

HTTP 缓存

HTTP 缓存会存储与请求关联的响应，并将存储的响应复用于后续请求。

可复用性有几个优点：

- 由于不需要将请求传递到源服务器，因此客户端和缓存越近，响应速度就越快。最典型的例子是浏览器本身为浏览器请求存储缓存。
- 源服务器不需要处理请求——因为它不需要解析和路由请求、根据 cookie 恢复会话、查询数据库以获取结果或渲染模板引擎。这减少了服务器上的负载。

1. Expires 或 max-age

Expires 标头使用明确的时间而不是通过指定经过的时间来指定缓存的生命周期。

```
Expires: Tue, 28 Feb 2022 22:22:22 GMT
```

但是时间格式难以解析，也发现了很多实现的错误，有可能通过故意偏移系统时钟来诱发问题。Cache-Control 采用了 max-age ——用于指定经过的时间。如果 Expires 和 Cache-Control: max-age 都可用，则将 max-age 定义为首选。

2. 验证响应

过时的响应不会立即被丢弃。HTTP 有一种机制，可以通过询问源服务器将陈旧的响应转换为新的响应。这称为验证，有时也称为重新验证。

验证是通过使用包含 If-Modified-Since 或 If-None-Match 请求标头的条件请求完成的。

2.1. If-Modified-Since

以下响应在 22:22:22 生成，max-age 为 1 小时，因此它在 23:22:22 之前是有效的。

```
HTTP/1.1 200 OK
Content-Type: text/html
Content-Length: 1024
Date: Tue, 22 Feb 2022 22:22:22 GMT
Last-Modified: Tue, 22 Feb 2022 22:00:00 GMT
Cache-Control: max-age=3600

<!doctype html>
...
```

到 23:22:22 时，响应会过时并且不能重用缓存。因此，下面的请求显示客户端发送带有 If-Modified-Since 请求标头的请求，以询问服务器自指定时间以来是否有任何的改变。

```
GET /index.html HTTP/1.1
Host: example.com
Accept: text/html
If-Modified-Since: Tue, 22 Feb 2022 22:00:00 GMT
```

由于此响应仅表示“没有变化”，因此没有响应主体——只有一个状态码——因此传输大小非常小。

```
HTTP/1.1 304 Not Modified
Content-Type: text/html
Date: Tue, 22 Feb 2022 23:22:22 GMT
Last-Modified: Tue, 22 Feb 2022 22:00:00 GMT
Cache-Control: max-age=3600
```

收到该响应后，客户端将存储的过期响应恢复为有效的，并可以在剩余的 1 小时内重复使用它。

2.2. If-None-Match

ETag 响应标头的值是服务器生成的任意值。服务器对于生成值没有任何限制，因此服务器可以根据他们选择的任何方式自由设置值——例如主体内容的哈希或版本号。

举个例子，如果 ETag 标头使用了 hash 值，index.html 资源的 hash 值是 deadbeef，响应如下：

```
Copy to Clipboard
HTTP/1.1 200 OK
Content-Type: text/html
Content-Length: 1024
Date: Tue, 22 Feb 2022 22:22:22 GMT
ETag: "deadbeef"
Cache-Control: max-age=3600

<!doctype html>
...
```

如果该响应是陈旧的，则客户端获取缓存响应的 ETag 响应标头的值，并将其放入 If-None-Match 请求标头中，以询问服务器资源是否已被修改：

```
Copy to Clipboard
GET /index.html HTTP/1.1
Host: example.com
Accept: text/html
If-None-Match: "deadbeef"
```

如果服务器为请求的资源确定的 ETag 标头的值与请求中的 If-None-Match 值相同，则服务器将返回 304 Not Modified。

但是，如果服务器确定请求的资源现在应该具有不同的 ETag 值，则服务器将其改为 200 OK 和资源的最新版本进行响应。