

# 缓存规则简介

版本号 (V1.0)



网宿科技股份有限公司

2016-08

## 目录

缓存规则简介 .....	1
1 影响缓存的 HTTP 头部 .....	3
1.1 场景一 .....	3
1.2 场景二 .....	4
2 CDN 节点缓存过期回源校验规则 .....	8
3 常见缓存规则 .....	10
3.1 遵循源站缓存 .....	10
3.2 CDN 智能缓存 .....	12
3.3 定制化缓存 .....	13
4 缓存相关增值服务 .....	14
4.1 专属缓存 .....	14

# 1 影响缓存的 HTTP 头部

## 1.1 场景一

在请求一个静态文件的时候（图片，css，js 等），这些文件的特点是文件不经常变化，浏览器将这些不经常变化的文件存储起来，对客户端来说是一个优化用户浏览体验的方法。

那么控制这些静态文件是否缓存、缓存多久是由什么决定的呢？HTTP 头部中的 Expires/Cache-Control 就是负责完成这件事情的。

### 1) Expires 头部

其值为：格林尼治标准时间；表示该内容在此时间前有效。

Eg.客户端发起对某个内容的请求，服务端做出响应，给出响应头部：Expires: Sun, 16 Oct 2016 05:43:02 GMT。 表示：服务器端希望客户端对该内容进行缓存，并且在该时间之前使用客户端本地的缓存。

### 2) Cache-Control 头部

常见值与含义如下表 1。

表格 1 Cache-Control 常见属性值说明表

请求头部	含义	响应头部	含义
<b>no-cache</b>	请求消息不能缓存（HTTP/1.0 用 Pragma 的 no-cache 替换）	<b>public</b>	响应可被任何缓存区缓存
<b>no-store</b>	在请求消息中发送将使得请求和响应消息都不使用缓存	<b>private</b>	对于单个用户的整个或部分响应消息，不能被共享缓存处理。
<b>max-age=detail-seconds</b>	客户机可以接收生存期不大于指定时间（以秒为单位）的响应。（即可缓存的最大时间）	<b>no-cache</b>	响应消息不能缓存（HTTP/1.0 用 Pragma 的 no-cache 替换）

[null]	[null]	<b>no-store</b>	在请求消息中发送将使得请求和响应消息都不使用缓存。
[null]	[null]	<b>max-age=detail-seconds</b>	客户机可以接收生存期不大于指定时间（以秒为单位）的响应。（即可缓存的最大时间）

Eg1.

```

▼ Request Headers view source
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Encoding: gzip, deflate, sdch
Accept-Language: zh-CN,zh;q=0.8
Cache-Control: no-cache
Connection: keep-alive
    
```

图 1 不缓存请求头示例

请求头中：Cache-Control:no-cache 表示所请求的内容不缓存。

Eg2.

```

▼ Response Headers view source
Cache-Control: private, max-age=10
Connection: keep-alive
Content-Encoding: gzip
Content-Type: text/html; charset=utf-8
    
```

图 2 可缓存响应头示例

响应头中：

A) Cache-Control:private 表示所响应的内容仅可被部分用户缓存；

B) max-age=10 表示所响应的内容可被缓存 10 秒钟。

**注意：** Cache-Control 优先级高于 Expires，若响应头部中存在 Cache-Control:no-cache 并且有 Expires 头部，则该文件优先遵循 Cache-Control 头部不做缓存。

## 1.2 场景二

对于已缓存的静态文件，若源站做了更新，那么当前的缓存便失去了有效性。

那么如何有效的保障缓存的可用性？HTTP 头部中的 Last-Modified/ If-Modified-since 和 ETag/ If-None-Match 有校验缓存是否过期的作用。

### 1.2.1 Last-Modified/ If-Modified-since

请求头 If-Modified-since，其值为：格林尼治标准时间；表示客户端向服务端发起校验请求，校验文件在该时间后是否有做过更新；

响应头 Last-Modified，其值为：格林尼治标准时间；表示该文件的最后修改时间。

Eg.

```
▼ Request Headers
:method: GET
:path:
:scheme: https
:version: HTTP/1.1
accept: text/css,*/*;q=0.1
accept-encoding: gzip, deflate, sdch
accept-language: zh-CN,zh;q=0.8
cache-control: max-age=0
if-modified-since: Wed, 11 May 2016 09:33:34 GMT
if-none-match: "3513-5328dbc457780"
```

