

java.lang.NullPointerException - Common cause of NullPointerException in Java Example

java.lang.NullPointerException or NullPointerException in Java is probably the first Exception you will face in Java. It is true nightmare for beginners in Java but pretty easy to solve once you get familiar with Exception handling in Java. What makes NullPointerException little tricky is its name which has pointer in itself and Java does [not support](#) pointers like [multiple inheritance in Java](#) . In this article we will see What is NullPointerException in Java, [How to solve](#) Exception in thread "main" java.lang.NullPointerException, finding possible cause of Java NullPointerException and how to troubleshoot NPE in Java. Based on my experience once you know little bit about NullPointerException its pretty easy to solve. By the way, as said, "prevention is better than cure", you can [avoid != null check and NullPointerException](#) by following some [best practices](#).

What is NullPointerException in Java

□ NullPointerException in Java is nothing but an error or precisely an exception which occurs if we tried [to perform](#) any operation on object which is null. In Java reference variable points to object created in heap but when you create a reference variable of type object by default its point to "null" and when you try to call any method on null, try to access any variable on null you will get this null pointer exception. no matter by what reason your reference variable is pointing to null be it not initialized or set to null if you perform any operation it will throw [Exception in thread main : java.lang.NullPointerException](#)

When does NullPointerException occurs in Java

Javadoc of java.lang.NullPointerException has outlined scenario when it could be occurred:

- 1) When you call instance method on a null object. you won't get null pointer exception if you call static method or class method on null object because static method doesn't require an instance to call any method.
- 2) While accessing or changing any variable or field on null object.
- 3) Throwing null when an Exception is expected to throw.
- 4) When calling length of array when array is null.
- 5) Accessing or changing slots of null just like an array.
- 6) When you try to synchronize on null object or using null inside synchronized block in Java

we will see examples of NullPointerException for each of above scenario to get it right and understand it better.

Common cause of NullPointerException in Java as Example

Based upon my experience java.lang.NullPointerException repeats itself on various format, I have collected most common cause of java.lang.NullPointerException in java code and explained them here, we will use following Trade class for example :

```
public class Trade {  
    private String symbol;  
    private int price;  
    public static String market;  
  
    public Trade(String symbol, int price) {  
        this.symbol = symbol;  
        this.price = price;  
    }  
  
    public int getPrice() {  
        return price;  
    }  
  
    public void setPrice(int price) {  
        this.price = price;  
    }  
  
    public String getSymbol() {  
        return symbol;  
    }  
  
    public void setSymbol(String symbol) {  
        this.symbol = symbol;  
    }  
}
```

1) Java NullPointerException while calling instance method on null object

This is probably the most common case of this error, you call method on some object and found that reference is null, always perform null check if you see possibility of null before calling any method on object.

```
Trade pennyStock = null;  
pennyStock.getPrice(); //this will throw NullPointerException
```

```
Exception in thread "main" java.lang.NullPointerException  
at test.NullPointerExceptionTest.main(NullPointerExceptionTest.java:23)
```

2) NullPointerException in Java while accessing field on null reference.

```
Trade fxtrade = null;
int price = fxtrade.price; //here fxtrade is null, you can't access field here
```

```
Exception in thread "main" java.lang.NullPointerException
at test.NullPointerExceptionTest.main(NullPointerExceptionTest.java:64)
```

3) java.lang.NullPointerException when throwing null as exception.

If you throw an Exception object and if that is null you will get null pointer exception as shown in below example

```
RuntimeException nullException = null;
throw nullException;
```

```
Exception in thread "main" java.lang.NullPointerException
at test.NullPointerExceptionTest.main(NullPointerExceptionTest.java:74)
```

4) example of NullPointerException when getting length of an array which is null.

```
Trade[] bluechips = null;
int length = bluechips.length; //array is null here
```

```
Exception in thread "main" java.lang.NullPointerException
at test.NullPointerExceptionTest.main(NullPointerExceptionTest.java:85)
```

5) Example of NPE when accessing element of a null array.

```
Trade[] bluechips = null;
Trade motorola = bluechips[0]; //array is null here
```

```
Exception in thread "main" java.lang.NullPointerException
at test.NullPointerExceptionTest.main(NullPointerExceptionTest.java:94)
```

6) You will also get NullPointerException in Java if you try to synchronize on null object or try to use null object inside synchronized block in Java.

```
Trade highbetaTrade = null;
synchronized(highbetaTrade){
    System.out.print("This statement is synchronized on null");
}
```

```
Exception in thread "main" java.lang.NullPointerException
at test.NullPointerExceptionTest.main(NullPointerExceptionTest.java:104)
```

How to solve NullPointerException in Java

To solve a NullPointerException in Java first we need to find cause, which is very easy just look the stack-trace of NullPointerException and it will show exact line number where NPE has occurred. now go to that line and look for possible object operation like *accessing field*, calling method or throwing exception etc, that will give you an idea which object is null. Now once you found that which object is null job is half done, now find out why that object is null and solve the java.lang.NullPointerException. This second part always vary sometime you get null object [from factory](#) or sometime some other thread might have set it null, though using [Assertion](#) in early phase of development you can minimize chances of java.lang.NullPointerException but as I said its little bit related to environment and can come on production even if tested fine in test environment. Its best to avoid NullPointerException by applying careful or defensive coding technique and null safe API methods.

When in Java Code NullPointerException doesn't come

1) When you access any static method or static variable with null reference.

If you are dealing with static variables or static method than you won't get null pointer exception even if you have your reference variable pointing to null because static variables and method call are bonded during [compile time](#) based on class name and not associated with object. for example below code will run fine and not throw NullPointerException because "market" is an [static variable](#) inside Trade Class.

```
Trade lowBetaTrade = null;
String market = lowBetaTrade.market; //no NullPointerException market is static variable
```

Important points on NullPointerException in Java

- 1) NullPointerException is an [unchecked exception](#) because its extends RuntimeException and it doesn't mandate try catch block to handle it.
- 2) When you get NullPointerException look at line number to find out which object is null, it may be object which is calling any method.
- 3) Modern IDE like Netbeans and Eclipse gives you hyper link of line where NullPointerException occurs
- 4) You can set an [Exception break point in Eclipse](#) to suspend execution when NullPointerException occurs read 10 tips on java debugging in Eclipse [more details](#).
- 5) Don't forget to see name of [Thread](#) on which NullPointerException occurs. in multi-threading NPE can be little tricky if some random thread is setting reference to null.
- 6) Its best to avoid NullPointerException while coding by following some coding best practices or putting null check on database as constraint.

That's all on **What is java.lang.NullPointerException**, When it comes and how to solve it. In next part of this tutorial we will look on some best java coding practices to avoid NullPointerException in Java.

