Stream API (java.util.stream)

- Introduced in **Java 8** to process collections in a functional style.
- Supports operations like filtering, mapping, and reducing data.
- Works with Collection objects such as List, Set, and Map.

Functional Interface in Java

A **Functional Interface** in Java is an interface that contains **exactly one abstract method**. It can have **multiple default or static methods**, but only **one abstract method**. Functional interfaces are primarily used for **lambda expressions** and **method references** in Java.

By Eslam Khder


```java
@FunctionalInterface
interface Greeting {
    void sayHello(String name);
}
```

```java
public class Main {
    public static void main(String[] args) {
        Greeting greet = (name) -> System.out.println("Hello, " + name);
        greet.sayHello("Alice"); // Output: Hello, Alice
    }
}
```

By Eslam Khder

1. `Predicate<T>` – Takes an argument and returns a boolean.

```java
Predicate<Integer> isEven = num -> num % 2 == 0;
System.out.println(isEven.test(4)); // true
```

`Function<T, R>` – Takes one argument and returns a result.

```java
Function<String, Integer> lengthFunction = str -> str.length();
System.out.println(lengthFunction.apply("Java")); // 4
```

By Eslam Khder

`Consumer<T>` – Takes an argument and returns nothing.

java

```java
Consumer<String> print = message -> System.out.println(message);
print.accept("Hello World!"); // Output: Hello World!
```

4. `Supplier<T>` – Takes no argument and returns a value.

java

```java
Supplier<Double> randomValue = () -> Math.random();
System.out.println(randomValue.get()); // Random number
```

By Eslam Khder


```java
import java.util.function.BiPredicate;

public class BiPredicateExample {
    public static void main(String[] args) {
        BiPredicate<Integer, Integer> isSumEven = (a, b) -> (a + b) % 2 == 0;

        System.out.println(isSumEven.test(3, 5)); // true (3 + 5 = 8, even)
        System.out.println(isSumEven.test(2, 3)); // false (2 + 3 = 5, odd)
    }
}
```

By Eslam Khder

java Copy Edi

```java
import java.util.function.BiFunction;

public class BiFunctionExample {
    public static void main(String[] args) {
        BiFunction<String, String, String> concat = (s1, s2) -> s1 + " " + s2;

        System.out.println(concat.apply("Hello", "World")); // Output: Hello World
    }
}
```

By Eslam Khder


```java
import java.util.function.BiConsumer;

public class BiConsumerExample {
    public static void main(String[] args) {
        BiConsumer<String, Integer> printPerson = (name, age) ->
            System.out.println(name + " is " + age + " years old.");

        printPerson.accept("Alice", 25); // Output: Alice is 25 years old.
    }
}
```

By Eslam Khder

Interface	Takes	Returns	Example Use
`BiPredicate<T, U>`	2 arguments	`boolean`	Check if sum of two numbers is even
`BiFunction<T, U, R>`	2 arguments	A value (`R`)	Concatenating two strings
`BiConsumer<T, U>`	2 arguments	`void` (No return)	Printing two values together

By Eslam Khder