

Content-disposition（内容-部署）是MIME协议类型的扩展，MIME协议指示MIME用户代理如何显示附加的文件。

Content-disposition 是 MIME 协议的扩展，MIME 协议指示 MIME 用户代理如何显示附加的文件。Content-disposition 其实可以控制用户请求所得的内容存为一个文件的时候提供一个默认的文件名，文件直接在浏览器上显示或者在访问时弹出文件下载对话框。

#### 格式说明：

content-disposition = "Content-Disposition" ":" disposition-type \*( ";" disposition-param )

#### 字段说明：

Content-Disposition为属性名

disposition-type是以什么方式下载，如attachment为以附件方式下载

disposition-param为默认保存时的文件名

服务端向客户端浏览器发送文件时，如果是浏览器支持的文件类型，一般会默认使用浏览器打开，比如txt、jpg等，会直接在浏览器中显示，如果需要提示用户保存，就要利用Content-Disposition进行一下处理，关键在于一定要加上attachment：

代码如下：

[复制代码](#)

```
Response.AppendHeader("Content-Disposition","attachment;filename=FileName.txt");
```

备注：这样浏览器会提示保存还是打开，即使选择打开，也会使用相关联的程序比如记事本打开，而不是IE直接打开了。

Content-Disposition就是当用户想把请求所得的内容存为一个文件的时候提供一个默认的文件名。具体的定义如下：

## 1. Content-Disposition的作用

Content-disposition是MIME协议的扩展，MIME协议指示MIME用户代理如何显示附加的文件。当Internet Explorer接收到头时，他会激活文件下载对话框，它的文件名框自动填充headers指定的文件名。

服务器向浏览器发送文件时，如果是浏览器支持的文件类型，一般会默认使用浏览器打开，比如txt、jpg等。如果需要提示用户保存，就要利用**Content-Disposition**进行处理，（敲黑板，划重点）关键在于一定要加上**attachment** [附件] [ə'tætʃmənt]。

#### 例如

```
Response.AppendHeader("ContentDisposition","attachment;filename=FileName.txt");
```

这样的话，浏览器在打开的时候会提示保存还是打开，即使选择打开，也会使用相关联的程序，比如记事本打开，而不是IE直接打开。

Content-Disposition就是当用户想把请求所得的内容存为一个文件的时候提供一个默认的文件名。

具体的定义如下：

```
1 //content-disposition的定义
2 content-disposition = "Content-Disposition" ":"
3     disposition-type
4     *("; " disposition-param)
5 //disposition-type的定义
6 disposition-type = "attachment" | disp-extension-token
7 //disposition-param的定义
8 disposition-param = filename-param | disp-extension-param
9 //filename-param的定义
10 filename-param = "filename" "=" quoted-string
11 //disp-extension-token的定义
12 disp-extension-token = token
13 //disp-extension-param
14 token "=" ( token | quoted-string )
15
```

例如：

Content-Disposition:attachment;filename="filename.xlsx;"

注意点：

当然filename参数可以包含路径信息，但User-Agent会忽略这些信息，只会把路径信息的最后一部分作为文件名。当响应类型为application/octet-stream情况下使用上面的头信息的话，那么就不能直接显示内容，而是弹出一个“文件下载”的对话框，接下来就是用户决定“打开”还是“保存”了。

## 2. 下载文件中文乱码

Content-Disposition如何适配各个浏览器以及解决中文乱码问题。

### IE浏览器下载乱码问题

总体下来就是这么几点：

两个IE11在使用第一个方法下载文件时中文文件名都会乱码，而使用第二个方法下载时其中一个IE11中文不会乱码，另一个IE11则会乱码；文件名中存在空格时两个IE11浏览器下载下来文件文件名空格会变成+号，其他浏览器没有这个问题；火狐浏览器下载时遇到文件名中有空格时下载下来的文件的文件名第一个空格后面的文字都会丢失。

1. \*\*老版IE的USER-AGENT参数中含有MISE关键字，但是IE11之后浏览器的USER-AGENT中去掉了MISE关键字。【IE11：Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like

Gecko; 】【 \*\*

2. IE下载文件名存在+号，是因为URLEncoder函数在对字符串进行转码后将空格替换成了+号，IE直接把+号显示出来。解决办法就是对文件名进行转码之后，使用replace方法将+号替换成%20即可，浏览器会将%20转换成空格输出。
3. 对于第三个问题则是因为代码在set响应头时Content-Disposition参数的attachment;filename=等号后面文件名字符串没有使用**双引号**括起来，火狐浏览器对于遇到文件名有空格时认为空格前的字符是一个完整的字符串。故下载下来文件时文件名就只剩空格前的那几个字了。解决办法就是：在filename两边加上双引号并加反斜杠转义。

## 编码类

```
1 public static String toUtf8String(String fileName, HttpServletRequest request) throws
   Exception {
2     final String userAgent = request.getHeader("USER-AGENT");
3     String finalFileName = null;
4     if (StringUtils.contains(userAgent, "MSIE") || StringUtils.contains(userAgent,
        "Trident")) { // IE浏览器（旧版/新版）
5         finalFileName = URLEncoder.encode(fileName, "UTF8");
6     } else if (StringUtils.contains(userAgent, "Mozilla")) { // google, 火狐浏览器
7         finalFileName = new String(fileName.getBytes(), "ISO8859-1");
8     } else {
9         finalFileName = URLEncoder.encode(fileName, "UTF8"); // 其他浏览器
10    }
11    return finalFileName;
12 }
13
```

```
1 @RequestMapping("/downExcel")
2 public void downExcel(HttpServletRequest request, HttpServletResponse response) {
3     out = response.getOutputStream(); // 取得输出流
4     response.reset(); // 清空输出流
5     response.setHeader("Content-disposition",
6         "attachment; filename=" + ExcelUtil.toUtf8String(fileName,
7         request) + ".xlsx"); // 设定输出文件头
8     response.setContentType("application/msexcel"); // 定义输出类型
9     // 将数据写入输出流
10 }
```

```
9      //TODO
10    }
11
```

注意：下载的时候使用的Content-Type大全，一般可以使用application/octet-stream。作者：小胖学编程链接：<https://www.jianshu.com/p/d4a85d025768>来源：简书著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。