

java 中常见的 NullPointerException 异常

下面列出了我们学习 java 时常出现的 NullPointerException 异常，以及它的解决办法

1, 空指针错误 java.lang.NullPointerException 使用基本的 JAVA 数据类型，变量的值要么已经是默认值，如果没有对其正常赋值，程序便不能通过编译，因此使用基本的 JAVA 数据类型

(double,float,boolean,char,int,long)一般不会引起空指针异常。由此可见，空指针异常主要跟与对象的操作相关。

下面先列出了可能发生空指针异常的几种情况及相应解决方案：

不管对象是否为空就直接开始使用。

(JSP)代码段 1: `out.println(request.getParameter("username"));`

描述：代码段 1 的功能十分简单，就是输出用户输入的表域"username"的值。

说明：看上去，上面的语句找不出什么语法错误，而且在大多数情况下也遇不到什么问题。但是，如果某个用户在输入数据时并没有提供表单域"username"的值，或通过某种途径绕过表单直接输入时，此时 `request.getParameter("username")`的值为空（不是空字符串，是空对象 null。），`out` 对象的 `println` 方法是无法直接对空对象操作，因此代

码段 1 所在的 JSP 页面将会抛出"java.lang.NullPointerException"异常。即使对象可能为空时，也调用 java.lang.Object 或 Object 对象本身的一些方法如 toString(), equals(Object obj)等操作。

```
(JSP)代码段 2: String userName =  
request.getParameter("username");  
If (userName.equals("root")) {...}
```

描述： 代码段 2 的功能是检测用户提供的用户名，如果是用户名称为"root"的用户时，就执行一些特别的操作。

说明： 在代码段 2 中，如果有用户没有提供表单域"username"的值时，字符串对象 userName 为 null 值，不能够将一个 null 的对象与另一个对象直接比较，同样，代码段 2 所在的 JSP 页面就会抛出 (java.lang.NullPointerException)空指针错误。

```
(JSP)代码段 3: String userName =  
session.getAttribute("session.username").toString();
```

描述： 代码段 3 的功能是将 session 中 session.username 的值取出，并将该值赋给字符串对象 userName。说明： 在一般情况下，如果在用户已经进行某个会话，则不会出现什么问题；但是，如果此时应用服务器重新启动，而用户还没有重新登录，（也可能是用户关闭浏览器，但是仍打开原来的页面。）那么，此时该 session 的值就会失效，同时

导致 session 中的 session.username 的值为空。对一个为 null 的对象的直接执行 toString()操作，就会导致系统抛出 (java.lang.NullPointerException)空指针异常。

解决方案： 为了确保进行操作或引用的对象非空，假若我们要对某对象进行操作或引用，我们首先去检查该对象是否已经实例化且不为空；并且在系统中加入针对对象为空时情况的处理。 如：采用 String 对象保存用户提交的结果；在如果涉及对象的操作时，先检测其是否为空后，检查到对象为空后，可再选择进行以下任一种处理方式：

处理方式 1) 检查到对象为空时，设置对象值为空字符串或一个默认值；

处理方式 2) 检测到对象为空时，根本不执行某操作，直接跳转到其他处理中。

处理方式 3) 检查到对象为空时，提示用户操作有错误。 将代码段 2 按以上方式进行改写，得到：

```
方式 1: String userName = request.getParameter("username"); // 该  
变量值为空时，转化为默认空字符串          If (userName == null)  
userName = ""; If (userName.equals("root")) {.....}
```

```
方式 2: String userName = request.getParameter("username"); // 该  
变量值为空时，转化为默认空字符串，不执行有关操作。 If  
(usreName != null) { If (userName.equals("root")) {.....} }
```

```
方式 3: String userName = request.getParameter("username"); // 该  
变量值为空时，转化为默认空字符串，不执行有关操作。 If (usreName
```

```
== null) { // 提示用户输入信息为空 }
```

实际中，上面提供到三种处理方式也同样适用于其他异常的处理：

异常处理方式 1) 检测到异常出现，设置对象值为空字符串或一个默认值；

异常处理方式 2) 检测到异常出现，根本不执行某操作，直接跳转到其他处理中。

异常处理方式 3) 检测到异常出现，提示用户操作有错误。