Optional class isPresent() and get() method

In previous posts we saw how to [avoid Null Pointer Exception](http://data-structure-learning.blogspot.com/2015/07/avoid-null-pointer-using-optional-class.html) using Optional<T> class, [Optional<T> class introduction](http://data-structure-learning.blogspot.com/2015/07/optional-class-introduction.html), Optional<T> class [of(), ofNullable() and empty()](http://data-structure-learning.blogspot.com/2015/07/optional-class-of-ofnullable-empty.html) method, [Optional<T> class ifPresent() method](http://data-structure-learning.blogspot.com/2015/07/optional-class-ifpresent-method.html) and Optional<T> class filter() method.

In this post we will see isPresent() and get() method of Optional<T> class.

First we will cover isPresent() method.

|  |
| --- |
| **public** **boolean** isPresent() {  **return** value != **null**;  }  This method is used to check whether the Optional<T> instance has a value in it or not.  If value is not null then it returns true.  If value is null then it returns false. |

Let us take the same example that we are using for Optional<T> class’s methods explanation. We will write a method that will pick an element that starts with some prefix String.

**public** **static** **void** find(**final** List<String> days, **final** String prefix){

**final** Optional<String> found=days.stream()

.filter(day -> day.startsWith(prefix))

.findFirst();

**if**(found.isPresent()){

//doSomething()

}

}

Now let us see get() method of Optional<T> class.

|  |
| --- |
| **public** T get() {  **if** (value == **null**) {  **throw** **new** NoSuchElementException("No value present");  }  **return** value;  }  get() method is used to return the value present in this Optional<T> object.  If value is null then it throws NoSuchElementException. Remember Optional instance may or may not contain non-null value. Read this [post for more](http://data-structure-learning.blogspot.com/2015/07/optional-class-introduction.html) information.  If value is present it is returned. |

Let us use get() method in the above code.

**public** **static** **void** find(**final** List<String> days, **final** String prefix){

**final** Optional<String> found = days.stream()

.filter(day -> day.startsWith(prefix))

.findFirst();

**if**(**found.isPresent**()){

System.***out***.println(**found.get()**);// Prints Wednesday

}

}

That’s all on isPresent() and get() method.