



**FREE eBook**

# LEARNING java-stream

Free unaffiliated eBook created from  
**Stack Overflow contributors.**

#java-  
stream

# Table of Contents

About.....	1
<b>Chapter 1: Getting started with java-stream.....</b>	<b>2</b>
Remarks.....	2
Examples.....	2
Installation or Setup.....	2
Working with Java Stream Api (Java 8) in Android using Android Studio.....	2
Stream is a new abstract layer introduced in Java 8. A stream is a sequence of elements (o.....	2
In Above code we created Integer List and add some data's then Iterate using for loop. On .....	3
<b>Chapter 2: Java 8 – Convert Map to List.....</b>	<b>4</b>
Introduction.....	4
Examples.....	4
Java 8 – Convert Map to List.....	4
<b>Credits.....</b>	<b>6</b>

---

# About

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [java-stream](#)

It is an unofficial and free java-stream ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official java-stream.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to [info@zzzprojects.com](mailto:info@zzzprojects.com)

---

# Chapter 1: Getting started with java-stream

## Remarks

This section provides an overview of what java-stream is, and why a developer might want to use it.

It should also mention any large subjects within java-stream, and link out to the related topics. Since the Documentation for java-stream is new, you may need to create initial versions of those related topics.

## Examples

### Installation or Setup

Detailed instructions on getting java-stream set up or installed.

### Working with Java Stream Api (Java 8) in Android using Android Studio

#### Gradle Setup :

*build.gradle(Module: app)*

```
compileOptions {
    sourceCompatibility JavaVersion.VERSION_1_8
    targetCompatibility JavaVersion.VERSION_1_8
}

jackOptions {
    enabled true
}
```

#### What is the Stream API ?

**Stream is a new abstract layer introduced in Java 8. A stream is a sequence of elements (objects, primitive types) from the stream source. Therefore, stream is not a data structure and it doesn't store the elements it works with. It can be of a finite or infinite size and allow effortless code parallelism**

#### Advantage's:

- It helps in using data in a declarative way. We can make use of Database functions like Max,

Min etc., without running a full iteration.

- It makes good use of multi-core architectures without worrying about multi-threading code.
- We can create a pipeline of data operations with Java Stream that can run in a sequence or in parallel.
- It provides support for group by, order by etc. operations.
- It supports writing for code in Functional programming style.
- It provides parallel processing of data.

**How streams work :**

**Normal Approach (Without using Stream Api) :**

```
List<Integer> numbers = new ArrayList<>();
numbers.addAll(Arrays.asList(1, 20, 3, 10, 20, 30, 4, 50, 80, 1, 2)); //adding dummy data

int i = 0;
List<String> number_str = new ArrayList<>();
for (Integer num : numbers) {
    if (i >= 5) //after 5 loop will stop
        break;

    if (num >= 10) // check number greater than or equal to 10
    {
        number_str.add(String.format("Number %d", num)); //Typecast Integer to String then
add to String List
        i++; //increment i count
    }
}

number_str.sort(Comparator.naturalOrder()); //sort the list
```

**In Above code we created Integer List and add some data's then Iterate using for loop. On each loop execution we check loop count( $i \geq 5$ ) and  $num \geq 10$ . finally sort the string list.**

**Now, let's rewrite the code using Java's 8 Stream API:**

```
List<Integer> numbers = new ArrayList<>();
numbers.addAll(Arrays.asList(1, 20, 3, 10, 20, 30, 4, 50, 80, 1, 2));

List<String> number_str = numbers.stream()
    .filter(num -> num >= 10) //check num greater than 10
    .limit(5) //stop loop at 5
    .sorted() //sort the list
    .map(num -> String.format("Number %d", num)) //typecast into String List
    .collect(Collectors.toList());
```

Read Getting started with java-stream online: <https://riptutorial.com/java-stream/topic/9339/getting-started-with-java-stream>

---

# Chapter 2: Java 8 – Convert Map to List

## Introduction

Convert Map to List in java 8 using stream api

## Examples

### Java 8 – Convert Map to List

```
public class ConvertMapToList {

    public static void main(String[] args) {

        Map<Integer, String> map = new HashMap<>();
        map.put(10, "apple");
        map.put(20, "orange");
        map.put(30, "banana");
        map.put(40, "watermelon");
        map.put(50, "dragonfruit");

        System.out.println("\n1. Export Map Key as List of Integer method 1");

        List<Integer> methodOneIntegers= map.entrySet().stream()
            .map(x -> x.getKey())
            .collect(Collectors.toList());

        methodOneIntegers.forEach(System.out::println);

        System.out.println("\n1. Export Map Key as List of Integer method 2");

        List<Integer> methodTwoIntegers= map.keySet().stream()
            .collect(Collectors.toList());

        methodOneIntegers.forEach(System.out::println);

        System.out.println("\n2. Export Map Value as List of String method 1");

        List<String> methodOneStrings= map.entrySet().stream()
            .map(x -> x.getValue())
            .collect(Collectors.toList());

        methodOneStrings.forEach(System.out::println);

        System.out.println("\n2. Export Map Value as List of String method 2");

        List<String> methodTwoStrings= map.values().stream()
            .map(x -> x.getValue())
            .collect(Collectors.toList());

        methodTwoStrings.forEach(System.out::println);

    }

}
```

```
}
```

## Output

```
Export Map Key as List of Integer method 1  
50  
20  
40  
10  
30
```

```
Export Map Key as List of Integer method 2  
50  
20  
40  
10  
30
```

```
Export Map Value as List of String method 1  
dragonfruit  
orange  
watermelon  
apple  
banana
```

```
Export Map Value as List of String method 2  
dragonfruit  
orange  
watermelon  
apple  
banana
```

Read Java 8 – Convert Map to List online: <https://riptutorial.com/java-stream/topic/9356/java-8---convert-map-to-list>

---

# Credits

S. No	Chapters	Contributors
1	Getting started with java-stream	<a href="#">Community</a> , <a href="#">Magesh Pandian</a>
2	Java 8 – Convert Map to List	<a href="#">VISHWANATH N P</a>