are
ure
veet
)

Java 8 Tutorials

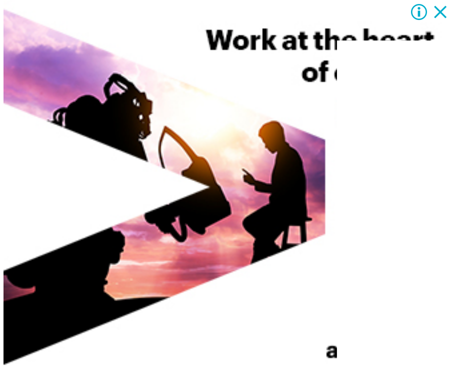
- java 8 - lambda expression
- java 8 - stream
- java 8 - deafult methods
- java 8 - compute
- java 8 - map reduce
- java 8 - string join
- java 8 - date time
- java 8 - flatmap
- java 8 - foreach
- java 8 - map function
- java 8 - books
- java 8 - filter
- java 8 - stream peek
- java 8 - comparator
- java 8 - LocalDate
- java 8 - sorting map
- java 8 - list to map
- java - date formatting
- java 8 - multiple inheritance
- java 8 - compare string
- java 8 - stringjoiner
- java 8 - tutorials

Blog Archive

► 2022 (332)
▼ 2021 (725)
► December (71)
► November (74)
► October (42)
► September (48)
► August (81)
► July (283)
▼ June (104)
How to format numbers in Java? - NumberFormat Example
What is fail safe and fail fast Iterator in Java?
How to convert ByteBuffer to String in Java [Example]

Disclosure: This article may contain affiliate links. When you purchase, we may earn a commission.

Java 8 Optional isPresent(), OrElse() and get() Examples



The `Optional` class in Java is one of many goodies we have got from the Java 8 release. If you use it correctly, Optional can result in **clean code** and can also help you to avoid `NullPointerException` which has bothered Java developers from its inception. Even though many of us have used null to indicate the absence of something, the big problem is that if you call a method or access a field on the null object (except static fields), you will get a `NullPointerException` and your whole program may crash.

The bigger problem is to find the faulty code or root cause because `NullPointerException` only indicates the line when you try to access a field or variable from a null object but how does that `null` is get created on the code is unknown.

I have seen many developers spending nights finding the source of the dreaded `NullPointerException`. Thankfully some of those problems can be resolved by **correctly using the Optional class.**

The biggest benefit of using the `Optional` class is that it improves the readability of code. Now, you don't need to guess whether a value is present or not. Optional itself indicates that a value *may be* present.

You just need to take the literal meaning of Optional that value may or may not be present and you have to code accordingly. This will think you what to do in case of value is not present like using a default value or throwing a proper error message.

On a related note, Java is moving really fast and we are already in Java 12, still, a lot of developers have to learn Java 8, particularly the functional programming aspect. If you think that your Java 8 skills are not at par or you want to improve yourself, I suggest you join a comprehensive Java course like **The Complete Java MasterClass**. It covers everything you need to know and also recently updated for Java 11.

How to use Optional in Java 8

Optional is nothing but a container, you can also view it as a Stream of one or none values. Now, let's see some examples of `Optional` class to write clean code in Java.

Spring Inte

- Spring B
- Spring C
- Spring M
- Microser
- 10 Sprin
- Spring B
- Spring F
- Hibernat Answers
- System I
- Java De Questior
- 40 Core with Ans
- 10 Frequ Interview

are
re
veet
)

How to convert long to String in Java? Example
How to append text to existing File in Java? Example
How to add element at first and last position of l...
How to convert String to double in Java with Example
How to get first and last elements form ArrayList ...
Difference between Abstraction and Polymorphism in...
Difference between synchronized ArrayList and Copy...
4 ways to concatenate Strings in Java [Example and...
java.lang.NoClassDefFoundError: org/apache/xmlbean...
How to use PriorityQueue in Java? An Example
2 Ways to Read a Text File in Java? BufferredReade...
How to Synchronize HashMap in Java with Example
How to parse String to Date in Java using JodaTime...
Right way to Compare String Objects in Java [Example]
How to Convert Byte array to String in Java with E...
2 Ways to Print Custom String Value of Java Enum
How to break from a nested

