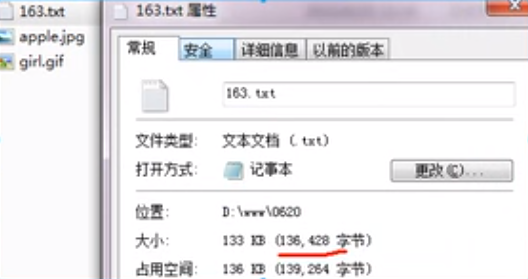
HTTP协议与内容压缩

观察：我们打开163的一篇新闻

看到如下响应头信息，注意，Content-Length：



同时，我们点右键保存其源码，得到的文本文件大小



思考：Content-Length在之前的学习中，我们知道，代表返回的主体的长度

但此处，为什么返回的主体长度和content-length不一致呢？

原因在于：Content-Encoding:gzip这个响应头信息在作用

原理：为了提高网页在网络上的传输速度，服务器对主体信息进行编码压缩

如常见的gzip压缩，deflate压缩，compress压缩以及goole chrome正在推的sdch压缩

压缩的过程是这样：

返回压缩内容

客户端：

接收压缩

再解压

再渲染页面

刚才那个情况的原因 ---- 服务器对页面进行了压缩，而content-length是“压缩后”的长度

如何在apache启用压缩功能？

1：开启deflate模块，或gzip模块

（将前面的“#”去掉）

2：重启apache

3：在conf文件中，写如下代码

<ifmodule mod\_deflate.c>

DeflateCompressionLevel 6 # 压缩级别为6，可选1-9，推荐为6

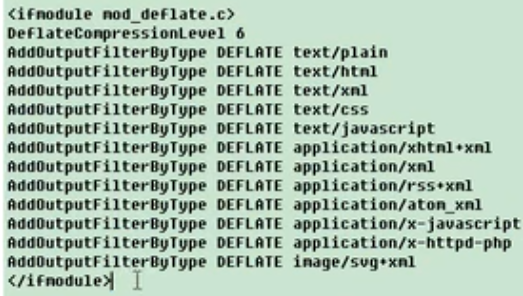
AddOutputFilterByType DEFLATE text/plain # 压缩文本文件

AddOutputFilterByType DEFLATE text/html # 压缩html文件

AddOutputFilterByType DEFLATE text/xml # 压缩xml

...

</ifmodule>

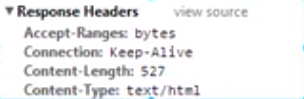


为什么要指定文件类型来压缩？

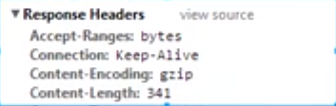
答：压缩也是要消耗CPU资源的，图片/电影等文件，压缩效果也不好

一般压缩文本格式

压缩前：



压缩后：



通过上面对比，节省了40%的流量

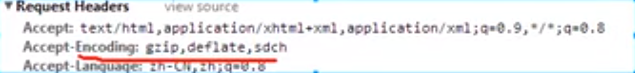
设某大型门户10亿PV，平均页面大小10000字节

每天节省流量 10亿\*10000\*0.4 === 4000G

问：服务器怎么知道我们的浏览器支持gzip的？

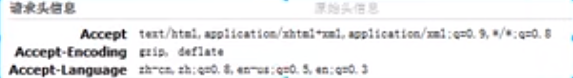
答：客户端允许发一个Accept-Encoding头信息，与服务器协商

如



这个例子可以看出，chrome浏览器支持gzip，deflate，sdch3种压缩方式

再用firefox器做测试，可以看出ff只支持gzip，deflate压缩方式



小技巧：当我们在采集时，可以不发送Accept-Encoding信息，这样采集直接是源码

当然也可以采集gzip（提高速度），再用PHP解压gzip的内容