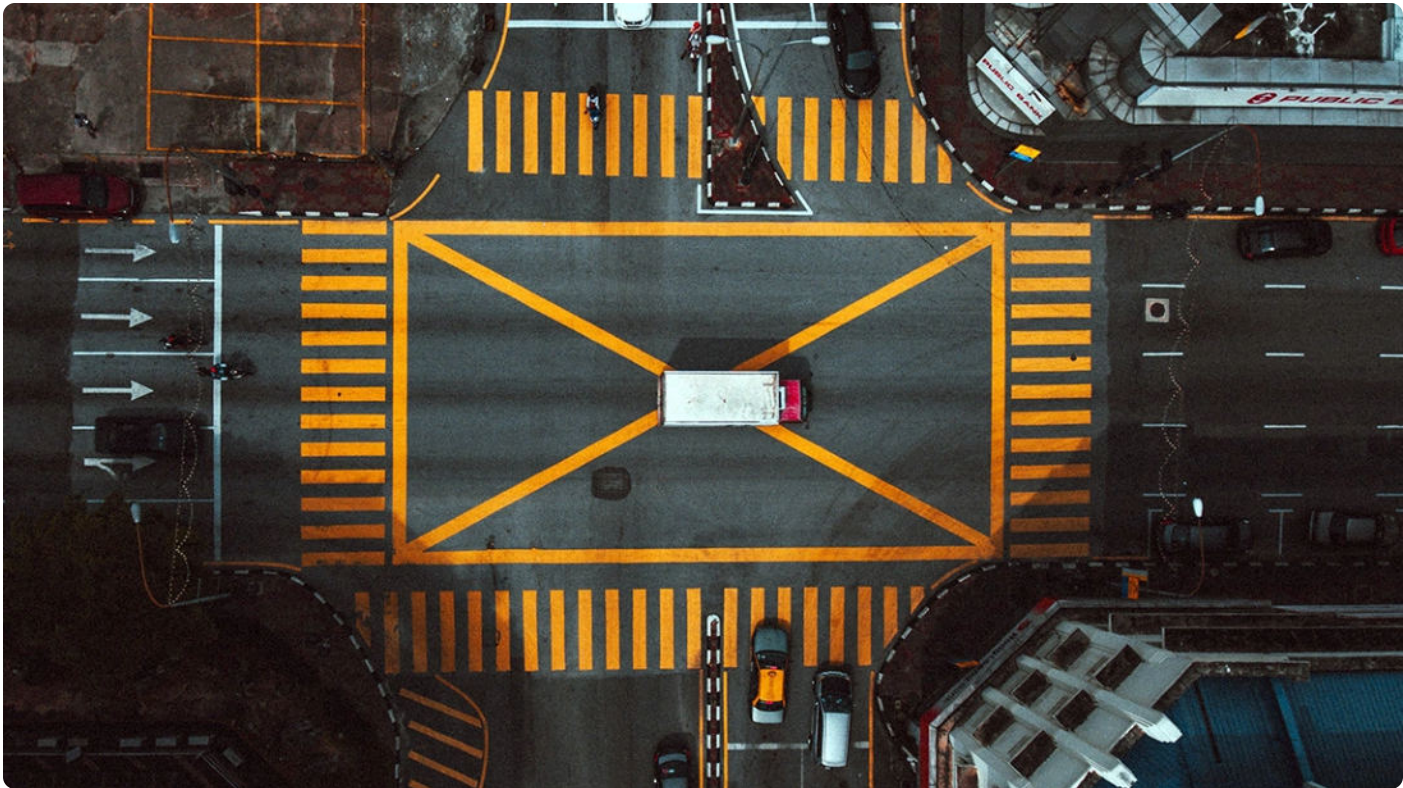


# 18 | 四通八达：HTTP的重定向和跳转

2019-07-08 Chrono

《透视HTTP协议》

课程介绍 >



讲述：Chrono

时长 10:34 大小 12.11M



在专栏 [第 1 讲](#) 时我曾经说过，为了实现在互联网上构建超链接文档系统的设想，蒂姆·伯纳斯 - 李发明了万维网，使用 HTTP 协议传输“超文本”，让全世界的人都能够自由地共享信息。

