HTTP协议缓存控制

我们观察图片的下载往往：

第一次请求时200 OK

第二次请求时304 Not Modified 未修改状态

解释：在网络上，有一些缓存服务器，另，浏览器自身也有缓存功能

当我们第一次访问某图片时，正常下载图片，返回值200

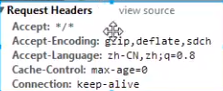
基于一个前提—图片不会经常改动，服务器在返回200的同时，还返回该图片的“签名”—Etag，（签名可以理解为图片的“指纹”）

当浏览再次访问该图片时，去服务器校验“指纹”

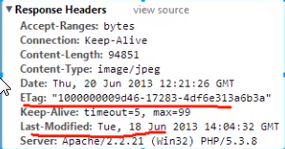
如果图片没有变化，直接使用缓存中的图片，这样减轻了服务器的负担

抓包观察：

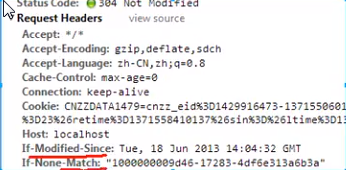
第一次请求头：



第一次响应头：



第二次请求头：



这两行的意思是：

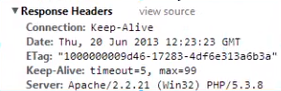
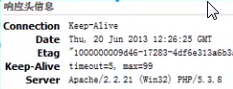
如果 自“Tue,18 Jun 2013 14:04:32 GMT”这个时间以后，图片修改过，则重新请求

如果该图片最新的Etag的值 和 If-None-Match的值不匹配，则重新请求

第二次响应信息



如果是304，就意味着浏览器从本地区缓存，节省了图片在网络上传输的时间



选学：

HTTP协议与缓存控制

如果网站比较大，有N台缓存服务器，这么着N台缓存服务器，如何处理主服务器上的文件

1：要不要缓存

2：缓存多久？

思考---这说明 缓存服务器与主服务器之间，应该有一些协议，来说明这2个问题？

追问---用什么协议来说明这2个问题？

答---还是用http协议，用头信息，cache-control来控制

具体用法：

在主服务器，打开apache的expires扩，利用该扩展来控制图片css，html等文件的

控制是否缓存及缓存生命周期

（“#”去掉，然后重启apache）

在.htaccess中，具体语法如下：（“[]”表示可写可不写）

ExpiresDefalut “<base> [plus] {<num> <type>}\*”

ExpiresByType type/encoding “>base [plus] {<num> <type>}\*”

ExpiresDefault 是设置默认的缓存参数

ExpireByType 是按照文件类型来设计独特的缓存参数

我们用第二种来测试给jpg图片设置一个月的生存周期

后面四个参数怎么理解？

Base：基于哪个时间点来计算缓存有效期

Access/now：基于请求响应的那一瞬间，比如从此瞬间到1个月以后

Modification：基于被请求文件的最后修改日期来计算，比如 最后修改日期的后1周内仍然有效

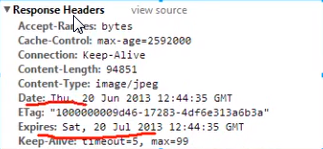
Num：缓存时间的大小（30）

Type：缓存时间的单位（天）

实例：

ExpiresActive On （查API……）

ExpiresByType image/jpeg “access plus 30 days”



如果这是在集群环境里，缓存服务器得到此图片，将会认为一个月内有效

减轻了主服务器的负担

我们能否设置服务器不让用缓存呢？

比如有些个人信息不允许缓存服务器缓存，必须到主服务器去请求

Control-cache:no-store,must-revalidate; //这意味着不允许缓存，必须去主服务器验证

例如可以利用apache的header模块，在上面代码的基础上进行添加，实现针对gif图，专门不缓存：

<FilesMatch “\.(gif)$”> //正则表达式

header set Cache-control “no-store,must-revalidate”

</FilesMatch>

多次刷新页面：





上例可以看出，girl.gif不允许缓存，因此每次都重新请求