**HTTP中的GZIP压缩实现**

**HTTP协议中关于压缩传输的规定**

第一：客户端传输到服务器请求中带有：Accept-Encoding:gzip, deflate 字段，向服务器表示，客户端支持的压缩格式（gzip或者deflate)，如果不发送改消息头，服务器是不会压缩的。  
第二：服务端在收到请求之后，如果发现请求头中含有Accept-Encoding字段，并且支持该类型的压缩，就对响应报文压缩之后返回给客户端，并且携带Content-Encoding:gzip消息头，表示响应报文是根据该格式压缩过的。  
第三：客户端接收到请求之后，先判断是否有Content-Encoding消息头，如果有，按该格式解压报文。否则按正常报文处理。

**服务端实现报文压缩**  
一般http容器都有配置实现压缩的，不需要代码实现  
*[nginx实现压缩，在nginx.conf中配置]*

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | #开启缓存  gzip on;  #启动压缩的最小报文大小  gzip\_min\_length 1k;  #压缩所使用的缓存  gzip\_buffers 4 16k;  #压缩等级1表示最快，9表示最好  gzip\_comp\_level 2;  #压缩支持的content-Type类型  gzip\_types text/plain text/javascript application/x-javascript text/css text/xml application/xml application/xml+rss; |

*[apache实现压缩，在httpd.conf中配置]*

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | #启动deflate模块  LoadModule deflate\_module modules/mod\_deflate.so  #配置压缩等级和压缩类型    DeflateCompressionLevel 6  AddOutputFilterByType DEFLATE text/plain text/html text/css text/javascript application/x-javascript |

*[tomcat配置压缩]*  
参考《[Tomcat配置gzip提高js文件下载速度](http://zhaoyanblog.com/archives/89.html" \o "Tomcat配置gzip提高js文件下载速度)》

**客户端实现报文解压缩**

*[浏览器]*  
浏览器在访问一个网站的时候，默认都是带着Accept-Encoding:gzip, deflate头信息的，当收到响应的时候，如果是压缩的格式，也会自然解压。这一般不需要我们关系的。  
*[JAVA代码实现]*  
java中访问一个url，一般使用apache的HttpClient，加上它的连接池管理，参考《[JAVA的那些池子](http://zhaoyanblog.com/archives/203.html" \o "JAVA的那些池子)》

获得一个HttpClient实现类对象：

|  |  |
| --- | --- |
| 1  2 | HttpClient httpClient =  **new** DefaultHttpClient(connectionManager, params); |

使用它去发送http请求。这个httpclient发出去的http请求是默认不带Accept-Encoding:gzip, deflate头信息的。自然返回的响应也不会是压缩的。如果你希望实现接收压缩，进行解压缩的功能，你可以使用org.apache.http.impl.client.DecompressingHttpClient，它只是对Httpclient进行了一个包装。构造方法如下

|  |  |
| --- | --- |
| 1  2  3 | HttpClient httpClient =  **new** DefaultHttpClient(connectionManager, params);  httpClient = **new** DecompressingHttpClient(httpClient); |

后面就和之前发http请求，一样使用这个httpClient了，对压缩解压缩的功能，对你写代码而言是完全透明的。