“超文本”里含有“超链接”，可以从一个“超文本”跳跃到另一个“超文本”，对线性结构的传统文档是一个根本性的变革。

能够使用“超链接”在网络上任意地跳转也是万维网的一个关键特性。它把分散在世界各地的文档连接在一起，形成了复杂的网状结构，用户可以在查看时随意点击链接、转换页面。再加上浏览器又提供了“前进”“后退”“书签”等辅助功能，让用户在文档间跳转时更加方便，有了更多的主动性和交互性。

领资料



那么，点击页面“链接”时的跳转是怎样的呢？具体一点，比如在 Nginx 的主页上点了一下“download”链接，会发生什么呢？

结合之前的课程，稍微思考一下你就能得到答案：浏览器首先要解析链接文字里的 URI。

复制代码

```
1 http://nginx.org/en/download.html
```

再用这个 URI 发起一个新的 HTTP 请求，获取响应报文后就会切换显示内容，渲染出新 URI 指向的页面。

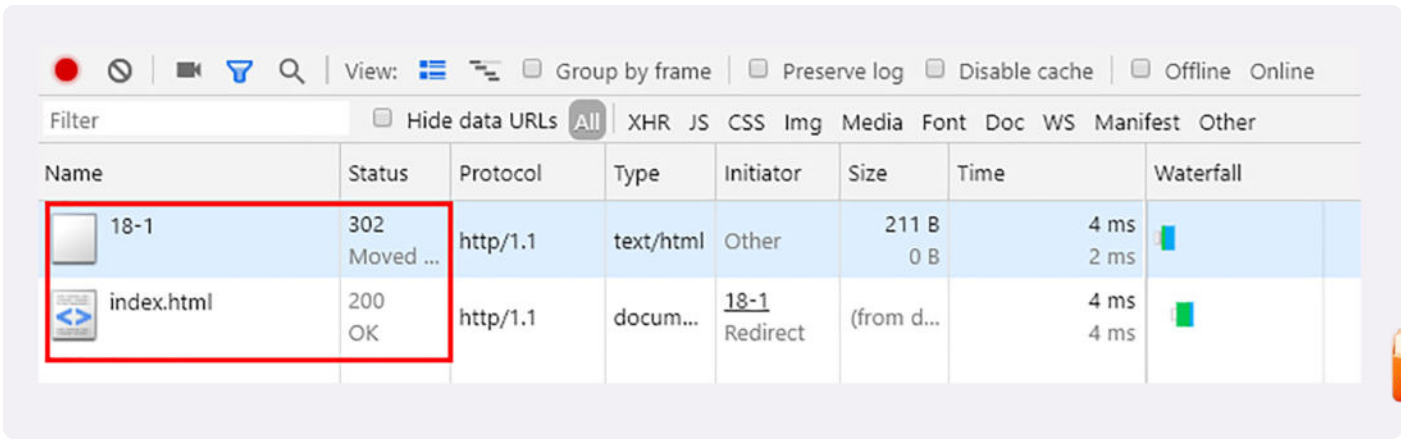
这样的跳转动作是由浏览器的使用者主动发起的，可以称为“**主动跳转**”，但还有一类跳转是由服务器来发起的，浏览器使用者无法控制，相对地就可以称为“**被动跳转**”，这在 HTTP 协议里有个专门的名词，叫做“**重定向**”（Redirection）。

### 重定向的过程

其实之前我们就已经见过重定向了，在 [第 12 讲](#)里 3xx 状态码时就说过，301 是“永久重定向”，302 是“临时重定向”，浏览器收到这两个状态码就会跳转到新的 URI。

