

重定向

URL 重定向（也称为 URL 转发）是一种为页面、表单或者整个 Web 站点/应用提供多个 URL 地址的技术。HTTP 对此操作有一种特殊类型的响应，称为 HTTP 重定向（HTTP redirect）。

重定向可实现许多目标：

- 站点维护或停机期间的临时重定向。
- 永久重定向将在更改站点的 URL 后，保留现有的链接/书签、上传文件时表示进度的页面等。

1. 原理

重定向操作由服务器向请求发送特殊的重定向响应而触发。重定向响应包含以 3 开头的状态码，以及 Location 标头，其保存着重定向的 URL。

浏览器在接收到重定向时，它们会立刻加载 Location 标头中提供的新 URL。除了额外的往返操作中会有一小部分性能损失之外，重定向操作对于用户来说是不可见的。

不同类型的重定向映射可以划分为三个类别：

- 永久重定向
- 临时重定向
- 特殊重定向

1.1. 永久重定向

这种重定向操作是永久性的。它表示原 URL 不应再被使用，而选用新的 URL 替换它。搜索引擎机器人、RSS 阅读器以及其他爬虫将更新资源原始的 URL。

状态码	状态文本	处理方法	典型应用场景
301	Moved Permanently	GET 方法不会发生变更。其他方法有可能会变更为 GET 方法。	网站重构。
308	Permanent Redirect	方法和消息主体都不发生变化。	使用用于非 GET 链接/操作重组网站。

1.2. 临时重定向

有时候请求的资源无法从其标准地址访问，但是却可以从另外的地方访问。在这种情况下，可以使用临时重定向。搜索引擎和其他爬虫不会记录新的、临时的 URL。

状态码	状态文本	处理方法	典型应用场景
302	Found	GET 方法不会发生变更。其他方法有可能会变更为 GET 方法。	由于不可预见的原因该页面暂不可用。

状态码	状态文本	处理方法	典型应用场景
303	See Other	GET 方法不会发生变更，其他方法会变更为 GET 方法（消息主体丢失）。	用于 PUT 或 POST 请求完成之后重定向，来防止由于页面刷新导致的操作的重复触发。
307	Temporary Redirect	方法和消息主体都不发生变化。	由于不可预见的原因该页面暂不可用。当站点支持非 GET 方法的链接或操作的时候，该状态码优于 302 状态码。

1.3. 特殊重定向

304（Not Modified）会使页面跳转到本地的缓存副本中（可能已过时），而 300（Multiple Choice）则是一种手动重定向：将消息主体以 Web 页面形式呈现在浏览器中：

状态码	状态文本	典型应用场景
300	Multiple Choice	不常用：所有的选项在消息主体的 HTML 页面中列出。鼓励在 Link 标头中加入机器可读的 rel=alternate
304	Not Modified	发送用于重新验证的条件请求。表示缓存的响应仍然是新的并且可以使用。

2. 指定重定向的其他方式

HTTP 重定向不是定义重定向的唯一方法。还有两个：

- 借助 HTML 的 `<meta>` 元素的 HTML 重定向机制
- 借助 DOM 的 JavaScript 重定向机制。

2.1. HTML 重定向

HTML 页面可以包含一个 `<meta>` 元素，并将其 `http-equiv` 属性的值设置为 `refresh`。该元素指定重定向：

```
<head>
  <meta http-equiv="Refresh" content="0; URL=http://example.com/" />
</head>
```

`content` 属性的值开头是一个数字，指示浏览器在等待该数字表示的秒数之后再进行跳转。在实际应用中，操作成功后的页面跳转就很适合这种 HTML 重定向。

2.2. JavaScript 重定向

在 JavaScript 中，重定向机制的原理是设置 `window.location` 的属性值，然后加载新的页面。

2.3. 优先级

1. HTTP 协议的重定向机制永远最先触发——它们甚至在没有传输页面的情况下就已经存在。
2. HTML 的重定向机制 (`<meta>`) 会在没有任何 HTTP 协议重定向的情况下触发。
3. JavaScript 的重定向机制总是作为最后诉诸的手段，并且只有在客户端开启了 JavaScript 的情况下才起作用。

3. 应用场景

3.1. 域名别称

(1) 扩大站点的用户覆盖面

假如站点位于 `www.example.com` 域名下，那么通过 `example.com` 也应该可以访问到。这种情况下，可以建立从 `example.com` 的页面到 `www.example.com` 的重定向。此外还可以提供域名常见的同义词，或者该域名容易导致的拼写错误的别称来重定向到你的网站。

(2) 迁移到新域名

公司改名后，希望用户在搜索旧名称的时候，依然可以访问到应用了新名称的站点。

4. 在服务器中配置重定向

(1) Nginx

在 Nginx 中，可以创建一个服务器模块来进行重定向设置：

```
server {  
    listen 80;  
    server_name example.com;  
    return 301 $scheme://www.example.com$request_uri;  
}
```

要将重定向应用于目录或者仅是部分页面，请使用 `rewrite` 指令：

```
rewrite ^/images/(.*)$ http://images.example.com/$1 redirect;  
rewrite ^/images/(.*)$ http://images.example.com/$1 permanent;
```

5. 重定向死锁（循环）

当后续的重定向路径重复之前的路径的时候，重定向循环就产生了。避免重定向循环非常重要，因为它会完全毁掉用户的体验。