|  |  |
| --- | --- |
| 表 1. 常用 cache-directive 值 | |
| **Cache-directive** | **说明** |
| public | 所有内容都将被缓存(客户端和代理服务器都可缓存) |
| private | 内容只缓存到私有缓存中(仅客户端可以缓存，代理服务器不可缓存) |
| no-cache | 必须先与服务器确认返回的响应是否被更改，然后才能使用该响应来满足后续对同一个网址的请求。因此，如果存在合适的验证令牌 (ETag)，no-cache 会发起往返通信来验证缓存的响应，如果资源未被更改，可以避免下载。 |
| no-store | 所有内容都不会被缓存到缓存或 Internet 临时文件中 |
| must-revalidation/proxy-revalidation | 如果缓存的内容失效，请求必须发送到服务器/代理以进行重新验证 |
| max-age=xxx (xxx is numeric) | 缓存的内容将在 xxx 秒后失效, 这个选项只在HTTP 1.1可用, 并如果和Last-Modified一起使用时, 优先级较高 |

**不同的情形下的浏览器响应**

表 2 表明在不同的情形下，浏览器是将请求重新发送到服务器还是使用缓存的内容。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 表 2. 对 cache-directive 值的浏览器响应 | | | | |
| **Cache-directive** | **打开一个新的浏览器窗口** | **在原窗口中单击 Enter 按钮** | **刷新** | **单击 Back 按钮** |
| public | 浏览器呈现来自缓存的页面 | 浏览器呈现来自缓存的页面 | 浏览器重新发送请求到服务器 | 浏览器呈现来自缓存的页面 |
| private | 浏览器重新发送请求到服务器 | 第一次，浏览器重新发送请求到服务器；此后，浏览器呈现来自缓存的页面 | 浏览器重新发送请求到服务器 | 浏览器呈现来自缓存的页面 |
| no-cache/no-store | 浏览器重新发送请求到服务器 | 浏览器重新发送请求到服务器 | 浏览器重新发送请求到服务器 | 浏览器重新发送请求到服务器 |
| must-revalidation/proxy-revalidation | 浏览器重新发送请求到服务器 | 第一次，浏览器重新发送请求到服务器；此后，浏览器呈现来自缓存的页面 | 浏览器重新发送请求到服务器 | 浏览器呈现来自缓存的页面 |
| max-age=xxx (xxx is numeric) | 在 xxx 秒后，浏览器重新发送请求到服务器 | 在 xxx 秒后，浏览器重新发送请求到服务器 | 浏览器重新发送请求到服务器 | 在 xxx 秒后，浏览器重新发送请求到服务器 |

Cache-Control是关于浏览器缓存的最重要的设置，因为它覆盖其他设置，比如 Expires 和 Last-Modified。另外，由于浏览器的行为基本相同，这个属性是处理跨浏览器缓存问题的最有效的方法。

**失效**

Expires 头部字段提供一个日期和时间，响应在该日期和时间后被认为失效。失效的缓存条目通常不会被缓存（无论是代理缓存还是用户代理缓存）返回，除非首先通过原始服务器（或者拥有该实体的最新副本的中介缓存）验证。（注意：cache-control max-age 和 s-maxage 将覆盖 Expires 头部。）

Expires 字段接收以下格式的值：“Expires: Sun, 08 Nov 2009 03:37:26 GMT”。如果查看内容时的日期在给定的日期之前，则认为该内容没有失效并从缓存中提取出来。反之，则认为该内容失效，缓存将采取一些措施。