那么，它们是怎么做到的呢？难道仅仅用这两个代码就能够实现跳转页面吗？

先在实验环境里看一下重定向的过程吧，用 Chrome 访问 URI “/18-1”，它会使用 302 立即跳转到“/index.html”。

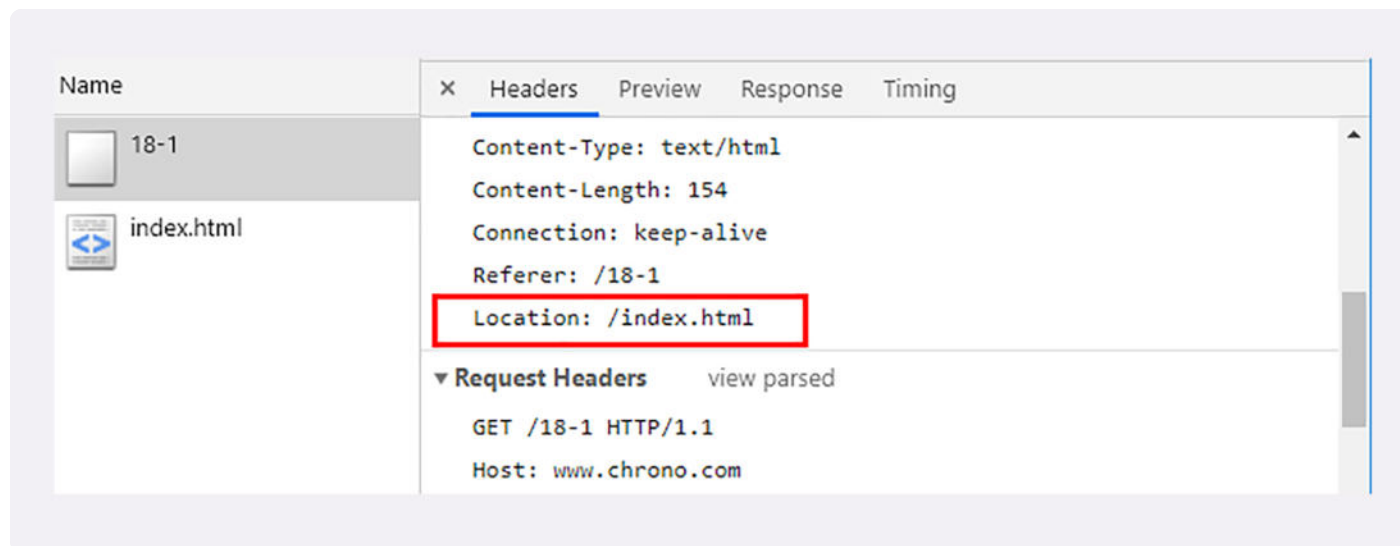


领资料

从这个实验可以看到，这一次“重定向”实际上发送了两次 HTTP 请求，第一个请求返回了 302，然后第二个请求就被重定向到了“/index.html”。但如果不用开发者工具的话，你是完全看不到这个跳转过程的，也就是说，重定向是“用户无感知”的。



我们再来看看第一个请求返回的响应报文：



这里出现了一个新的头字段“Location: /index.html”，它就是 301/302 重定向跳转的秘密所在。

“Location”字段属于响应字段，必须出现在响应报文里。但只有配合 301/302 状态码才有意义，它标记了服务器要求重定向的 URI，这里就是要求浏览器跳转到“index.html”。

浏览器收到 301/302 报文，会检查响应头里有没有“Location”。如果有，就从字段值里提取出 URI，发出新的 HTTP 请求，相当于自动替我们点击了这个链接。

在“Location”里的 URI 既可以使用绝对 URI，也可以使用相对 URI。所谓“绝对 URI”，就是完整形式的 URI，包括 scheme、host:port、path 等。所谓“相对 URI”，就是省略了 scheme 和 host:port，只有 path 和 query 部分，是不完整的，但可以从请求上下文里计算得到。

例如，刚才的实验例子中的“Location: /index.html”用的就是相对 URI。它没有说明访问 URI 的协议和主机，但因为是由“<http://www.chrono.com/18-1>”重定向返回的响应报文，所以浏览器就可以拼出完整的 URI：



复制代码

```
1 http://www.chrono.com/index.html
```

实验环境的 URI“/18-1”还支持使用 query 参数“dst=xxx”，指明重定向的 URI，你可以用这种形式再多试几次重定向，看看浏览器是如何工作的。

复制代码

```
1 http://www.chrono.com/18-1?dst=/15-1?name=a.json
2 http://www.chrono.com/18-1?dst=/17-1
```

