/)
- Java 8 Method References (https://javaconceptoftheday.com/java-8-method-references/)
- Java 8 Interface changes (https://javaconceptoftheday.com/java-8-interface-changes-default-methods-and-static-methods/)
- Java 8 Oracle Docs (https://docs.oracle.com/javase/8/docs/)

PREVIOUS POST
Java 8 Collectors Tutorial (https://javaconceptoftheday.com/java-8-collectors-tutorial/)

NEXT POST
Java Singleton Design Pattern Implementation With Examples (https://javaconceptoftheday.com/java-singleton-design-pattern-implementation-with-examples/)

Related Posts

60+ Java 8 Interview Questions And Answers (https://javaconceptoftheday.com/java-8-interview-questions-and-answers/)

Differences Between Iterator Vs Spliterator In Java 8 (https://javaconceptoftheday.com/differences-between-iterator-vs-spliterator-in-java-8/)

Java 8 Spliterator (https://javaconceptoftheday.com/java-8-spliterator/)

January 11, 2022

January 3, 2021 /
1 Comment (https://javaconceptoftheday.com/differences-between-iterator-vs-spliterator-in-java-8/#comments)

December 19, 2020 /
1 Comment (https://javaconceptoftheday.com/java-8-spliterator/#comments)

28 Comments

Pankaj
SEPTEMBER 6, 2019 / 4:35 PM (HTTPS://JAVACONCEPTOFTHEDAY.COM/SOLVING-REAL-TIME-QUERIES-USING-JAVA-8-FEATURES-EMPLOYEE-MANAGEMENT-SYSTEM/#COMMENT-4595)
REPLY

Please give same king of real time example with different use cases

Abhishek Kumar
JUNE 14, 2020 / 2:00 AM (HTTPS://JAVACONCEPTOFTHEDAY.COM/SOLVING-REAL-TIME-QUERIES-USING-JAVA-8-FEATURES-EMPLOYEE-MANAGEMENT-SYSTEM/#COMMENT-4843)
REPLY

Query 3.8 line no 3 , &&
please change && with &&

Karthik
AUGUST 29, 2020 / 6:43 PM (HTTPS://JAVACONCEPTOFTHEDAY.COM/SOLVING-REAL-TIME-QUERIES-USING-JAVA-8-FEATURES-EMPLOYEE-MANAGEMENT-SYSTEM/#COMMENT-4890)
REPLY

if i want to find second largest salary in the list what i have to do?

Pankaj Lilhore
SEPTEMBER 18, 2020 / 7:59 PM (HTTPS://JAVACONCEPTOFTHEDAY.COM/SOLVING-REAL-TIME-QUERIES-USING-JAVA-8-FEATURES-EMPLOYEE-MANAGEMENT-SYSTEM/#COMMENT-4900)
REPLY

Optional maxSal= employeeList.stream().max(Comparator.comparingDouble(Employee::getSalary));

Optional sec=employeeList.stream().filter(s -> s.getSalary() !=
maxSal.get().getSalary()).max(Comparator.comparing(Employee::getSalary));
System.out.println("sec sal:"+sec.get());

venkat
SEPTEMBER 24, 2022 / 11:41 PM (HTTPS://JAVACONCEPTOFTHEDAY.COM/SOLVING-REAL-TIME-QUERIES-USING-JAVA-8-FEATURES-EMPLOYEE-MANAGEMENT-SYSTEM/#COMMENT-5490)
REPLY

System.out.println("==Get the details of highest paid employee in the organization?==");
System.out.println(employeeList.stream().
sorted(Comparator.comparing(Employee::getSalary).reversed()).findFirst());
System.out.println("==Get the details of highest paid employee in the organization?==method2");
System.out.println(employeeList.stream().max(Comparator.comparing(Employee::getSalary)));
System.out.println("==Get the details of highest paid employee in the organization?==method3");
System.out.println(Arrays.stream(employeeList.stream().
sorted((o1, o2) -> (int)(o2.getSalary()- o1.getSalary())).limit(1).toArray()).findFirst());
System.out.println("==Get the details of highest paid employee in the organization?==method4");
System.out.println(Arrays.stream(employeeList.stream().
sorted(Comparator.comparing(Employee::getSalary)).skip(employeeList.size()-1).toArray()).findFirst());

