http://blog.csdn.net/chenlycly/article/details/51820727

1.只有字母和数字[0-9a-zA-Z]、一些特殊符号“$-\_.+!\*'(),”[不包括双引号]、以及某些保留字，才可不经过编码直接用于URL。

2.如果URL中有汉字，就必须编码后使用。麻烦的是，RFC 1738没有规定具体的编码方法，而是交给应用程序（浏览器）自己决定。这导致“URL编码”成为了一个混乱的领域。在使用url进行参数传递时，经常会传递一些中文名（或含有特殊字符）的参数或URL地址，在后台处理时会发生转换错误。这些特殊符号在URL中是不能直 接传递的，如果要在URL中传递这些特殊符号，那么就要使用他们的编码了。编码的格式为：%加字符的ASCII码，即一个百分号%，后面跟对应字符的 ASCII（16进制）码值。

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 特殊字符 | 含义 | 十六进制值 |
| 1. | + | URL 中+号表示空格 | %2B |
| 2. | 空格 | URL中的空格可以用+号或者编码 | %20 |
| 3. | / | 分隔目录和子目录 | %2F |
| 4. | ? | 分隔实际的 URL 和参数 | %3F |
| 5. | % | 指定特殊字符 | %25 |
| 6. | # | 表示书签 | %23 |
| 7. | & | URL 中指定的参数间的分隔符 | %26 |
| 8. | = | URL 中指定参数的值 | %3D |

3例子:

URLDecoder.decode 特殊字符处理

% 在URL中是特殊字符，需要特殊转义一下：

解决办法：使用%25替换字符串中的%号，使用%2B替换+

url = url.replaceAll("%(?![0-9a-fA-F]{2})", "%25");

url = url.replaceAll("\\+", "%2B");

String urlStr = URLDecoder.decode(url, "UTF-8");