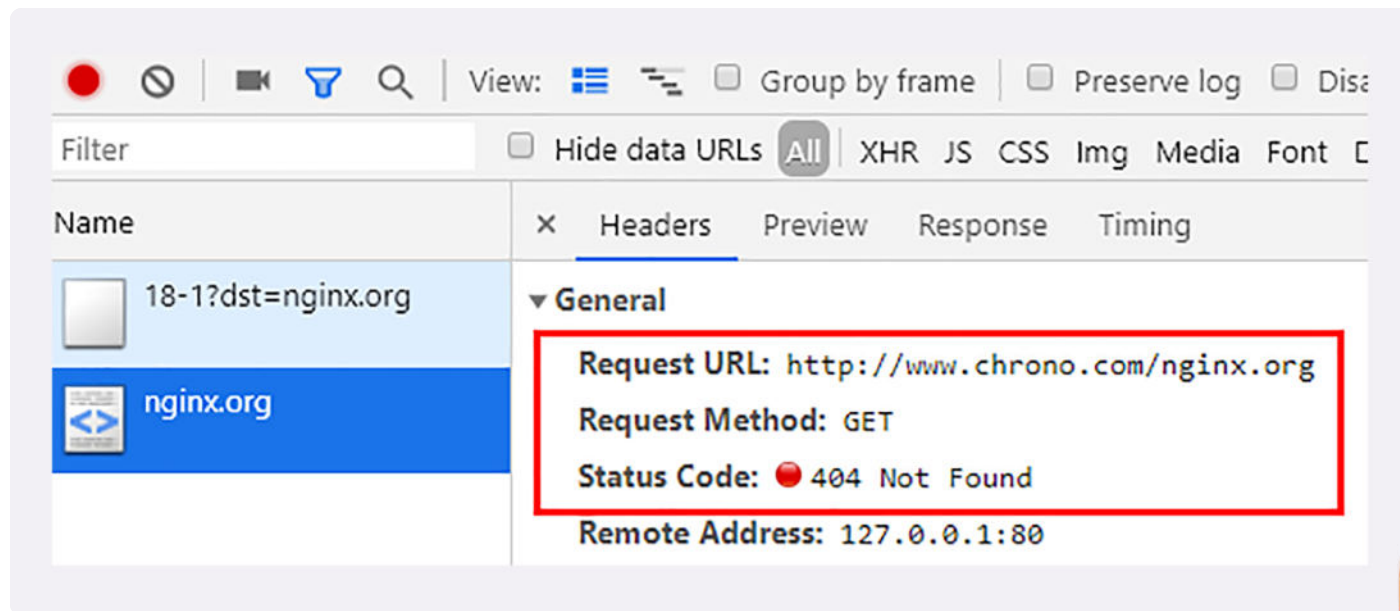
注意，在重定向时如果只是在站内跳转，你可以放心地使用相对 URI。但如果要跳转到站外，就必须用绝对 URI。

例如，如果想跳转到 Nginx 官网，就必须在“nginx.org”前把“http://”都写出来，否则浏览器会按照相对 URI 去理解，得到的就会是一个不存在的

URI“<http://www.chrono.com/nginx.org>”

复制代码

```
1 http://www.chrono.com/18-1?dst=nginx.org #错误
2 http://www.chrono.com/18-1?dst=http://nginx.org #正确
```



领资料

那么，如果 301/302 跳转时没有 Location 字段会怎么样呢？

这个你也可以自己试一下，使用第 12 讲里的 URI“/12-1”，查询参数用“code=302”：

复制代码

## 重定向状态码

刚才我把重定向的过程基本讲完了，现在来说一下重定向用到的状态码。

最常见的重定向状态码就是 301 和 302，另外还有几个不太常见的，例如 303、307、308 等。它们最终的效果都差不多，让浏览器跳转到新的 URI，但语义上有一些细微的差别，使用的时候要特别注意。

**301** 俗称“永久重定向”（Moved Permanently），意思是原 URI 已经“永久”性地不存在了，今后的所有请求都必须改用新的 URI。

浏览器看到 301，就知道原来的 URI“过时”了，就会做适当的优化。比如历史记录、更新书签，下次可能会直接用新的 URI 访问，省去了再次跳转的成本。搜索引擎的爬虫看到 301，也会更新索引库，不再使用老的 URI。

