# 1 正确认识Optional

Option位于java.util包

A container object which may or may not contain a non-null value. If a value is present, isPresent() will return true and get() will return the value.

Additional methods that depend on the presence or absence of a contained value are provided, such as [orElse()](https://docs.oracle.com/javase/8/docs/api/java/util/Optional.html#orElse-T-) (return a default value if value not present) and [ifPresent()](https://docs.oracle.com/javase/8/docs/api/java/util/Optional.html#ifPresent-java.util.function.Consumer-)(execute a block of code if the value is present).

This is a [value-based](https://docs.oracle.com/javase/8/docs/api/java/lang/doc-files/ValueBased.html) class; use of identity-sensitive operations (including reference equality (==), identity hash code, or synchronization) on instances of Optional may have unpredictable results and should be avoided.

# 2 属性

//静态属性

**private static final** Optional<?> ***EMPTY*** = **new** Optional<>();

//包含的值

*/\*\*  
 \* If non-null, the value; if null, indicates no value is present  
 \*/***private final** T **value**;

# 3构造方法

public static <T> [Optional](https://docs.oracle.com/javase/8/docs/api/java/util/Optional.html)<T> empty()

Returns an empty Optional instance. No value is present for this Optional.

**API Note:**

Though it may be tempting to do so, avoid testing if an object is empty by comparing with == against instances returned by Option.empty(). There is no guarantee that it is a singleton. Instead, use [isPresent()](https://docs.oracle.com/javase/8/docs/api/java/util/Optional.html#isPresent--).

**Type Parameters:**

T - Type of the non-existent value

**Returns:**

an empty Optional

public static <T> [Optional](https://docs.oracle.com/javase/8/docs/api/java/util/Optional.html)<T> of(T value)

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

**Type Parameters:**

T - the class of the value

**Parameters:**

value - the value to be present, which must be non-null

**Returns:**

an Optional with the value present

**Throws:**

[NullPointerException](https://docs.oracle.com/javase/8/docs/api/java/lang/NullPointerException.html) - if value is null

public static <T> [Optional](https://docs.oracle.com/javase/8/docs/api/java/util/Optional.html)<T> ofNullable(T value)

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

**Type Parameters:**

T - the class of the value

**Parameters:**

value - the possibly-null value to describe

**Returns:**

an Optional with a present value if the specified value is non-null, otherwise an empty Optional

# 4 常用方法

## 4.1 ifPresent

避免这种写法

Object object = new Object() ;

if(object ! = null){

执行相关操作

}

public void ifPresent(Consumer<? super T> consumer) {  
 if (value != null)  
 consumer.accept(value);  
}

使用方法：

If a value is present, invoke the specified consumer with the value, otherwise do nothing.

**Parameters:**

consumer - block to be executed if a value is present

**Throws:**

[NullPointerException](https://docs.oracle.com/javase/8/docs/api/java/lang/NullPointerException.html) - if value is present and consumer is null

## 4.2 orElse

 假定实例 Optional<User> user, 应避免 if(user.isPresent()) { ... } else { ... } 几中应用方式.

orElseGet

orElseThrow

orElse

## 4.3 filter

## 4.3 map

由对象[Optional](https://docs.oracle.com/javase/8/docs/api/java/util/Optional.html)<T>产生一个新对象[Optional](https://docs.oracle.com/javase/8/docs/api/java/util/Optional.html)<U>

当 user.isPresent() 为真, 获得它关联的 orders, 为假则返回一个空集合时,无法使用 orElse, orElseGet 等方法, 可以使用 map 函数

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | return user.map(u -> u.getOrders()).orElse(Collections.emptyList())    //上面避免了我们类似 Java 8 之前的做法  if(user.isPresent()) {    return user.get().getOrders();  } else {    return Collections.emptyList();  } |

map  是可能无限级联的, 比如再深一层, 获得用户名的大写形式

|  |  |
| --- | --- |
| 1  2  3 | return user.map(u -> u.getUsername())             .map(name -> name.toUpperCase())             .orElse(null); |

这要搁在以前, 每一级调用的展开都需要放一个 null 值的判断

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | User user = .....  if(user != null) {    String name = user.getUsername();    if(name != null) {      return name.toUpperCase();    } else {      return null;    }  } else {    return null;  } |

## 4.4 flatMap

public <U> [Optional](https://docs.oracle.com/javase/8/docs/api/java/util/Optional.html)<U> flatMap([Function](https://docs.oracle.com/javase/8/docs/api/java/util/function/Function.html)<? super [T](https://docs.oracle.com/javase/8/docs/api/java/util/Optional.html),[Optional](https://docs.oracle.com/javase/8/docs/api/java/util/Optional.html)<U>> mapper)

# 5 总结

使用 Optional 时尽量不直接调用 Optional.get() 方法, Optional.isPresent() 更应该被视为一个私有方法, 应依赖于其他像 Optional.orElse(), Optional.orElseGet(), Optional.map() 等这样的方法.