Arvind
OCTOBER 26, 2021 / 10:37 PM (HTTPS://JAVACONCEPTOFTHEDAY.COM/SOLVING-REAL-TIME-QUERIES-USING-JAVA-8-FEATURES-EMPLOYEE-MANAGEMENT-SYSTEM/#COMMENT-5245)
REPLY

Optionalemp=emptiest.stream().sorted(comparingDouble(Employee:: getSalary).reverse()).skip(1).findFirst();

harish
JANUARY 9, 2021 / 6:51 PM (HTTPS://JAVACONCEPTOFTHEDAY.COM/SOLVING-REAL-TIME-QUERIES-USING-JAVA-8-FEATURES-EMPLOYEE-MANAGEMENT-SYSTEM/#COMMENT-4993)
REPLY

i need to find 1st three employee details based on highest salary?

ravi
MAY 26, 2021 / 3:13 PM (HTTPS://JAVACONCEPTOFTHEDAY.COM/SOLVING-REAL-TIME-QUERIES-USING-JAVA-8-FEATURES-EMPLOYEE-MANAGEMENT-SYSTEM/#COMMENT-5122)
REPLY

employeeList.stream().sorted(comparingDouble(Employee::getSalary).reversed()).limit(3).forEach(System.out::println);

harish
JANUARY 10, 2021 / 10:35 AM (HTTPS://JAVACONCEPTOFTHEDAY.COM/SOLVING-REAL-TIME-QUERIES-USING-JAVA-8-FEATURES-EMPLOYEE-MANAGEMENT-SYSTEM/#COMMENT-4995)
REPLY

if i have an employee details is in text file how can we apply these examples to text file, please give me these examples by using text files.

Satish Kumar
FEBRUARY 11, 2021 / 8:42 PM (HTTPS://JAVACONCEPTOFTHEDAY.COM/SOLVING-REAL-TIME-QUERIES-USING-JAVA-8-FEATURES-EMPLOYEE-MANAGEMENT-SYSTEM/#COMMENT-5036)
REPLY

Very useful , I was looking for this for a long time.

Shivraj Singh
MAY 24, 2021 / 11:29 AM (HTTPS://JAVACONCEPTOFTHEDAY.COM/SOLVING-REAL-TIME-QUERIES-USING-JAVA-8-FEATURES-EMPLOYEE-MANAGEMENT-SYSTEM/#COMMENT-5120)
REPLY

Thanks Sir Very help full.

Pooja Almiya
JUNE 2, 2021 / 12:52 AM (HTTPS://JAVACONCEPTOFTHEDAY.COM/SOLVING-REAL-TIME-QUERIES-USING-JAVA-8-FEATURES-EMPLOYEE-MANAGEMENT-SYSTEM/#COMMENT-5128)
REPLY

Very useful

neha
JUNE 24, 2021 / 1:41 PM (HTTPS://JAVACONCEPTOFTHEDAY.COM/SOLVING-REAL-TIME-QUERIES-USING-JAVA-8-FEATURES-EMPLOYEE-MANAGEMENT-SYSTEM/#COMMENT-5144)
REPLY

Create simple and small Employee class and write codes to filter all Female Employees working in CSE department from list of Employees by using stream.

Sameer

AUGUST 12, 2021 / 7:18 PM (HTTPS://JAVACONCEPTOFTHEDAY.COM/SOLVING-REAL-TIME-QUERIES-USING-JAVA-8-FEATURES-EMPLOYEE-MANAGEMENT-SYSTEM/#COMMENT-5181)

REPLY

Very Useful. Thanks Alot

Rupendra Raghu

AUGUST 29, 2021 / 12:39 PM (HTTPS://JAVACONCEPTOFTHEDAY.COM/SOLVING-REAL-TIME-QUERIES-USING-JAVA-8-FEATURES-EMPLOYEE-MANAGEMENT-SYSTEM/#COMMENT-5187)

REPLY

As per my understanding for string comparison we should use .equals method not ==.

AJAY LAMKHADE

SEPTEMBER 1, 2021 / 10:37 PM (HTTPS://JAVACONCEPTOFTHEDAY.COM/SOLVING-REAL-TIME-QUERIES-USING-JAVA-8-FEATURES-EMPLOYEE-MANAGEMENT-SYSTEM/#COMMENT-5190)

REPLY

No better example can be imagined than this one for such a topic, very well constructed. thanks a lot for making it simpler.