**302** 俗称“临时重定向”（“Moved Temporarily”），意思是原 URI 处于“临时维护”状态，新的 URI 是起“顶包”作用的“临时工”。

浏览器或者爬虫看到 302，会认为原来的 URI 仍然有效，但暂时不可用，所以只会执行简单的跳转页面，不记录新的 URI，也不会有其他的多余动作，下次访问还是用原 URI。

301/302 是最常用的重定向状态码，在 3xx 里剩下的几个还有：

- 303 See Other：类似 302，但要求重定向后的请求改为 GET 方法，访问一个结果页面，避免 POST/PUT 重复操作；
- 307 Temporary Redirect：类似 302，但重定向后请求里的方法和实体不允许变动，含义比 302 更明确；
- 308 Permanent Redirect：类似 307，不允许重定向后的请求变动，但它是 301“永久重定向”的含义。

不过这三个状态码的接受程度较低，有的浏览器和服务器可能不支持，开发时应当慎重，测试确认浏览器的实际效果后才能使用。

领资料



## 重定向的应用场景

理解了重定向的工作原理和状态码的含义，我们就可以在**服务器端拥有主动权**，控制浏览器的行为，不过要怎么利用重定向才好呢？

使用重定向跳转，核心是要理解“**重定向**”和“**永久 / 临时**”这两个关键词。

先来看什么时候需要重定向。

一个最常见的原因就是“**资源不可用**”，需要用另一个新的 URI 来代替。

至于不可用的原因那就很多了。例如域名变更、服务器变更、网站改版、系统维护，这些都会导致原 URI 指向的资源无法访问，为了避免出现 404，就需要用重定向跳转到新的 URI，继续为网民提供服务。

另一个原因就是“**避免重复**”，让多个网址都跳转到一个 URI，增加访问入口的同时还不会增加额外的工作量。

例如，有的网站都会申请多个名称类似的域名，然后把它们再重定向到主站上。比如，你可以访问一下“qq.com”“github.com”“bing.com”（记得事先清理缓存），看看它是如何重定向的。

决定要实行重定向后接下来要考虑的就是“永久”和“临时”的问题了，也就是选择 301 还是 302。

301 的含义是“**永久**”的。

如果域名、服务器、网站架构发生了大幅度的改变，比如启用了新域名、服务器切换到了新机房、网站目录层次重构，这些都算是“永久性”的改变。原来的 URI 已经不能用了，必须用 301“永久重定向”，通知浏览器和搜索引擎更新到新地址，这也是搜索引擎优化（SEO）要考虑的因素之一。

302 的含义是“**临时**”的。

领资料



原来的 URI 在将来的某个时间点还会恢复正常，常见的应用场景就是系统维护，把网站重定向到一个通知页面，告诉用户过一会儿再来访问。另一种用法就是“服务降级”，比如在双十一促销的时候，把订单查询、领积分等不重要的功能入口暂时关闭，保证核心服务能够正常运行。

## 重定向的相关问题

重定向的用途很多，掌握了重定向，就能够在架设网站时获得更多的灵活性，不过在使用时还需要注意两个问题。

第一个问题是“**性能损耗**”。很明显，重定向的机制决定了一个跳转会有两次请求 – 应答，比正常的访问多了一次。

虽然 301/302 报文很小，但大量的跳转对服务器的影响也是不可忽视的。站内重定向还好说，可以长连接复用，站外重定向就要开两个连接，如果网络连接质量差，那成本可就高多了，会严重影响用户的体验。

