

低版本不支持此方法

缓存指定访问路径请求

`process-vary on` #指令允许你控制 HAProxy 如何处理 HTTP 响应中的 Vary 头。

Vary 头用于指示响应内容可能根据请求头的不同而变化，比如根据 `accept-encoding`（编码方式）、`referer`（引用页面）或 `origin`（请求来源）等。当设置为“on”时，HAProxy 会根据这些指定的头来创建多个缓存版本，以适应不同的请求条件。这增加了缓存的灵活性，但可能需要更多的缓存空间和管理复杂度。如果设置为“off”，则任何带有 Vary 头的响应都不会被缓存，简化了缓存逻辑，但可能导致某些情况下无法有效利用缓存。

`max-secondary-entries 12` #是与 `process-vary` 紧密相关的配置项，它允许你设定一个上限，控制可以有多少个基于上述 HTTP 头变体的缓存条目。

这意味着，即使对于相同的 URL，如果因为 Vary 头的不同而需要存储多个版本的响应，这个参数限制了最多能存储多少这样的变体。设置此参数要求 `process-vary` 必须开启，并且该值必须是一个正整数。默认值为 10，意味着在任何给定的 URL 下，最多可以有 10 种不同的缓存变体。

cache mycache

`total-max-size 4` #这个指令设置了缓存可以使用的最大内存总量，单位是 MB

`max-object-size 10000` #这指定了单个对象（如网页资源、图片等）能够存储在缓存中的最大大小，单位是字节

`max-age 240` #这个参数定义了缓存对象的有效期，单位是秒。这意味着一旦对象被缓存，它将在 30 秒后被视为过期，除非在此期间有新的请求刷新了其缓存时间

`process-vary on` #指令允许你控制 HAProxy 如何处理 HTTP 响应中的 Vary 头。此项一般无需设置

`max-secondary-entries 12` #是与 `process-vary` 紧密相关的配置项，它允许你设定一个上限，控制可以有多少个基于上述 HTTP 头变体的缓存条目。此项一般无需设置

frontend example

```
bind :80
```

```
filter cache mycache
```

```
http-request cache-use mycache #注意：frontend 部分不能用 acl 规则
```

```
http-response cache-store mycache
```

cache mycache

`total-max-size 4` #这个指令设置了缓存可以使用的最大内存总量，单位是 MB

`max-object-size 10000` #这指定了单个对象（如网页资源、图片等）能够存储在缓存中的最大大小，单位是字节

`max-age 240` #这个参数定义了缓存对象的有效期，单位是秒。这意味着一旦对象被缓存，它将在 30 秒后被视为过期，除非在此期间有新的请求刷新了其缓存时间

`process-vary on` #指令允许你控制 HAProxy 如何处理 HTTP 响应中的 Vary 头。此项一般无需设置

`max-secondary-entries 12` #是与 `process-vary` 紧密相关的配置项，它允许你设定一个上限，控制可以有多少个基于上述 HTTP 头变体的缓存条目。此项一般无需设置

backend servers

http-request cache-use mycache if { path_end .css }

http-response cache-store mycache