

JAVA 8 - OPTIONAL CLASS

http://www.tutorialspoint.com/java8/java8_optional_class.htm

Copyright © tutorialspoint.com

Optional is a container object which is used to contain not-null objects. Optional object is used to represent null with absent value. This class has various utility methods to facilitate code to handle values as 'available' or 'not available' instead of checking null values. It is introduced in Java 8 and is similar to what **Optional** is in Guava.

Class Declaration

Following is the declaration for **java.util.Optional<T>** class –

```
public final class Optional<T>
extends Object
```

Class Method

S. No.	Method & Description
1	static <T> Optional<T> empty Returns an empty Optional instance.
2	boolean equalsObjectobj Indicates whether some other object is "equal to" this Optional.
3	Optional<T> filterPredicate < ?super < T > predicate If a value is present and the value matches a given predicate, it returns an Optional describing the value, otherwise returns an empty Optional.
4	<U> Optional<U> flatMapFunction < ?superT, Optional < U >> mapper If a value is present, it applies the provided Optional-bearing mapping function to it, returns that result, otherwise returns an empty Optional.
5	T get If a value is present in this Optional, returns the value, otherwise throws NoSuchElementException.
6	int hashCode Returns the hash code value of the present value, if any, or 0 <i>zero</i> if no value is present.
7	void ifPresentConsumer < ?superT > consumer If a value is present, it invokes the specified consumer with the value, otherwise does nothing.
8	boolean isPresent Returns true if there is a value present, otherwise false.
9	<U>Optional<U> mapFunction < ?superT, ?extendsU > mapper

If a value is present, applies the provided mapping function to it, and if the result is non-null, returns an Optional describing the result.

10 **static <T> Optional<T> ofTvalue**

Returns an Optional with the specified present non-null value.

11 **static <T> Optional<T> ofNullableTvalue**

Returns an Optional describing the specified value, if non-null, otherwise returns an empty Optional.

12 **T orElseTother**

Returns the value if present, otherwise returns other.

13 **T orElseGetSupplier < ? extends T > other**

Returns the value if present, otherwise invokes other and returns the result of that invocation.

14 **<X extends Throwable> T orElseThrowSupplier < ? extends X > exceptionSupplier**

Returns the contained value, if present, otherwise throws an exception to be created by the provided supplier.

15 **String toString**

Returns a non-empty string representation of this Optional suitable for debugging.

Note – This class inherits methods from the **java.lang.Object** class.

Optional Example

To understand how **Optional** is used in practice, let us see the following example. Write the following program, execute and verify result to get more insight of it.

Java8Tester.java

```
import java.util.Optional;

public class Java8Tester {
    public static void main(String args[]){

        Java8Tester java8Tester = new Java8Tester();
        Integer value1 = null;
        Integer value2 = new Integer(10);

        //Optional.ofNullable - allows passed parameter to be null.
        Optional<Integer> a = Optional.ofNullable(value1);

        //Optional.of - throws NullPointerException if passed parameter is null
        Optional<Integer> b = Optional.of(value2);
        System.out.println(java8Tester.sum(a, b));
    }

    public Integer sum(Optional<Integer> a, Optional<Integer> b){

        //Optional.isPresent - checks the value is present or not
```

```
System.out.println("First parameter is present: " + a.isPresent());
System.out.println("Second parameter is present: " + b.isPresent());

//Optional.orElse - returns the value if present otherwise returns
//the default value passed.
Integer value1 = a.orElse(new Integer(0));

//Optional.get - gets the value, value should be present
Integer value2 = b.get();
return value1 + value2;
}
}
```

Verify the Result

Compile the class using **javac** compiler as follows –

```
$javac Java8Tester.java
```

Now run the Java8Tester as follows –

```
$java Java8Tester
```

It should produce the following output –

```
First parameter is present: false
Second parameter is present: true
10
```

```
Loading [MathJax]/jax/output/HTML-CSS/jax.js
```