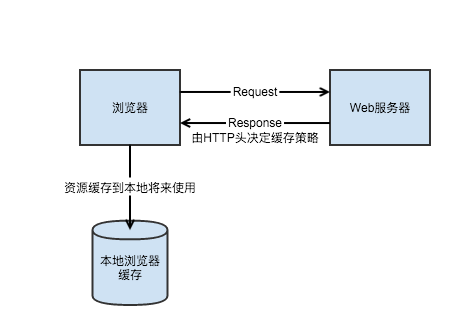
之前的文章：http://caibaojian.com/http-cache.html写得有点复杂，这篇文章从头部请求和发送源码出发，主要与Cache-Control、Expires、**Last-Modified/If-Modified-Since** 与 **ETag/If-None-Match等相关。**

以下来自：<http://foofish.net/blog/95/http-cache-policy>

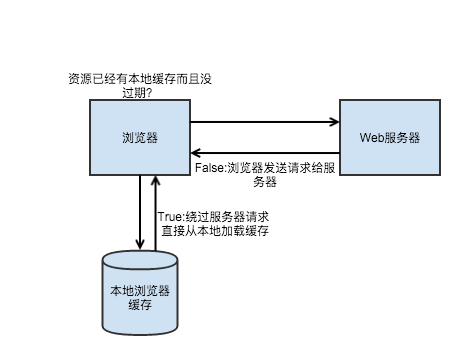
浏览器一般缓存图片、[CSS](http://caibaojian.com/css3/)、[JS](http://caibaojian.com/javascript/)等静态文件，因为这些文件的更新频率相对来说比较低，合理利用浏览器的缓存对网站的性能提升有很大帮助。[HTTP](http://caibaojian.com/t/http)缓存分为两部分，分别是本地缓存和缓存协商，当本地缓存不生效时会启用缓存协商。[HTTP](http://caibaojian.com/t/http)缓存主要由HTTP协议的头（Header）信息来制定。

[](http://img.caibaojian.com/uploads/2016/02/http-cache1.png)

本地缓存

本地缓存是指浏览器请求资源时命中了浏览器本地的缓存资源，浏览器并不会发送真正的请求给服务器了。它的执行过程是：

1. 第一次浏览器发送请求给服务器时，此时浏览器还没有本地缓存副本，服务器返回资源给浏览器，响应码是200 OK，浏览器收到资源后，把资源和对应的响应头一起缓存下来。
2. 第二次浏览器准备发送请求给服务器时候，浏览器会先检查上一次服务端返回的响应头信息中的Cache-Control，它的值是一个相对值，单位为秒，表示资源在客户端缓存的最大有效期，过期时间为第一次请求的时间减去Cache-Control的值，过期时间跟当前的请求时间比较，如果本地缓存资源没过期，那么命中缓存，不再请求服务器。
3. 如果没有命中，浏览器就会把请求发送给服务器，进入缓存协商阶段。

[](http://img.caibaojian.com/uploads/2016/02/http-cache2.png)

与本地缓存相关的头有：Cache-Control、Expires，Cache-Control有多个可选值代表不同的意义，而Expires就是一个日期格式的绝对值。

Cache-Control

Cache-Control是HTPP缓存策略中最重要的头，它是HTTP/1.1中出现的，它由如下几个值

* no-cache：不使用本地缓存。需要使用缓存协商，先与服务器确认返回的响应是否被更改，如果之前的响应中存在ETag，那么请求的时候会与服务端验证，如果资源未被更改，则可以避免重新下载。
* no-store：直接禁止游览器缓存数据，每次用户请求该资源，都会向服务器发送一个请求，每次都会下载完整的资源。
* public：可以被所有的用户缓存，包括终端用户和CDN等中间代理服务器。
* private：只能被终端用户的浏览器缓存，不允许CDN等中继缓存服务器对其缓存。
* max-age：从当前请求开始，允许获取的响应被重用的最长时间（秒）。

