



EBook Gratis

APRENDIZAJE java-stream

Free unaffiliated eBook created from
Stack Overflow contributors.

#java-
stream

Tabla de contenido

| | |
|---|----------|
| Acerca de..... | 1 |
| Capítulo 1: Empezando con java-stream..... | 2 |
| Observaciones..... | 2 |
| Examples..... | 2 |
| Instalación o configuración..... | 2 |
| Trabajar con Java Stream Api (Java 8) en Android con Android Studio..... | 2 |
| Stream es una nueva capa abstracta introducida en Java 8. Un stream es una secuencia de el..... | 2 |
| En el código anterior creamos la Lista de enteros y agregamos algunos datos, luego Iterate..... | 3 |
| Capítulo 2: Java 8 - Convertir mapa a la lista..... | 4 |
| Introducción..... | 4 |
| Examples..... | 4 |
| Java 8 - Convertir mapa a la lista..... | 4 |
| Creditos..... | 6 |

Acerca de

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

Capítulo 1: Empezando con java-stream

Observaciones

Esta sección proporciona una descripción general de qué es java-stream y por qué un desarrollador puede querer usarlo.

También debe mencionar cualquier tema grande en java-stream y vincular a los temas relacionados. Dado que la Documentación para java-stream es nueva, es posible que deba crear versiones iniciales de esos temas relacionados.

Examples

Instalación o configuración

Instrucciones detalladas sobre cómo configurar o instalar java-stream.

Trabajar con Java Stream Api (Java 8) en Android con Android Studio

Configuración de Gradle:

build.gradle (Módulo: aplicación)

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

jackOptions {
    enabled true
}
```

¿Qué es la API de Stream?

Stream es una nueva capa abstracta introducida en Java 8. Un stream es una secuencia de elementos (objetos, tipos primitivos) del origen del stream. Por lo tanto, el flujo no es una estructura de datos y no almacena los elementos con los que trabaja. Puede ser de un tamaño finito o infinito y permitir paralelismo de código sin esfuerzo

Ventajas de

- Ayuda en el uso de datos de forma declarativa. Podemos utilizar funciones de base de datos

como Max, Min, etc., sin ejecutar una iteración completa.

- Hace un buen uso de arquitecturas de múltiples núcleos sin preocuparse por el código de subprocesos múltiples.
- Podemos crear una tubería de operaciones de datos con Java Stream que puede ejecutarse en una secuencia o en paralelo.
- Proporciona soporte para operaciones grupales, ordenadas, etc.
- Es compatible con la escritura de código en el estilo de programación funcional.
- Proporciona procesamiento paralelo de datos.

Cómo funcionan las corrientes:

Enfoque normal (sin utilizar 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
```

En el código anterior creamos la Lista de enteros y agregamos algunos datos, luego iterate usando for loop. En cada ejecución de bucle verificamos el conteo de bucles ($i \geq 5$) y $num \geq 10$. finalmente ordenar la lista de cadenas.

Ahora, reescribamos el código usando la API 8 Stream de Java:

```
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());
```

Lea Empezando con java-stream en línea: <https://riptutorial.com/es/java-stream/topic/9339/empezando-con-java-stream>

Capítulo 2: Java 8 - Convertir mapa a la lista

Introducción

Convertir Mapa a Lista en java 8 usando la api de flujo

Examples

Java 8 - Convertir mapa a la lista

```
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);

    }

}
```

```
}
```

Salida

```
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
```

Lea Java 8 - Convertir mapa a la lista en línea: <https://riptutorial.com/es/java-stream/topic/9356/java-8---convertir-mapa-a-la-lista>

Creditos

| S. No | Capítulos | Contributors |
|-------|------------------------------------|--|
| 1 | Empezando con java-stream | Community , Magesh Pandian |
| 2 | Java 8 - Convertir mapa a la lista | VISHWANATH N P |