代码案例：<https://gitee.com/zx19890628/spring-boot-example/tree/master/lab_071_util_compress>

# GZIP

## GZIP概念

ZIP最早由Jean-loup Gailly和Mark Adler创建，用于UNⅨ系统的文件压缩。我们在Linux中经常会用到后缀为.gz的文件，

它们就是GZIP格式的。现今已经成为Internet上使用非常普遍的一种数据压缩格式，或者说一种文件格式。HTTP协议上的GZIP编码

是一种用来改进WEB应用程序性能的技术。大流量的WEB站点常常使用GZIP压缩技术来让用户感受更快的速度。这一般是指WWW服务器

中安装的一个功能，当有人来访问这个服务器中的网站时，服务器中的这个功能就将网页内容压缩后传输到来访的电脑浏览器中显示出来.

一般对纯文本内容可压缩到原大小的40%.这样传输就快了，效果就是你点击网址后会很快的显示出来.当然这也会增加服务器的负载.

一般服务器中都安装有这个功能模块的。

## gzip 命令

减少文件大小有两个明显的好处，一是可以减少存储空间，二是通过网络传输文件时，可以减少传输的时间。gzip 是在 Linux 系统中经常使用的一个对文件进行压缩和解压缩的命令，既方便又好用。

语法：gzip [选项] 压缩（解压缩）的文件名

该命令的各选项含义如下：

-c 将输出写到标准输出上，并保留原有文件。

-d 将压缩文件解压。

-l 对每个压缩文件，显示下列字段：

压缩文件的大小；未压缩文件的大小；压缩比；未压缩文件的名字

-r 递归式地查找指定目录并压缩其中的所有文件或者是解压缩。

-t 测试，检查压缩文件是否完整。

-v 对每一个压缩和解压的文件，显示文件名和压缩比。

-num 用指定的数字 num 调整压缩的速度，-1 或 --fast 表示最快压缩方法（低压缩比），

-9 或--best表示最慢压缩方法（高压缩比）。系统缺省值为 6。

指令实例：

gzip \*

% 把当前目录下的每个文件压缩成 .gz 文件。

gzip -dv \*

% 把当前目录下每个压缩的文件解压，并列出详细的信息。

gzip -l \*

% 详细显示例1中每个压缩的文件的信息，并不解压。

gzip usr.tar

% 压缩 tar备份文件usr.tar，此时压缩文件的扩展名为.tar.gz。

## 在liunx下操作gzip命令

gzip file用于压缩，如 gzip a.txt 将得到文件 a.txt.gz，同时删除文件a.txt。

gzip -d file.gz用于解压缩，如gzip -d a.txt.gz将得到文件a.txt，同时删除文件a.txt.gz。

## Java中的gzip文件

GZIP常常用在linxu环境下，是一种非常简单的压缩算法。在Java实现API中，它仅仅包含两个实现类：

GZIPInputStream和GZIPOutputStream。

* GZIPOutputStream类用于压缩

先说压缩实现，GZIPOutputStream只有一个方法用于压缩，就是带定长的write方法。最后调用finish方法和flush方法

* GZIPInputStream类用于解压缩

GZIPInputStream也对应GZIPOutputStream提供了一个带定长的read方法

## 案例

### 数据压缩

public static void compress(InputStream is, OutputStream os) throws Exception {

GZIPOutputStream gos = new GZIPOutputStream(os);

int count;

byte data[] = new byte[BUFFER];

while ((count = is.read(data, 0, BUFFER)) != -1) {

gos.write(data, 0, count);

}

gos.finish();//只有执行了finish方法，才是将写入的字节进行压缩算法。close方法中，也执行了finish方法。

gos.flush();

gos.close();

}

### 数据解压缩

public static void decompress(InputStream is, OutputStream os) throws Exception {

GZIPInputStream gis = new GZIPInputStream(is);

int count;

byte data[] = new byte[BUFFER];

while ((count = gis.read(data, 0, BUFFER)) != -1) {

os.write(data, 0, count);

}

gis.close();

}

## web全局gzip压缩

gzip是http协议中使用的一种加密算法,客户端向web服务器端发出了请求后，通常情况下服务器端会将页面文件和其他资源，返回到客户

端，客户端加载后渲染呈现，这种情况文件一般都比较大，如果开启Gzip ，那么服务器端响应后，会将页面，JS,CSS等文本文件或者其他文件

通过高压缩算法将其压缩，然后传输到客户端，由客户端的浏览器负责解压缩与呈现。通常能节省40%以上的流量（一般都有60%左右），一些

PHP，JSP文件也能够进行压缩。

找到Tomcat 目录下的conf下的server.xml，并找到如下信息

<Connector port="8080" maxHttpHeaderSize="8192" maxThreads="150" minSpareThreads="25" maxSpareThreads="75"

enableLookups="false" redirectPort="8443" acceptCount="100" connectionTimeout="20000"

disableUploadTimeout="true">

将它改成如下的形式（其实在上面代码的下面已经有了，将他们打开而已。）：

<Connector connectionTimeout="20000" protocol="HTTP/1.1" port="8080" maxHttpHeaderSize="8192" maxThreads="150"

minSpareThreads="25" maxSpareThreads="75" enableLookups="false" redirectPort="8443" acceptCount="100"

disableUploadTimeout="true" compression="on" compressionMinSize="2048" noCompressionUserAgents="gozilla,

traviata" compressableMimeType="text/html,text/xml" />

这样，就能够对html和xml进行压缩了，如果要压缩css 和 js，那么需要将

<Connector port="8080" ......... compressableMimeType="text/html,text/xml,text/css,text/javascript" >

你甚至可以压缩图片：

<Connector port="8080" .........

compressableMimeType="text/html,text/xml,text/css,text/javascript,image/gif,image/jpg" >

开启后重启Tomcat ，通过浏览器查看headers信息就能看到是否开启（firebug中有），

如果开启了，那么transfer-encoding就会是Gzip，否则就是chunked。

# Zip

使用jdk提供的工具即可

教程：<https://blog.csdn.net/qq_36761831/article/details/80643163>