Suppose, we need to write a method that takes a letter and returns the first book starting with that letter in the library. If there is no book found for that letter we return `null`.

```
package tool;

import java.util.Arrays;
import java.util.List;

/**
 * Program to demonstrate how to use Optional in Java 8
 *
 * @author WINDOWS 8
 *
 */
public class Hello {
  private static List<String> listOfBooks = Arrays.asList("Effective Java",
      "Clean Code", "Test Driven");

  /*
   * Return the first book start with a letter.
   */
  public static String getBook(String letter) {
    String found = null;
    for (String book : listOfBooks) {
      if (book.startsWith(letter)) {
        found = book;
        break;
      }
    }

    return found != null ? found : "Book not Found";
  }
}
```

You can replace the null check in this method using `Optional.get()` and `Optional.isPresent()` method and imperative code with the new Java 8 functional code using Stream API.

If you are not familiar with Stream API then check the Javadoc of Java SE 8 or higher version, and if you find Javadoc boring then you can also check out **From Collections to Streams in Java 8 Using Lambda Expressions** course on Pluralsight. It's one of the best courses on Collection and Stream and I highly recommend all Java developers to take a look at it.

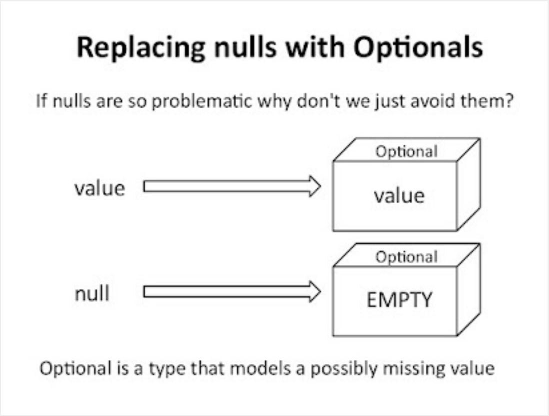
Subscribe f
Your ema

Interview Q

- core jav
- SQL int
- data str
- coding i
- java col
- java des questio
- thread i
- hiberna
- j2ee inte
- Spring I
- object o questio

are
ıre
veet
)

loop in Java? [Example]
- Difference between instance and Object in Java
- How To Convert Map to List in Java with Example
- How to Convert a Double to Long in Java - Example ...
- How to reverse ArrayList in Java with Example
- How to Read, Write XLSX File in Java - Apach POI E...
- How to create User Defined Exception class in Java
- How to Sort HashMap in Java based on Keys and Values
- How to convert String to Float in Java and vice-v...
- How to Generate Random Number between 1 to 10 - Ja...
- Second Highest Salary in MySQL and SQL Server - Le...
- How to Make Executable JAR file in Eclipse IDE - Java
- How to calculate Sum of Digits using Recursion in ...
- How Constructor Works in Java? [Answer]
- 3 Examples to Read FileInputStream as String in Ja...
- How to use Static Method in Java with Example
- How to add Zeros at the Beginning of a Number in J...
- How to Synchronize


Replacing nulls with Optionals
If nulls are so problematic why don't we just avoid them?
value
Optional
value
null
Optional
EMPTY
Optional is a type that models a possibly missing value

It's an excellent course but you would need a Pluralsight membership to access this course, which costs around $29 per month or $299 per year. This may seem a bit expensive at first but it's well worth it because it provides access to more than 5000 online courses on Pluralsight which you can use to learn any latest technology.

I have bought Pluralsight membership and I also suggest you join it if you can invest that much money in a year for your learning, and they are also offering $100 off now, just check.

Anyway, even if you don't have Pluralsight membership, you can still access this course for free by signing up for the **10-day Free Pass** which provides 200 minutes of access to all of their courses.