所以重定向应当适度使用，决不能滥用。

第二个问题是“**循环跳转**”。如果重定向的策略设置欠考虑，可能会出现“A=>B=>C=>A”的无限循环，不停地在这个链路里转圈圈，后果可想而知。

所以 HTTP 协议特别规定，浏览器必须具有检测“循环跳转”的能力，在发现这种情况时应当停止发送请求并给出错误提示。

实验环境的 URI “/18-2”就模拟了这样的“循环跳转”，它跳转到“/18-1”，并用参数“dst=/18-2”再跳回自己，实现了两个 URI 的无限循环。

使用 Chrome 访问这个地址，会得到“该网页无法正常运作”的结果：

领资料





该网页无法正常工作

www.chrono.com 将您重定向的次数过多。

Elements Console Sources Network Performance Memory Application Security Audits							
View: [Icon] [Icon] [Icon] Group by frame [ ] Preserve log [ ] Disable cache [ ] Offline Online [v]							
Filter [ ] Hide data URLs [x] All XHR JS CSS Img Media Font Doc WS Manifest Other							
Name	Status	Protocol	Type	Initiator	Size	Time	Waterfall
18-2 www.chrono.com	302 Moved ...	http/1.1	text/html	18-1 Redirect	198 B 0 B	4 ms 3 ms	
18-1?dst=18-2 www.chrono.com	302 Moved ...	http/1.1	text/html	18-2 Redirect	213 B 0 B	3 ms 2 ms	
18-2 www.chrono.com	302 Moved ...	http/1.1	text/html	18-1 Redirect	193 B 0 B	4 ms 2 ms	
18-1?dst=18-2 www.chrono.com	302 Moved ...	http/1.1	text/html	18-2 Redirect	213 B 0 B	24 ms 23 ms	
18-2 www.chrono.com	302 Moved ...	http/1.1	text/html	18-1 Redirect	198 B 0 B	5 ms 3 ms	

## 小结

今天我们学习了 HTTP 里的重定向和跳转，简单小结一下这次的内容：

1. 重定向是服务器发起的跳转，要求客户端改用新的 URI 重新发送请求，通常会自动进行，用户是无感知的；
2. 301/302 是最常用的重定向状态码，分别是“永久重定向”和“临时重定向”；
3. 响应头字段 Location 指示了要跳转的 URI，可以用绝对或相对的形式；
4. 重定向可以把一个 URI 指向另一个 URI，也可以把多个 URI 指向同一个 URI，用途很多；
5. 使用重定向时需要当心性能损耗，还要避免出现循环跳转。

## 课下作业

1. 301 和 302 非常相似，试着结合第 12 讲，用自己的理解再描述一下两者的异同点。
2. 你能结合自己的实际情况，再列出几个应当使用重定向的场景吗？

欢迎你把自己的学习体会写在留言区，与我和其他同学一起讨论。如果你觉得有所收获，也欢迎把文章分享给你的朋友。







## == 课外小贴士 ==

- 01 网页的“入链接”和“出链接”也是标记网页重要性的关键指标，最著名的就是 Google 发明的 PageRank。
- 02 “300 Multiple Choices”也是一个特殊的重定向状态码，它会返回一个有多个链接选项的页面，由用户自行选择要跳转的链接，用的较少。
- 03 重定向报文里还可以用 Refresh 字段，实现延时重定向，例如“Refresh: 5; url=xxx”告诉浏览器 5 秒钟后再跳转。
- 04 与跳转有关的还有一个“Referer”和“Referrer-Policy”（注意前者是个拼写错误，但已经“将错就错”），表示浏览器跳转的来源（即引用地址），可用于统计分析和防盗链。
- 05 301/302 重定向是由浏览器执行的，对于服务器来说可以称为“外部重定向”，相应的也就有服务器的“内部重定向”，直接在服务器中