**不同操作下不同浏览器的响应**

表 3-6 表明针对不同用户操作的不同浏览器的行为。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 表 3. 当用户打开一个新的浏览器窗口时的失效操作 | | | | |
|  | **Firefox 3.5** | **IE 8** | **Chrome 3** | **Safari 4** |
| 内容没有失效 | 浏览器呈现来自缓存的页面 | 浏览器重新发送请求到服务器。返回代码是 200 | 浏览器呈现来自缓存的页面 | 浏览器呈现来自缓存的页面 |
| 内容失效 | 浏览器重新发送请求到服务器。返回代码是 200 | 浏览器重新发送请求到服务器。返回代码是 200 | 浏览器重新发送请求到服务器。返回代码是 200 | 浏览器重新发送请求到服务器。返回代码是 200 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 表 4. 当用户在原始浏览器窗口中单击 Enter 按钮时的失效操作 | | | | |
|  | **Firefox 3.5** | **IE 8** | **Chrome 3** | **Safari 4** |
| 内容没有失效 | 浏览器呈现来自缓存的页面 | 浏览器呈现来自缓存的页面 | 浏览器重新发送请求到服务器。返回代码是 304 | 浏览器重新发送请求到服务器。返回代码是 304 |
| 内容失效 | 浏览器重新发送请求到服务器。返回代码是 200 | 浏览器呈现来自缓存的页面 | 浏览器重新发送请求到服务器。返回代码是 200 | 浏览器重新发送请求到服务器。返回代码是 200 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 表 5. 当用户按 F5 键刷新页面时的失效操作 | | | | |
|  | **Firefox 3.5** | **IE 8** | **Chrome 3** | **Safari 4** |
| 内容没有失效 | 浏览器重新发送请求到服务器。返回代码是 304 | 浏览器重新发送请求到服务器。返回代码是 304 | 浏览器重新发送请求到服务器。返回代码是 304 | 浏览器重新发送请求到服务器。返回代码是 304 |
| 内容失效 | 浏览器重新发送请求到服务器。返回代码是 200 | 浏览器重新发送请求到服务器。返回代码是 200 | 浏览器重新发送请求到服务器。返回代码是 200 | 浏览器重新发送请求到服务器。返回代码是 200 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 表 6. 当用户单击 Back 或 Forward 按钮时的失效操作 | | | | |
|  | **Firefox 3.5** | **IE 8** | **Chrome 3** | **Safari 4** |
| 内容没有失效 | 浏览器呈现来自缓存的页面 | 浏览器呈现来自缓存的页面 | 浏览器呈现来自缓存的页面 | 浏览器呈现来自缓存的页面 |
| 内容失效 | 浏览器呈现来自缓存的页面 | 浏览器呈现来自缓存的页面 | 浏览器呈现来自缓存的页面 | 浏览器重新发送请求到服务器。返回代码是 200 |

注意：所有浏览器都假定为使用默认设置运行。

**Last-Modified/E-Tag**

Last-Modified 实体头部字段值通常用作一个缓存验证器。简单来说，如果实体值在 Last-Modified 值之后没有被更改，则认为该缓存条目有效。ETag 响应头部字段值是一个实体标记，它提供一个 “不透明” 的缓存验证器。这可能在以下几种情况下提供更可靠的验证：不方便存储修改日期；HTTP 日期值的 one-second 解决方案不够用；或者原始服务器希望避免由于使用修改日期而导致的某些冲突。