Modified Java Program using Optional Object

Anyway here is our modified Java program using the Optional feature of Java 8:

```
package tool;

import java.util.Arrays;
import java.util.List;
import java.util.Optional;

/**
 * Program to demonstrate how to use Optional in Java 8
 *
 * @author WINDOWS 8
 *
 */
public class Hello {
  private static List<String> listOfBooks = Arrays.asList("Effective Java",
      "Clean Code", "Test Driven");

  /*
   * Return the first book start with a letter.
   */
  public static String getBook(String letter) {
```

Recommen
- 5 Free R Beginne
- 5 Free C and Spri
- 10 Free and Exp
- 10 Fram Should L
- 10 Book Read
- 10 Open Framew
- 10 Progr
- 10 Book Read
- 5 Great I
- 5 Free D Courses
- 10 Free Algorith

Books and I
- Best Boo Beginne
- 5 Books Core Spi
- 2 books develope

ArrayList in Java with Example

2 Ways to sort HashMap in Java? Example

5 Examples of Formatting Float or Double Numbers ...

Difference between StringBuilder and StringBuffer ...

3 Examples to Loop Map in Java - Foreach vs Iterator

Why Abstract class is Important in Java? [Example]

Difference between String literal and New String o...

What is the difference between a Class and an Obje...

What is the Actual Use of interface in Java?

How to use Modulo , Modulus, or Remainder Operator...

Difference between Polymorphism and Inheritance in...

Array length vs ArrayList Size in Java [Example]

5 Difference between Iterator and ListIterator in ...

How to Print Array with elements in Java? [Solutio...

HelloWorld Program in Java with Example

Difference between GenericServlet vs HttpServlet i...

How to Remove Duplicates

```
    Optional<String> book
          = listOfBooks.stream()
                       .filter(b -> b.startsWith(letter))
                       .findFirst();

    return book.isPresent() ? book.get() : "Book Not Found";
  }
}
```

What we have done here is instead of using String to represent a book, we have used `Optional<String>` which means the variable may or may not contain a book. This is the most important benefit of using Optional, it forces developers to code accordingly.

We have also used the new Stream API to convert our imperative style code to the new Java 8 functional style. Since Stream API returns `Optional` it makes it easy to explain the example.

Anyway, if you look closely, the new check is not better than the old null check. If you compare, the old one was even shorter and cleaner. **So, we are not really using `Optional` in the right way**.

That's actually the main problem for Java developers who have not understood the `Optional` concept fully. Instead of using `ifPresent()` and `get()`, it's better to use `OrElse()` method which can return a default value if Optional doesn't contain a value. Using plain `get()` is not an option because it either returns value or null.

Anyway, by using the `orElse()` method we can rewrite the above code as shown below, which is much more cleaner and expressive.

```
/*
   * Return the first book start with a letter.
   */
  public static String getBook(String letter) {
    Optional<String> book
          = listOfBooks.stream()
                       .filter(b -> b.startsWith(letter))
                       .findFirst();

    return book.orElse("Book Not Found");
  }
```

You can see that the `OrElse()` method is much nicer than the combination of `isPresent()` and `get()`. It not only removes the check but also more expressive like either return a book or else a specified String.

The Optional API is also rich and provides a couple of useful methods like `ifPresent()` and `orElseThrow()` which can be used to throw an Exception if the value is not present. You can further see **The Complete Java MasterClass** on Udemy to learn about Java API and the modern way of Java Coding.

- 12 Advai Books fc
- 5 Free Ja
- 5 Books
- Books E Read
- Top 10 C
- 10 Free
- 5 Books
- Books E Read
- Top 10 C