例如：

Cache-Control:public, max-age=1000

表示资源可以被所有用户以及代理服务器缓存，最长时间为1000秒。

Expires

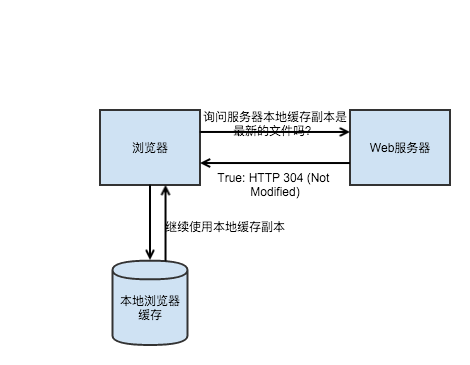
Expires是HTTP/1.0出现的头信息，同样是用于决定本地缓存策略的头，它是一个绝对时间，时间格式是如Mon, 10 Jun 2015 21:31:12 GMT，只要发送请求时间是在Expires之前，那么本地缓存始终有效，否则就会去服务器发送请求获取新的资源。如果同时出现Cache-Control：max-age和Expires，那么max-age优先级更高。他们可以这样组合使用：

Cache-Control:public

Expires: Mon, 10 Jun 2015 21:31:12 GMT

缓存协商

当第一次请求时服务器返回的响应头中没有Cache-Control和Expires或者Cache-Control和Expires过期抑或它的属性设置为no-cache时，那么浏览器第二次请求时就会与服务器进行协商，询问浏览器中的缓存资源是不是旧版本，需不需要更新，此时，服务器就会做出判断，如果缓存和服务端资源的最新版本是一致的，那么就无需再次下载该资源，服务端直接返回304 Not Modified 状态码，如果服务器发现浏览器中的缓存已经是旧版本了，那么服务器就会把最新资源的完整内容返回给浏览器，状态码就是200 Ok，那么服务端是根据什么来判断浏览器的缓存是不是最新的呢？其实是根据HTTP的另外两组头信息，分别是：**Last-Modified/If-Modified-Since** 与 **ETag/If-None-Match**。

[](http://img.caibaojian.com/uploads/2016/02/http-cache3.png)

Last-Modified与If-Modified-Since

浏览器第一次请求资源时，服务器会把资源的最新修改时间Last-Modified:Thu, 29 Dec 2011 18:23:55 GMT放在响应头中返回给浏览器，第二次请求时，浏览器就会把上一次服务器返回的修改时间放在请求头If-Modified-Since:Thu, 29 Dec 2011 18:23:55发送给服务器，服务器就会拿这个时间跟服务器上的资源的最新修改时间进行对比，如果两者相等或者大于服务器上的最新修改时间，那么表示浏览器的缓存是有效的，此时缓存会命中，服务器就不再返回内容给浏览器了，同时Last-Modified头也不会返回，因为资源没被修改，返回了也没什么意义。如果没命中缓存则最新修改的资源连同Last-Modified头一起返回。

第一次请求返回的响应头：

Cache-Control:max-age=3600

Expires:Tue, 26 Jan 2016 08:28:52 GMT

Last-Modified:Fri, 15 Jan 2016 12:06:06 GMT

第二次请求的请求头信息：

If-Modified-Since:Fri, 15 Jan 2016 12:06:06 GMT

这组头信息是基于资源的修改时间来判断资源有没有更新，另一种方式就是根据资源的内容来判断，就是接下来要讨论的ETag与If-None-Match

ETag与If-None-Match

ETag/If-None-Match与Last-Modified/If-Modified-Since的流程其实是类似的，唯一的区别是它基于资源的内容的摘要信息（比如MD5 hash）来判断。浏览器发送第二次请求时，会把第一次的响应头信息ETag的值放在If-None-Match的请求头中发送到服务器，与最新的资源的摘要信息对比，如果相等，取浏览器缓存，否则内容有更新，最新的资源连同最新的摘要信息返回。用ETag的好处是如果因为某种原因到时资源的修改时间没改变，那么用ETag就能区分资源是不是有被更新。

第一次请求返回的响应头：

Cache-Control:public, max-age=31536000

ETag: "15f0fff99ed5aae4edffdd6496d7131f"

第二次请求的请求头信息：

If-None-Match: "15f0fff99ed5aae4edffdd6496d7131f"