图 3 带缓存校验请求头部示例

```
▼ Response Headers
accept-ranges: bytes
age: 851189
cache-control: max-age=315360000
content-encoding: gzip
content-length: 0
content-type: text/css
date: Sat, 21 May 2016 06:44:36 GMT
etag: "3513-5328dbc457780"
expires: Sat, 09 May 2026 10:18:07 GMT
last-modified: Wed, 11 May 2016 09:33:34 GMT
server: bfe/1.0.8.13-sslpool-patch
```

图 4 缓存校验响应头部示例

解析该步骤如下：

1) 客户端发起验证请求，询问服务器：该文件在 2016 年 5 月 11 日周三 09:33:34 后是否做过更新或修改？

2) 服务器校验文件：

若发现文件的最后更新时间为：2016 年 5 月 11 日周三 09:33:34，则认为该内容未更新，响应状态码 304 给客户端，不响应文件内容；

若服务端发现该文件在 2016 年 5 月 11 日周三 09:33:34 之后做过更新，即文

件最后更新时间晚于 2016 年 5 月 11 日周三 09:33:34，则认为文件做过更新，响应状态码 200 给客户端并响应更新后的文件内容；

3) 客户端根据服务器端响应的状态码判断缓存是否过期：

若状态码为 304，判断为：该文件未发生过更新或者修改，当前缓存可继续使用；

若状态码为 200，则客户端认为：该文件在 2016 年 5 月 11 日周三 09:33:34 后源站发生过更新，则删除本地旧的缓存，同步源站最新数据到本地。

注：若服务器端响应给客户端的时间早于客户端本地已缓存内容的时间，则客户端本地缓存不更新。

### 1.2.2 ETag/ If-None-Match

请求头 If-None-Match，其值为十六进制数据（可能包含“-”），表示文件标签（每个文件都有属于自己的标签，不会与其他文件相同，有唯一性。若文件发生修改，则该标签值会发生变化）；

响应头 Etag，其值为十六进制数据（可能包含“-”），表示文件标签。

Eg.

```
▼ Request Headers
:host: [REDACTED]
:method: GET
:scheme: https
:version: HTTP/1.1
accept: image/webp,image/*,*/*;q=0.8
accept-encoding: gzip, deflate, sdch
accept-language: zh-CN,zh;q=0.8
cache-control: max-age=0
if-modified-since: Thu, 26 Nov 2015 08:32:39 GMT
if-none-match: "44c-5256d6aeaf7c0"
```

图 5 带 ETag 校验请求头部示例

```
▼ Response Headers
accept-ranges: bytes
age: 15221573
cache-control: max-age=315360000
content-length: 0
content-type: image/png
date: Sat, 21 May 2016 06:44:36 GMT
etag: "44c-5256d6aeaf7c0"
expires: Mon, 24 Nov 2025 02:31:43 GMT
last-modified: Thu, 26 Nov 2015 08:32:39 GMT
```

图 6 带 ETag 响应头部示例

该步骤解析如下：

客户端发起验证请求，询问服务器端：该文件的标签值是否为：  
44c-5256d6aeaf7c0？

服务器校验文件，发现文件的标签值为：44c-5256d6aeaf7c0，与客户端缓存文件一致，认为该内容未更新响应状态码 304 给客户端，不响应文件内容（若服务端发现该文件的标签值不为：44c-5256d6aeaf7c0，则认为文件做过更新，响应状态码 200 给客户端并响应更新后的文件内容）；

客户端根据服务器端响应的状态码 304，判断为：该文件未发生过更新或者修改，当前缓存可继续使用（若服务器端响应的状态码为 200，则客户端认为：该文件在上次缓存后源文件发生过更新，则删除本地旧的缓存，同步源上最新数据到本地）。





图 8 完整校验头部内容更新示例

如图 8 节点回源校验文件是否更新，使用请求头部：`If-Modified-since: Thu, 15 Oct 2014 09:57:06 GMT` 和 `If-None-Match: "517-1413453426000"`，源服务器响应校验结果为：内容已更新，状态码 200 并响应最新的文件数据。CDN 节点认为：当前缓存已不可用，则删除本地缓存，从源站获取最新的文件数据并进行缓存。

注：若源服务器数据已经更新，但是节点回源进行校验时，源服务器响应状态码 304，那 CDN 节点仍认为源服务器对应数据未发生过更新，节点当前缓存仍可使用，不做任何更新。（客户需要保障源服务器对该请求响应的准确性，以保障节点能够从源服务器上获取到最新的数据）。

## 3 常见缓存规则

### 3.1 遵循源站缓存

遵循源的前提是源站有对不同文件类型给出对应的缓存规则，若是没有给缓存头部，CDN 默认缓存 1 分钟。

1) 源服务器响应头中包含：Cache-control:no-cache 时，即使 CDN 配置该文件可缓存，节点也会遵循源站规则，不对该文件进行缓存。可考虑配置不遵循源站缓存，则 CDN 的缓存规则优先级高于源站缓存规则。

2) 源服务器响应头中包含 Cache-control: max-age=detail-seconds 时：则该文件在 CDN 上实际缓存时间遵循源站。

Eg.