Followers

Followers (

Follow

Privacy

- Privacy
- Terms a

Recommen

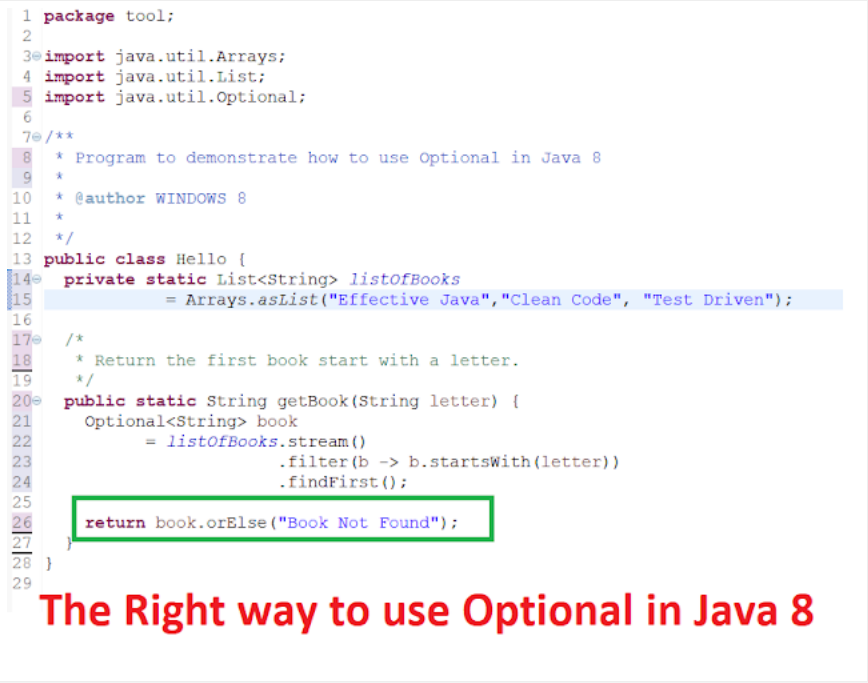
How to F java.lang

are
ıre
veet
)

from ArrayList in Java [E...
10 AWT Swing Interview Questions Answers in Java
How to convert ArrayList to String in Java - Sprin...
How to iterate over HashSet in Java - loop or trav...
Difference between RuntimeException and checked Ex...
Write a Program to Find Sum of Digits in Java
Java 7 Improved Exception Handling - Multiple Catc...
How to format Date in Java - SimpleDateFormat Example
Difference between URL, URI and URN - Interview Qu...
How to traverse iterate or loop ArrayList in Java
2 Ways to Remove Elements/Objects From ArrayList i...
How to use Java Enum in Switch Case Statement - Ex...
Difference between Array vs ArrayList in Java
What is CopyOnWriteArrayList in Java - Example Tu...
Difference between Error vs Exception in Java - In...
Java program to get SubList from ArrayList - Example
Difference between HashMap and ConcurrentHashMap i...

Java Program Using Optional Features of Java 8

And, here is the screenshot of a complete program to use Optional in Java 8 for your reference:



```
package tool;

import java.util.Arrays;
import java.util.List;
import java.util.Optional;

/**
 * Program to demonstrate how to use Optional in Java 8
 *
 * @author WINDOWS 8
 *
 */
public class Hello {
  private static List<String> listOfBooks
          = Arrays.asList("Effective Java","Clean Code", "Test Driven");

  /*
   * Return the first book start with a letter.
   */
  public static String getBook(String letter) {
    Optional<String> book
        = listOfBooks.stream()
                     .filter(b -> b.startsWith(letter))
                     .findFirst();

    return book.orElse("Book Not Found");
  }
}
```

That's all about **how to use the Optional class in Java 8**. Even though it's a poor cousin of Scala's `Option` type and Haskel's `Maybe` type, you can still write cleaner code and avoid `NullPointerException` if used correctly. Don't just use it blindly otherwise you would end up with poor code, sometimes worse than plain null checks.

Related Java 8 Tutorials

oracle.jd
[Solved]
3 ways t
convertir
using Ja
How to F
Java - A
Top 15 S
Questior
How to c
dimensic
Example
Top 15 J
Program
Answers
Top 10 F
Books - I
Online
How to C
LocalDat
Tutorial
Java 8 C
and get(
Top 21 S
Coding I
Solution

Copyright a

Sharing is a
contents of
original pict
property of
more than v
from the blo
to the origir
about it. Se
comment.

