

ring文件资源操作和Web相关工具类盘点 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/271/2021\\_2022\\_ring\\_E6\\_96\\_87\\_E4\\_BB\\_B6\\_c104\\_271655.htm](https://www.100test.com/kao_ti2020/271/2021_2022_ring_E6_96_87_E4_BB_B6_c104_271655.htm) Spring 不但提供了一个功能全面的应用开发框架，本身还拥有众多可以在程序编写时直接使用的工具类，您不但可以在 Spring 应用中使用这些工具类，也可以在其它的应用中使用，这些工具类中的大部分是可以在脱离 Spring 框架时使用的。了解 Spring 中有哪些好用的工具类并在程序编写时适当使用，将有助于提高开发效率、增强代码质量。在这个分为两部分的文章中，我们将从众多的 Spring 工具类中遴选出那些好用的工具类介绍给大家。第 1 部分将介绍与文件资源操作和 Web 相关的工具类。文件资源操作 文件资源的操作是应用程序中常见的功能，如当上传一个文件后将其保存在特定目录下，从指定地址加载一个配置文件等等。我们一般使用 JDK 的 I/O 处理类完成这些操作，但对于一般的应用程序来说，JDK 的这些操作类所提供的方法过于底层，直接使用它们进行文件操作不但程序编写复杂而且容易产生错误。相比于 JDK 的 File，Spring 的 Resource 接口（资源概念的描述接口）抽象层面更高且涵盖面更广，Spring 提供了许多方便易用的资源操作工具类，它们大大降低资源操作的复杂度，同时具有更强的普适性。这些工具类不依赖于 Spring 容器，这意味着您可以在程序中象一般普通类一样使用它们。加载文件资源 Spring 定义了一个 `org.springframework.core.io.Resource` 接口，Resource 接口是为了统一各种类型不同的资源而定义的，Spring 提供了若干 Resource 接口的实现类，这些实现类可以轻松地加载不同类

型的底层资源，并提供了获取文件名、URL 地址以及资源内容的操作方法。访问文件资源 假设有一个文件地位于 Web 应用的类路径下，您可以通过以下方式对这个文件资源进行访问：通过 `FileSystemResource` 以文件系统绝对路径的方式进行访问；通过 `ClassPathResource` 以类路径的方式进行访问；通过 `ServletContextResource` 以相对于 Web 应用根目录的方式进行访问。相比于通过 JDK 的 `File` 类访问文件资源的方式，Spring 的 `Resource` 实现类无疑提供了更加灵活的操作方式，您可以根据情况选择适合的 `Resource` 实现类访问资源。下面，我们分别通过 `FileSystemResource` 和 `ClassPathResource` 访问同一个文件资源：

```
清单 1. FileSourceExamplepackage
com.baobaotao.io. import java.io.IOException. import
java.io.InputStream. import
org.springframework.core.io.ClassPathResource. import
org.springframework.core.io.FileSystemResource. import
org.springframework.core.io.Resource. public class
FileSourceExample {public static void main(String[] args) {try
{String filePath =
"D:/masterSpring/chapter23/webapp/WEB-INF/classes/conf/file1.txt". // 使用系统文件路径方式加载文件Resource res1 = new
FileSystemResource(filePath). // 使用类路径方式加载文件
Resource res2 = new ClassPathResource("conf/file1.txt").
InputStream ins1 = res1.getInputStream(). InputStream ins2 =
res2.getInputStream(). System.out.println("res1:"
res1.getFilename()). System.out.println("res2:" res2.getFilename()).
} catch (IOException e) {e.printStackTrace(). }}} 在获取资源后，
```

您就可以通过 Resource 接口定义的多个方法访问文件的数据和其它的信息：如您可以通过 `getFileName()` 获取文件名，通过 `getFile()` 获取资源对应的 File 对象，通过 `getInputStream()` 直接获取文件的输入流。此外，您还可以通过 `createRelative(String relativePath)` 在资源相对地址上创建新的资源。在 Web 应用中，您还可以通过 `ServletContextResource` 以相对于 Web 应用根目录的方式访问文件资源，如下所示：

```
< %@ page language="java" contentType="text/html.  
charset=utf-8" pageEncoding="utf-8"% > < jsp:directive.page  
import="org.springframework.web.context.support.ServletContext  
Resource"/ > < jsp:directive.page  
import="org.springframework.core.io.Resource"/ > < %//    注意  
文件资源地址以相对于 Web 应用根路径的方式表示Resource  
res3 = new ServletContextResource(application,  
"/WEB-INF/classes/conf/file1.txt"). out.print(res3.getFilename()).  
% > 对于位于远程服务器（Web 服务器或 FTP 服务器）的文件  
资源，您则可以方便地通过 UriResource 进行访问。为了方便  
访问不同类型的资源，您必须使用相应的 Resource 实现类，  
是否可以在不显式使用 Resource 实现类的情况下，仅根据  
带特殊前缀的资源地址直接加载文件资源呢？Spring 提供了  
一个 ResourceUtils 工具类，它支持“classpath:”和“file:”的  
地址前缀，它能够从指定的地址加载文件资源，请看下面的  
例子：清单 2. ResourceUtilsExamplepackage com.baobaotao.io.  
import java.io.File. import org.springframework.util.ResourceUtils.  
public class ResourceUtilsExample {public static void main(String[]  
args) throws Throwable{File clsFile =
```

```
ResourceUtils.getFile("classpath:conf/file1.txt").  
System.out.println(clsFile.isFile()). String httpFilePath =  
"file:D:/masterSpring/chapter23/src/conf/file1.txt". File httpFile =  
ResourceUtils.getFile(httpFilePath).  
System.out.println(httpFile.isFile()). } } ResourceUtils 的  
getFile(String resourceLocation) 方法支持带特殊前缀的资源地址，  
这样，我们就可以在不和 Resource 实现类打交道的情况下使用 Spring  
文件资源加载的功能了。 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
```