

java资源获取分析计算机等级考试 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/645/2021_2022_java_E8_B5_84_E6_BA_90_c97_645444.htm

在开发java程序的过程中，我们经常要做的一件事就是获取资源。那么什么是资源呢？说白了，在计算机里那就是一堆数据。只是这堆数据对我们的java程序有多种表现形式，一般来说有File，URL，InputStream等等。而单就文件这一项就有很多种：配置文件，java类文件，jps文件，图片、css、js文件等等。面对这林林总总的资源，我们在设计一个读取资源的接口时，就需要针对不同形式的资源提供方法，这样就导致我们的接口还是与实际的资源形式绑定在一起，未能完全的抽象。另外，在java程序中资源的存放位置也是各异的。有的存放在classpath中，有的存放在文件系统中，有的存放在web应用中。而对于不同位置的资源，java程序获取这些资源的方法各有不同。

A、获取classpath中的资源：

```
Java代码 URL url =
```

```
this.getClass().getResource("resource_name"). URL url =  
this.getClass().getClassLoader().getResource("resource_name").
```

```
URL url =
```

```
Thread.currentThread().getContextClassLoader().getResource("resource_name").
```

甜性涩爱下载

<http://www.bt285.cn/content.php?id=1196863> 那么在jdk中为什么又提供了三种方式来获取classpath下的资源呢？这其中是有些来头的。第一行代码中是利用Class类的实例来获取，第二行代码是使用加载当前类的classloader来获取。看下jdk中的源代码会发现class类的实例最后还是委托加载他的classloader来获

取资源的。Java代码 `public java.net.URL getResource(String name) { name = resolveName(name). ClassLoader cl = getClassLoader0(). if (cl==null) { // A system class. return ClassLoader.getSystemResource(name). } return cl.getResource(name). }` 从上面的代码中可以看出，对于资源的加载并没有像类加载所采用的双亲委托机制。而是当前类的classloader不为null的情况下先从当前类的classloader中加载资源。而只有当前类的classloader为null的时候才从system classloader中去加载资源。这样可以方便我们自定义配置类覆盖一些默认配置。当然，j2se应用中如果没有特别定制classloader时，我们自己写的类都是被system classloader加载的。到底利用class去获取资源和利用classloader去获取资源有什么区别呢？区别就在 `resolveName(name)` 这个方法中。两种方式对于资源名称的表示方式不同。下面是一个简单的包结构，/表示类路径的根 / |-com.cn.test |-Test.class |-test2.txt |-test1.txt

Java代码 // 获取与当前类在同一个包下的资源 URL `url1 = this.getClass().getResource("test2.txt").` // 获取com.cn.test包下的资源,需加/ URL `url2 = this.getClass().getResource("/com/cn/test/test2.txt").` // 获取类路径根下的资源 URL `url3 = this.getClass().getClassLoader().getResource("test1.txt").` // 获取包com.cn.test包下的资源 URL `url4 = this.getClass().getResource("com/cn/test/test2.txt").` 而第三利用当前线程的contextClassLoader来获取资源的解释可以参考我的另一篇 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com