What Java developer Should Know about Object and j...

Difference between Serializable vs Externalizable ...

Difference between transient vs volatile variable ...

20 EJB 3.0 Interview Questions and Answers - Java ...

Autoboxing, Enum, Generics, Varargs methods - Java...

JDOM Example : Reading and Parsing XML with SAX pa...

What is Struts Action Class in Java J2EE - How to use

Difference between List and Set in Java [Answered]

How to Override Equals, HashCode and CompareTo met...

Difference between Comparator and Comparable in Ja...

How to Iterate over Java Enum : values() Example

How to Split String in Java using Regular Expression

Why String Class is made Immutable or Final in Jav...

Difference between Deep and Shallow Copy in Java O...

How HashSet works in Java [Explained]

How to Create File and Directory in

are
ıre
veet
)

If you are interested in learning more about the new features of Java 8, here are my earlier articles covering some of the important concepts of Java 8:

- How to join String in Java 8 (example)
- How to use filter() method in Java 8 (tutorial)
- Java 8 - Stream.collect() Example (example)
- Java 8 Stream.findFirst() + filter() example (see)
- How to convert List to Map in Java 8 (solution)
- How to join String in Java 8 (example)
- Difference between abstract class and interface in Java 8? (answer)
- 10 Free Courses for Experienced Java Programmers (courses)
- How to use peek() method in Java 8 (example)
- How to use Stream class in Java 8 (tutorial)
- How to use forEach() method in Java 8 (example)
- What is the double-colon operator of Java 8? (tutorial)
- 20 Examples of Date and Time in Java 8 (tutorial)
- 5 Books to Learn Java 8 from Scratch (books)
- How to use peek() method in Java 8 (example)
- How to sort the may by values in Java 8? (example)
- How to format/parse the date with LocalDateTime in Java 8? (tutorial)
- 5 Free Courses to learn Java 8 and 9 (courses)

Thanks for reading this article so far. If you like this tutorial then please share it with your friends and colleagues. If you have any questions or feedback then please drop a note.

P.S.: If you want to learn more about new features in Java 8 then please see the **Java Programming, Lambda, and more (Java 13, 12, 11, 10, 9,8)**, a free course on Udemy. It explains all the important features of Java 8 like lambda expressions, streams, functional interface, Optional, new Date Time API, and other miscellaneous changes from recent Java releases.

- Java with Example
- Best Way to Compare Two Strings in Java Alphabetic...
- How to use ArrayList in Java with Examples
- Reading Input and Password in Java using java.io.C...
- How to replace characters on String in Java
- When to use ArrayList vs LinkedList in Java? [Answ...
- Java Regular Expression to Check If String contain...
- How to Convert java.util.Date to java.sql.Date - E...
- java.lang.UnsatisfiedLinkError : Library not found ...
- Can we Override Private Method in Java? Inner Class?
- How to Iterate over Array in Java 1.5 using foreac...
- How to Fix java.lang.OufOfMemoryError: Direct Buff...
- Difference between JDK and JRE in Java Platform
- Can abstract class have Constructor in Java - Inte...
- How to use Iterator Java - Example Tutorial
- Difference between this and super keywords in Java

► May (3)

► April (3)

► March (7)

► January (9)

► 2020 (2)

Posted by javin paul

Labels: core java, Java 8

4 comments:

Santhosh February 1, 2021 at 11:19 AM

The orElse() example is not working!

Reply

Replies

javin paul February 4, 2021 at 8:40 PM

Hello Santhosh, what error are you getting?

Ambayi Abhijith Namboothiri June 30, 2021 at 11:59 AM

It worked fine for me thanks

javin paul July 1, 2021 at 9:33 PM

Thx for confirmation Abhijit. Much appreciated.

Reply

Enter Comment

Feel free to comment, ask questions if you have any doubt.

Newer Post Home Older Post

Subscribe to: Post Comments (Atom)

are

ıre

veet

)

Subscribe for New Posts on Email

Get new posts by email:

Enter your email

Subscribe