

JavaSE6新特性:HTTP增强 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/252/2021_2022_JavaSE6_E6_96_B0_c104_252406.htm 概述 Java 语言从诞生的那天起，就非常注重网络编程方面的应用。随着互联网应用的飞速发展，Java 的基础类库也不断地对网络相关的 API 进行加强和扩展。在 Java SE 6 当中，围绕着 HTTP 协议出现了很多实用的新特性：NTLM 认证提供了一种 Window 平台下较为安全的认证机制；JDK 当中提供了一个轻量级的 HTTP 服务器；提供了较为完善的 HTTP Cookie 管理功能；更为实用的 NetworkInterface；DNS 域名的国际化支持等等。NTLM 认证不可避免，网络中有很多资源是被安全域保护起来的。访问这些资源需要对用户的身份进行认证。下面是一个简单的例子：

```
import java.net.*;import java.io.*;public class Test { public static void main(String[] args) throws Exception { URL url = new URL("http://PROTECTED.com"); URLConnection connection = url.openConnection(); InputStream in = connection.getInputStream(); byte[] data = new byte[1024]; while(in.read(data)>0) { //do something for data } in.close(); }
```

当 Java 程序试图从一个要求认证的网站读取信息的时候，也就是说，从联系于 `http://Protected.com` 这个 `URLConnection` 的 `InputStream` 中 `read` 数据时，会引发 `FileNotFoundException`。尽管笔者认为，这个 `Exception` 的类型与实际错误发生的原因实在是相去甚远；但这个错误确实是由网络认证失败所导致的。要解决这个问题，有两种方法：其一，是给 `URLConnection` 设定一个“Authentication”属性：`String credit`

```
= USERNAME ":" PASSWORD.String encoding = new  
sun.misc.BASE64Encoder().encode  
(credit.getBytes()).connection.setRequestProperty ("Authorization",  
"Basic " encoding). 这里假设 http://PROTECTED.COM 使用了  
基本 ( Basic ) 认证类型。 从上面的例子，我们可以看出，设  
定 Authentication 属性还是比较复杂的：用户必须了解认证方  
式的细节，才能将用户名/密码以一定的规范给出，然后用特  
定的编码方式加以编码。 Java 类库有没有提供一个封装了认  
证细节，只需要给出用户名/密码的工具呢？这就是我们要介  
绍的另一种方法，使用 java.net.Authentication 类。 每当遇到  
网站需要认证的时候，HttpURLConnection 都会向  
Authentication 类询问用户名和密码。 Authentication 类不会知  
道究竟用户应该使用哪个 username/password 那么用户如何向  
Authentication 类提供自己的用户名和密码呢？提供一个继承于  
Authentication 的类，实现 getPasswordAuthentication 方法，在  
PasswordAuthentication 中给出用户名和密码：class  
DefaultAuthenticator extends Authenticator { public  
PasswordAuthentication getPasswordAuthentication () { return new  
PasswordAuthentication ("USER", "PASSWORD".toCharArray()).  
}} 然后，将它设为默认的 ( 全局  
) Authentication:Authenticator.setDefault (new  
DefaultAuthenticator()). 那么，不同的网站需要不同的用户名/  
密码又怎么办呢？ Authentication 提供了关于认证发起者的足够  
多的信息，让继承类根据这些信息进行判断，在  
getPasswordAuthentication 方法中给出了不同的认证信息  
： getRequestingHost() getRequestingPort() getRequestingPrompt()
```

getRequestingProtocol() getRequestingScheme()
getRequestingURL() getRequestingSite() getRequestorType() 另一
件关于 Authentication 的重要问题是认证类型。不同的认证类
型需要 Authentication 执行不同的协议。至 Java SE 6.0 为止
， Authentication 支持的认证方式有： HTTP Basic authentication
HTTP Digest authentication NTLM Http SPNEGO Negotiate
Kerberos NTLM 这里我们着重介绍 NTLM。 NTLM 是 NT
LAN Manager 的缩写。早期的 SMB 协议在网络上明文传输口
令，这是很不安全的。微软随后提出了 WindowsNT 挑战/响
应验证机制，即 NTLM。 NTLM 协议是这样的：客户端首先
将用户的密码加密成为密码散列；客户端向服务器发送自己的
用户名，这个用户名是用明文直接传输的；服务器产生一个
16 位的随机数字发送给客户端，作为一个 challenge（挑战
）；客户端用步骤1得到的密码散列来加密这个 challenge，
然后把这个返回给服务器；服务器把用户名、给客户端的
challenge、客户端返回的 response 这三个东西，发送域控制
器；域控制器用这个用户名在 SAM 密码管理库中找到这个
用户的密码散列，然后使用这个密码散列来加密 challenge；
域控制器比较两次加密的 challenge，如果一样，那么认证成功；
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