领资料



有服务器的 内部重定向，直接在服务器内部跳转 URI，因为不会发出 HTTP 请求，所以没有额外的性能损失。

分享给需要的人，Ta订阅超级会员，你将得 50 元

Ta单独购买本课程，你将得 20 元

 生成海报并分享

 赞 9  提建议

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 17 | 排队也要讲效率：HTTP的连接管理

下一篇 19 | 让我知道你是谁：HTTP的Cookie机制

## 学习推荐

# JVM + NIO + Spring

各大厂面试题及知识点详解

限时免费 



领资料





钱

2020-03-29

之前面试官好像比较喜欢问外部重定向和内部重定向的区别？

外部重定向，服务器会把重定向的地址给浏览器，然后浏览器再次的发起请求，地址栏的地址变化了。

内部重定向，服务器会直接把重定向的资源返给浏览器，不需要再次在浏览器发起请求，地址栏的地址不变。

重定向我的经验，主要用在未登录或者权限不足的场景，跳转到对应的登录或提升页面之中。当然，通用的SSO也是这样做的。

作者回复: 1.内部重定向对于用户来说成本低，因为只在网站服务器内部跳转，所谓的router。

2.对，这是最常见的应用场景之一。

共 5 条评论 >

👍 36



Geek\_5b0e47

2020-03-27

笔记：

主动跳转：跳转动作是由浏览器的使用者主动发起的；

被动跳转：跳转动作是由服务器发起的，浏览器使用者无法控制。

## 1、重定向状态码

301：俗称“永久重定向”，原URI已经“永久”性地不存在了，今后的所有请求都必须改用新的URI。

302：俗称“临时重定向”，原URI处于“临时维护”状态，新的URI是起“顶包”作用的临时工。

303 See Other: 类似302，但要求重定向后的请求改为GET方法，访问一个结果页面，避免POST/PUT重复操作；

307 Temporary Redirect: 类似302，但重定向后请求里的方法和实体不允许变动，含义比302更明确；

308 Permanent Redirect: 类似307，不允许重定向后的请求变动，但它是301“永久重定向”的含义

## 2、重定向的应用场景

一个最常见的原因就是“资源不可用”，需要用另一个新的URI来代替。

不可用的原因：如域名变更、服务器变更、网站改版、系统维护。

另一个原因就是“避免重复”，让多个网址都跳转到一个URI，增加访问入口的同时还不会增加额外的工作量。如：有的网站会申请多个名称类似的域名，然后把它们重定向到主站上。

领资料



### 3、重定向的相关问题

第一个问题是“性能损耗”。重定向的机制决定了一个跳转会有两次请求-应答，比正常的访问多了一次。

第二个问题是“循环跳转”。如果重定向的策略设置欠考虑，可能会出现“A=>B=>C=>A”的无限循环。

作者回复: great。



13



Shopee内推码: NTAGx...

2019-08-19

老师，使用301会比302有较大的性能提升么

作者回复: 对于单次请求来说是没什么差别的，但浏览器会对301做优化，后续的请求就不会再有跳转动作，所以会快一些。



9



一步

2019-07-18

老师，这个301，302，303重定向要求前后协议一致吗？http不能调转https？

作者回复: 没有要求，当然可以跳转到https。



8



毛毛

2019-07-08

重定向和转发的区别和用途，以后章节会讲吗？

作者回复: 你说的“转发”指什么？是代理吗？如果是的话很快就会讲到。

共 4 条评论 >

6

领资料



夏目

2019-12-03

1、301用于废弃原地址跳转新地址，302用于暂时无法访问原地址跳转新地址，两者都需要浏览器重新发起一次请求

2、最开始接触重定向的时候就是用于未登录跳转登录页了



作者回复: 说的很好。



👍 6



snake

2020-06-22

站外重定向就要开两个连接，如果网络连接质量差，那成本可就高多了，会严重影响用户的体验。

-----》

老师这个我不理解，站外重定向，比如重定向到其他网站，那客户端的连接跟自己的服务端应该没有什么关系了吧？为什么还有两个连接呢？不是客户端应该跟其他网站的服务器连接吗？

作者回复: 是这样的：首先与本站是一个连接，然后到外站，就必须再开一个连接，这样就是两个连接了，两次tcp握手，对于客户端成本就很高了，当然服务器是无所谓的。



👍 4



III

2020-09-19

“另一个原因就是“避免重复”，让多个网址都跳转到一个 URI，增加访问入口的同时还不会增加额外的工作量。”这句话怎么理解呢？

作者回复: 比如说Google就注册了很多的域名，比如goo.gl、g.cn等等，但都指向的是同一个服务，这样对于用户来说就很方便，随便记一个就能访问。

共 2 条评论 >

👍 2



尿布

2020-09-09

重定向报文里还可以用Refresh字段，实现延时重定向，例如”Refresh: 5; url=xxx“告诉浏览器5秒钟后再跳转

与跳转有关的还有一个”Referer“和”Refereer-Policy“（注意前者是个拼写错误，但已经”将错就错“），表示浏览器跳转的来源（即引用地址），可用于统计分析和防盗链

作者回复: 把重要的信息当做笔记保存起来，这个习惯很好。



👍 2

领资料





Maske

2020-06-14

1.好比去寄快递，我去到常去的寄送点，发现寄送店有一块告示栏，被告知当前地点近期处于维修状态，需要前往另一个临时寄送点办理（302状态码），临时地址即为location字段值。或者被告知当前寄送店已永久搬迁至新地址（301状态码），临时地址为location字段值。两者都表示当前访问地址已失效，区别在于一个为临时的，短期的，另一个为永久性的。