Amit Kumar

OCTOBER 2, 2021 / 9:26 PM (HTTPS://JAVACONCEPTOFTHEDAY.COM/SOLVING-REAL-TIME-QUERIES-USING-JAVA-8-FEATURES-EMPLOYEE-MANAGEMENT-SYSTEM/#COMMENT-5210)

REPLY

Great Explain No Second Thought

Gajendra Singh

OCTOBER 11, 2021 / 11:45 AM (HTTPS://JAVACONCEPTOFTHEDAY.COM/SOLVING-REAL-TIME-QUERIES-USING-JAVA-8-FEATURES-EMPLOYEE-MANAGEMENT-SYSTEM/#COMMENT-5227)

REPLY

Thank you so much very helpful.

KONDA RAGHU

OCTOBER 12, 2021 / 12:18 AM (HTTPS://JAVACONCEPTOFTHEDAY.COM/SOLVING-REAL-TIME-QUERIES-USING-JAVA-8-FEATURES-EMPLOYEE-MANAGEMENT-SYSTEM/#COMMENT-5232)

REPLY

How to find highest salary in each department

Soumya Mukherjee

OCTOBER 27, 2021 / 8:10 AM (HTTPS://JAVACONCEPTOFTHEDAY.COM/SOLVING-REAL-TIME-QUERIES-USING-JAVA-8-FEATURES-EMPLOYEE-MANAGEMENT-SYSTEM/#COMMENT-5246)

REPLY

```
Map<Integer, Optional> empWithMaxSalaryDeptWise =
employeeList.stream().collect(Collectors.groupingBy(Employee::getDeptId,
Collectors.reducing(BinaryOperator.maxBy(Comparator.comparing(Employee::getSalary)))));

empWithMaxSalaryDeptWise.entrySet().forEach(entry->
System.out.println(entry.getKey()+"————"+entry.getValue().get().getSalary()));
```

Vivek Manhar

NOVEMBER 4, 2021 / 3:41 PM (HTTPS://JAVACONCEPTOFTHEDAY.COM/SOLVING-REAL-TIME-QUERIES-USING-JAVA-8-FEATURES-EMPLOYEE-MANAGEMENT-SYSTEM/#COMMENT-5261)

REPLY

How to write below code
1. List of Student has id, subject and marks need to fatch highest marks in each subject;
2. Map of College Name and StudentName List
filter out all college names starting with Letter "S" and output list of students who belong to that college.
also

Avinash Jayakar

NOVEMBER 17, 2021 / 11:27 AM (HTTPS://JAVACONCEPTOFTHEDAY.COM/SOLVING-REAL-TIME-QUERIES-USING-JAVA-8-FEATURES-EMPLOYEE-MANAGEMENT-SYSTEM/#COMMENT-5271)

REPLY

Exceptional Content .. Thanks a lot !!

---

shankar s.

JANUARY 27, 2022 / 2:27 PM (HTTPS://JAVACONCEPTOFTHEDAY.COM/SOLVING-REAL-TIME-QUERIES-USING-JAVA-8-FEATURES-EMPLOYEE-MANAGEMENT-SYSTEM/#COMMENT-5317)

REPLY

Thanks a lot. very very useful.

Victor

APRIL 5, 2022 / 1:48 AM (HTTPS://JAVACONCEPTOFTHEDAY.COM/SOLVING-REAL-TIME-QUERIES-USING-JAVA-8-FEATURES-EMPLOYEE-MANAGEMENT-SYSTEM/#COMMENT-5384)

REPLY

Hello,
What if you want to return yout key to be name if the departament and your value to be the salary foe the departament ?
I cánd get they key like this but I don't know how to continue to get the salary per position
Map getDeoNameAndSalary(List employeeList){
return employeeLis.stream().map(Employee ::getDepartment).collect(Collectors.groupingby(Function.identity(),// I know here I should take the value but I cannot figure it out :(();

Bj raman

MAY 5, 2022 / 6:10 AM (HTTPS://JAVACONCEPTOFTHEDAY.COM/SOLVING-REAL-TIME-QUERIES-USING-JAVA-8-FEATURES-EMPLOYEE-MANAGEMENT-SYSTEM/#COMMENT-5425)

REPLY

how can we format above averages output into 2 decimal places?