URL1: <http://caidd.8686c.com/yuyan.jpg> 源配置带 Cache-control:no-cache 头部（如图 9 所示）；

```
09:50:47]$ curl -vosa "http://caidd.8686c.com/yuyan.jpg" -x 60.220.197.8:80
* About to connect() to proxy 60.220.197.8 port 80 (#0)
* Trying 60.220.197.8... connected
* Connected to 60.220.197.8 (60.220.197.8) port 80 (#0)
> GET http://caidd.8686c.com/yuyan.jpg HTTP/1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.19.1 Basic ECC zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: caidd.8686c.com
> Accept: */*
> Proxy-Connection: Keep-Alive
>
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
  0     0     0     0     0     0     0     0     0  0  0  0  0  0  0  0  0  0  0  0  0
< Date: Fri, 03 Jun 2016 01:51:29 GMT
< Server: Apache
< Last-Modified: Wed, 20 Apr 2016 03:03:49 GMT
< ETag: "5d0038-473b9-530e1d7bfcf40"
< Accept-Ranges: bytes
< Content-Length: 291769
< Vary: Accept-Encoding
< Cache-Control: no-cache
< Connection: close
< Content-Type: image/jpeg
<
{ [data not shown]
100 284k 100 284k 0 0 259k 0 0:00:01 0:00:01 --:--:-- 277k* Closing connection #0
```

图 9 源不缓存文件配置示例

URL2: <http://caidd.8686c.com/yuyan1.gif> 源配置带 Cache-Control: max-age=86400 头部（如图 10 所示）；

```

09:36:16]$ curl -vosa "http://caidd.8686c.com/yuyan1.gif" -x 60.220.197.8:80
* About to connect() to proxy 60.220.197.8 port 80 (#0)
* Trying 60.220.197.8... connected
* Connected to 60.220.197.8 (60.220.197.8) port 80 (#0)
> GET http://caidd.8686c.com/yuyan1.gif HTTP/1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.19.1 Basic ECC zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: caidd.8686c.com
> Accept: */*
> Proxy-Connection: Keep-Alive
>
% Total % Received % Xferd Average Speed Time Time Time Current
 0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
0K HTTP/1.1 200 OK
< Date: Fri, 03 Jun 2016 01:50:46 GMT
< Server: Apache
< Last-Modified: Thu, 02 Jun 2016 08:57:26 GMT
< ETag: "5d001c-473b9-53447cb8ea980"
< Accept-Ranges: bytes
< Content-Length: 291769
< Vary: Accept-Encoding
< Cache-Control: max-age=86400
< Connection: close
< Content-Type: image/gif
<
{ [data not shown]
100 284k 100 284k 0 0 312k 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
338k* Closing connection #0
    
```

图 10 源缓存文件配置示例

以上两个文件，在 CDN 上均配置缓存 2 小时。

通过 CDN 访问 UR

```

09:24:55]$ curl -vosa "http://caidd.8686c.com/yuyan.jpg"
* About to connect() to caidd.8686c.com port 80 (#0)
* Trying 220.162.97.209... connected
* Connected to caidd.8686c.com (220.162.97.209) port 80 (#0)
> GET /yuyan.jpg HTTP/1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.19.1 Basic ECC zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: caidd.8686c.com
> Accept: */*
>
% Total % Received % Xferd Average Speed Time Time Time Current
 0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
0K HTTP/1.1 200 OK
< Date: Fri, 03 Jun 2016 01:24:59 GMT
< Server: Apache
< Last-Modified: Wed, 20 Apr 2016 03:03:49 GMT
< ETag: "5d0038-473b9-530e1d7bfcf40"
< Accept-Ranges: bytes
< Content-Length: 291769
< Cache-Control: no-cache
< Content-Type: image/jpeg
< X-Via: 1.1 jqzh233:0 (Cdn Cache Server U2.0)
< Connection: keep-alive
<
{ [data not shown]
100 284k 100 284k 0 0 315k 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
344k* Connection #0 to host caidd.8686c.com left intact
* Closing connection #0
    
```