不同的浏览器有不同的配置行为。

表 7-10 表明针对不同用户操作的不同浏览器的行为。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 表 7. 当用户打开一个新的浏览器窗口时的 Last-Modified E-Tag 操作 | | | | |
|  | **Firefox 3.5** | **IE 8** | **Chrome 3** | **Safari 4** |
| 内容自上次访问以来没有被修改 | 浏览器重新发送请求到服务器。返回代码是 304 | 浏览器重新发送请求到服务器。返回代码是 200 | 浏览器重新发送请求到服务器。返回代码是 304 | 浏览器重新发送请求到服务器。返回代码是 304 |
| 内容自上次访问以来已经被修改 | 浏览器重新发送请求到服务器。返回代码是 200 | 浏览器重新发送请求到服务器。返回代码是 200 | 浏览器重新发送请求到服务器。返回代码是 200 | 浏览器重新发送请求到服务器。返回代码是 200 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 表 8. 当用户在原始浏览器窗口中单击 Enter 按钮时的 Last-Modified E-Tag 操作 | | | | |
|  | **Firefox 3.5** | **IE 8** | **Chrome 3** | **Safari 4** |
| 内容自上次访问以来没有被修改 | 浏览器呈现来自缓存的页面 | 浏览器呈现来自缓存的页面 | 浏览器重新发送请求到服务器。返回代码是 304 | 浏览器重新发送请求到服务器。返回代码是 304 |
| 内容自上次访问以来已经被修改 | 浏览器重新发送请求到服务器。返回代码是 200 | 浏览器呈现来自缓存的页面 | 浏览器重新发送请求到服务器。返回代码是 200 | 浏览器重新发送请求到服务器。返回代码是 200 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 表 9. 当用户按 F5 键刷新页面时的 Last-Modified E-Tag 操作 | | | | |
|  | **Firefox 3.5** | **IE 8** | **Chrome 3** | **Safari 4** |
| 内容自上次访问以来没有被修改 | 浏览器重新发送请求到服务器。返回代码是 304 | 浏览器重新发送请求到服务器。返回代码是 304 | 浏览器重新发送请求到服务器。返回代码是 304 | 浏览器重新发送请求到服务器。返回代码是 304 |
| 内容自上次访问以来已经被修改 | 浏览器重新发送请求到服务器。返回代码是 200 | 浏览器重新发送请求到服务器。返回代码是 200 | 浏览器重新发送请求到服务器。返回代码是 200 | 浏览器重新发送请求到服务器。返回代码是 200 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 表 10. 没有缓存设置且用户单击 Back 或 Forward 按钮 | | | | |
|  | **Firefox 3.5** | **IE 8** | **Chrome 3** | **Safari 4** |
| 内容自上次访问以来没有被修改 | 浏览器呈现来自缓存的页面 | 浏览器呈现来自缓存的页面 | 浏览器呈现来自缓存的页面 | 浏览器呈现来自缓存的页面 |
| 内容自上次访问以来已经被修改 | 浏览器呈现来自缓存的页面 | 浏览器呈现来自缓存的页面 | 浏览器呈现来自缓存的页面 | 浏览器重新发送请求到服务器。返回代码是 200 |

注意：所有浏览器都假定使用默认设置运行。

不进行任何缓存相关设置

如果您不定义任何缓存相关设置，则不同的浏览器有不同的行为。有时，同一个浏览器在相同的情形下每次运行时的行为都是不同的。情况可能很复杂。另外，有些不该缓存的内容如果被缓存，将会导致安全问题。  
　　不同的浏览器有不同的行为。

表 11 展示了不同的浏览器行为。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 表 11. 没有缓存设置且用户打开一个新的浏览器窗口 | | | | |
|  | **Firefox 3.5** | **IE 8** | **Chrome 3** | **Safari 4** |
| 打开一个新页面 | 浏览器重新发送请求到服务器。返回代码是 200 | 浏览器重新发送请求到服务器。返回代码是 200 | 浏览器重新发送请求到服务器。返回代码是 200 | 浏览器重新发送请求到服务器。返回代码是 200 |
| 在原始窗口中单击 Enter 按钮 | 浏览器重新发送请求到服务器。返回代码是 200 | 浏览器呈现来自缓存的页面。 | 浏览器重新发送请求到服务器。返回代码是 200 | 浏览器重新发送请求到服务器。返回代码是 200 |
| 按 F5 键刷新 | 浏览器重新发送请求到服务器。返回代码是 200 | 浏览器重新发送请求到服务器。返回代码是 200 | 浏览器重新发送请求到服务器。返回代码是 200 | 浏览器重新发送请求到服务器。返回代码是 200 |
| 单击 Back 或 Forward 按钮 | 浏览器呈现来自缓存的页面。 | 浏览器呈现来自缓存的页面。 | 浏览器重新发送请求到服务器。返回代码是 200 | 浏览器重新发送请求到服务器。返回代码是 200 |

注意：所有浏览器都假定使用默认设置运行。

**关键结论**

最后, 概括下关键的结论:

|  |  |
| --- | --- |
| 关键结论 | |
| 打开新窗口 | 如果指定cache-control的值为private、no-cache、must-revalidate,那么打开新窗口访问时都会重新访问服务器。而如果指定了max-age值,那么在此值内的时间里就不会重新访问服务器,例如：Cache-control: max-age=5 表示当访问此网页后的5秒内不会去再次访问服务器. |
| 在地址栏回车 | 如果值为private或must-revalidate,则只有第一次访问时会访问服务器,以后就不再访问。如果值为no-cache,那么每次都会访问。如果值为max-age,则在过期之前不会重复访问。 |
| 按后退按扭 | 如果值为private、must-revalidate、max-age,则不会重访问,而如果为no-cache,则每次都重复访问. |
| 按刷新按扭 | 无论为何值,都会重复访问. |