---

manideep

JUNE 20, 2022 / 2:54 AM (HTTPS://JAVACONCEPTOFTHEDAY.COM/SOLVING-REAL-TIME-QUERIES-USING-JAVA-8-FEATURES-EMPLOYEE-MANAGEMENT-SYSTEM/#COMMENT-5443)

REPLY

Very useful

---

ss

JUNE 27, 2022 / 8:33 PM (HTTPS://JAVACONCEPTOFTHEDAY.COM/SOLVING-REAL-TIME-QUERIES-USING-JAVA-8-FEATURES-EMPLOYEE-MANAGEMENT-SYSTEM/#COMMENT-5448)

REPLY

DoubleSummaryStatistics is it created locally??

venkat

SEPTEMBER 24, 2022 / 11:39 PM (HTTPS://JAVACONCEPTOFTHEDAY.COM/SOLVING-REAL-TIME-QUERIES-USING-JAVA-8-FEATURES-EMPLOYEE-MANAGEMENT-SYSTEM/#COMMENT-5489)

REPLY

List down the names of all employees in each department?
System.out.println(employeeList.stream().collect(Collectors.groupingBy(Employee::getDepartment,
Collectors.mapping(Employee::getName,Collectors.toList())
)));

Leave a Reply

Name * **Email *** **Website**

Add Comment

Notify me of follow-up comments by email.

Notify me of new posts by email.

Post Comment

Tutorials :

Java 11 (https://javaconceptoftheday.com/category/java-11/)
Java 10 (https://javaconceptoftheday.com/category/java-10/)
Java 9 (https://javaconceptoftheday.com/category/java-9/)
Java 8 (https://javaconceptoftheday.com/category/java-8/)
Java Collections (https://javaconceptoftheday.com/collection-framework-class-hierarchy/)
Java Threads (https://javaconceptoftheday.com/category/threads-java-tutorial/)
Exception Handling (https://javaconceptoftheday.com/category/exception-handling/)
Java Generics (https://javaconceptoftheday.com/category/generics/)
Java Strings (https://javaconceptoftheday.com/category/strings/)
Java Arrays (https://javaconceptoftheday.com/category/arrays/)
JDBC (https://javaconceptoftheday.com/category/jdbc/)

Quiz :

Java Strings Quiz (https://javaconceptoftheday.com/java-strings-quiz-questions-and-answers/)
++ And -- Quiz (https://javaconceptoftheday.com/quiz-on-increment-and-decrement-operators/)
Java Arrays Quiz (https://javaconceptoftheday.com/10-java-interview-sample-coding-questions-on-arrays/)
Java Enums Quiz (https://javaconceptoftheday.com/java-practice-coding-questions-on-enum-types/)
Nested Classes Quiz (https://javaconceptoftheday.com/java-practice-coding-questions-on-nested-classes/)
Java Modifiers Quiz (https://javaconceptoftheday.com/java-practice-questions-on-access-modifiers/)
Java Interfaces Quiz (https://javaconceptoftheday.com/java-practice-coding-questions-on-interfaces/)
Java Abstract Classes Quiz (https://javaconceptoftheday.com/java-practice-coding-questions-on-abstract-classes/)
Java Polymorphism Quiz (https://javaconceptoftheday.com/java-practice-questions-on-method-overloading-and-overriding/)
Java Inheritance Quiz (https://javaconceptoftheday.com/java-inheritance-practice-coding-questions/)
Java Classes & Objects Quiz (https://javaconceptoftheday.com/java-practice-questions-on-classes-and-objects/)

Interview Q&A :

Java 8 Interview Questions (https://javaconceptoftheday.com/java-8-interview-questions-and-answers/)
Java Threads Interview Questions (https://javaconceptoftheday.com/java-threads-interview-questions-and-answers/)
Exceptions Handling Interview Questions (https://javaconceptoftheday.com/java-exception-handling-interview-questions-and-answers/)
Java Strings Interview Questions (https://javaconceptoftheday.com/java-string-interview-questions-and-answers/)
Java Arrays Interview Questions (https://javaconceptoftheday.com/java-array-interview-questions-and-answers/)
300+ Core Java Interview Questions (https://javaconceptoftheday.com/java-interview-questions-and-answers/)