图 11 通过 CDN 访问源不缓存文件示例

如图 11 可见，在源站有不缓存头部时，遵循源站缓存的规则下，该文件不会缓存。

通过 CDN 访问 URL2:

```

09:34:07] curl -vosa "http://caidd.8686c.com/yuyan1.gif"
* About to connect() to caidd.8686c.com port 80 (#0)
* Trying 220.162.97.209... connected
* Connected to caidd.8686c.com (220.162.97.209) port 80 (#0)
> GET /yuyan1.gif HTTP/1.1
> User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 NSS/3.19.1 Basic ECC zlib/1.2.3 libidn/1.18 libssh2/1.4.2
> Host: caidd.8686c.com
> Accept: */*
>
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
  0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
< Date: Fri, 03 Jun 2016 01:36:15 GMT
< Server: Apache
< Last-Modified: Thu, 02 Jun 2016 08:57:26 GMT
< ETag: "5d001c-473b9-53447cb8ea980"
< Accept-Ranges: bytes
< Content-Length: 291769
< Cache-Control: max-age=86400
< Content-Type: image/gif
< X-Via: 1.1 quanzhou242:6 (Cdn Cache Server V2.0)
< Connection: keep-alive
<
{ [data not shown]
100 284k 100 284k 0 0 415k 0 --:--:-- --:--:-- --:--:-- 433k* Connection #0 to host caidd.8686c.com left intact
    
```

图 12 通过 CDN 访问源可缓存文件示例

如图 12 可见，在源站有响应缓存时间控制头部 max-age 时，遵循源站缓存的规则下，具体的缓存时间也遵循源站。

## 3.2 CDN 智能缓存

### 3.2.1 网页加速智能缓存

- 1) zip、exe、wmv、swf、gif、png、bmp、wma、rar、jpeg、jpg、css、flv、mp4、txt、ico 静态文件缓存 12-24 个小时；
- 2) mp3,wma 缓存一天；
- 3) aspx、jsp、php、do、asp、do、dwr、cgi、fcgi、action、ashx、axd、json 文件不缓存；
- 4) 静态 html、htm、shtml js 文件 缓存 12 个小时；
- 5) 不符合以上四件条件的文件一律 12~24 小时；
- 6) 以上有缓存的文件如果有带问号，则缓存 10 分钟。

### 3.2.2 WSA 加速（全站加速）智能缓存

- 1) exe、wmv、mp3、wma、ogg、swf、flv、mp4、avi、mpg、mpeg、f4v、rmvb、rm、3gp、img、cur、gif、png、jpe、ico、bmp、jpeg、jpg、rpm、deb、tar、gz、bin、bz2、zip、rar、ipa、apk、jar、sis、xap、msi、cab、7z、pdf、doc、docx、xls、xlsx、ppt、pptx、txt、css 以上类型的文件缓存 12-24 个小时；
- 2) 在 1) 中缓存的文件如果有带? 则缓存 10 分钟；
- 3) html、htm、js 结尾的缓存 12 小时；
- 4) 在 3) 中的文件如果文件中带了?，则不缓存；
- 5) 不符合以上规则的文件都不缓存。

### 3.2.3 下载加速智能缓存（默认去问号缓存）

- 1)
  - 7z|apk|cab|dhp|exe|flv|gz|ipa|iso|mpk|MPQ|pbcv|pxl|qnp|r00|rar|xy|xy2|zip

- |CAB 文件类型缓存 1 个月;
- 2) php|aspx|asp|jsp|do 文件不缓存;
- 3) 其余文件缓存 5 分钟。

### 3.2.4 HTTP 点播加速智能缓存

- 1) php|aspx|asp|jsp|do 不缓存;
- 2) cgi 和带问号的文件缓存 10 分钟;
- 3) wma,mp3 文件缓存 1 天;
- 4) 其余文件缓存 1~5 小时。

## 3.3 定制化缓存

在正常情况下，CDN 节点从源站取回数据后，会按照网宿制定的默认缓存规则（CDN 智能缓存）将数据缓存在节点上，以减少回源量。但在特殊情况下，针对少数关键 URL，网宿可以根据网站的实际需求定制缓存规则，即支持由网站自行设置缓存规则。

## 4 缓存相关增值服务

### 4.1 专属缓存

专属缓存服务是网宿根据网站的实际需求新增的一项增值服务,用于实现内容永久缓存的特性。

专属缓存根据网站需要,强制将文件永久缓存在边缘节点。可保证一些重要或冷门文件(如一些教育视频文件)能及时被终端用户获取,为终端用户提供快速、优质的服务。同时,文件存储在边缘节点受到其他方面的影响更少,服务质量更加稳定。

专属缓存可分频道、文件类型和具体 URL 等方式进行功能配置。

专属缓存功能,是通过提高文件缓存的优先级,保证其存储在边缘节点不被删除,从而可以一直以缓存文件直接响应用户的请求(过期后会回源验证是否需要更新)。