

GUAVA - OPTIONAL CLASS

http://www.tutorialspoint.com/guava/guava_optional_class.htm

Copyright © tutorialspoint.com

Optional is an immutable object used to contain a not-null object. Optional object is used to represent null with absent value. This class has various utility methods to facilitate the code to handle values as available or not available instead of checking null values.

Class Declaration

Following is the declaration for **com.google.common.base.Optional<T>** class:

```
@GwtCompatible(serializable=true)
public abstract class Optional<T>
    extends Object
    implements Serializable
```

Class Methods

Sr.No	Method & Description
1	static <T> Optional<T> absent Returns an Optional instance with no contained reference.
2	abstract Set<T> asSet Returns an immutable singleton Set whose only element is the contained instance if it is present; an empty immutable Set otherwise.
3	abstract boolean equalsObjectobject Returns true if object is an Optional instance, and either the contained references are equal to each other or both are absent.
4	static <T> Optional<T> fromNullableTnullableReference If nullableReference is non-null, returns an Optional instance containing that reference; otherwise returns absent.
5	abstract T get Returns the contained instance, which must be present.
6	abstract int hashCode Returns a hash code for this instance.
7	abstract boolean isPresent Returns true if this holder contains a <i>non – null</i> instance.
8	static <T> Optional<T> ofTreference Returns an Optional instance containing the given non-null reference.
9	abstract Optional<T> orOptional < ? extendsT > secondChoice

Returns this Optional if it has a value present; secondChoice otherwise.

10 **abstract T orSupplier** < ? extends T > *supplier*

Returns the contained instance if it is present; supplier.get otherwise.

11 **abstract T orTdefaultValue**

Returns the contained instance if it is present; defaultValue otherwise.

12 **abstract T orNull**

Returns the contained instance if it is present; null otherwise.

13 **static <T> Iterable<T> presentInstances**

Iterable < ? extends Optional < ? extends T >> *optionals*

Returns the value of each present instance from the supplied optionals, in order, skipping over occurrences of absent.

14 **abstract String toString**

Returns a string representation for this instance.

15 **abstract <V> Optional<V> transformFunction** < ? super T, V > *function*

If the instance is present, it is transformed with the given Function; otherwise, absent is returned.

Methods Inherited

This class inherits methods from the following class:

- java.lang.Object

Example of Optional class

Create the following java program using any editor of your choice in say **C:/> Guava**.

GuavaTester.java

```
import com.google.common.base.Optional;

public class GuavaTester {
    public static void main(String args[]) {
        GuavaTester guavaTester = new GuavaTester();

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

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

        //Optional.of - throws NullPointerException if passed parameter is null
        Optional<Integer> b = Optional.of(value2);

        System.out.println(guavaTester.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.or - returns the value if present otherwise returns
    //the default value passed.
    Integer value1 = a.or(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:

```
C:\Guava>javac GuavaTester.java
```

Now run the GuavaTester to see the result.

```
C:\Guava>java GuavaTester
```

See the result.

```

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

```

Loading [MathJax]/jax/output/HTML-CSS/fonts/TeX/fontdata.js