2.比如商城类的页面，需要浏览个人中心或者订单列表等页面时需要进行登录态校验，如果没有登录或者登录态失效了，需要重定向到登录页。某个页面进行了重构，且url发生了变化，由于老url遍布站点的很多页面，不好直接修改跳转url，此时将老url重定向到新url是比较合适的。

有个疑问请教一下老师：

内部重定向和外部重定向一般在使用场合上有什么区别？

查了一下 资料说内部重定向不会造成浏览器地址栏url的变化，实际对客户端是无感知的，只是代理转发到另一个url。个人感觉又不太对，因为资料上又说nginx的重定向是属于内部，但是实际客户端url确实变了，比较矛盾

作者回复: 说的挺好。

内部重定向是服务器内部处理流程的跳转，不会发给客户端让客户端去跳转。

Nginx同时支持外部和内部重定向，写法是不同的，区别就在于客户端是否有感知。



👍 2



响雨

2019-07-09

我这边要做一个web升级，在升级过程中要展示升级进度，就打算301重定向到另一个服务来展示升级的进度

作者回复: 不错。

共 3 条评论 >

👍 2



TerryGoForIt

2019-07-08

重定向可以应用于实现负载均衡。

作者回复: 对，但多了一次请求的成本，比较重。



👍 2

领资料



饭饭





饭饭  
2019-07-08

老师您好，

重定向，我一般使用在移动PC互切的情况下会使用，因为使用到了域名会不一样。还有一种情况会在判断浏览器的时候会使用到重定向，比如IE。。。

但是有一个问题，302是临时重定向，想问一下浏览器在每次访问的时候，都会直接访问原先URI吗？还是会有什么过期时间呢？

作者回复: 302不改变原uri，所以每次都会找原uri，成本较高，应当尽量少用。



👍 2



天方夜

2021-02-21

有同学提到 ajax 中的重定向，这是个有点复杂的问题。简单说，xhr 方式，存在 Location 的情况下只能看到重定向后的最终结果；fetch 方式，可以看到 Location，而且对于重定向有不同的处理模式，可以设置。

作者回复: ajax属于前端技术了，我了解的不多，欢迎同学补充交流。



👍 2



天方夜

2021-02-20

从短域名跳长域名（z.cn – www.amazon.cn），从 http 跳 https，分别应当用哪种重定向，有没有最佳实践呢？

作者回复: 我觉得正文里的可以参考，就是看这个跳转是否是“临时”的，如果是永久条件就用301，而跟长短域名、http到https没有特别的关系。



👍 2



keep it simple

2020-01-21

老师你好，提交几个问题：

1.在自己的测试中，保存了一个https://bing.com的书签，访问后得到301，location为cn.bing.com，但书签中的地址并没有改变，还是https://bing.com。是否意味着我使用的浏览器没有遵循301的规范？

2.关于303的使用场景是什么？如果原请求方式就是POST，用于上传一组数据，然后服务器返回303，浏览器端只能用GET，那需要上传的数据就无法被上传了吧？

领资料



3.关于307的描述，言外之意是302可以改变请求里的方法和实体，但服务端只返回location的情况下，浏览器也不会改变请求方法和实体吧？

作者回复：

1.这个没有错，书签是一个正确的地址，只是bing.com给了你一个301跳转，是否更新书签是浏览器的问题，这已经不是http协议的问题了，可以更新也可以不更新。

2.303可以用来防止客户端重复post，比如post一次，后续再多次post都转向一个固定的等待响应页面。

3.302、307这些都是对客户端的“指示”，表示服务器希望客户端接下来要怎么做最合适，决定权还是在客户端。



1



Geek\_steven\_wang

2019-08-19

Sso就会用重定向引导用户登录。

但如果location中没有地址，那浏览器也不知跳转到那，会出错。

作者回复：是的。

共 2 条评论 >



1



狼的诱惑

2019-08-12

老师好，又来请教问题了

1.状态码301，302是有些客户端遵循http规范默认支持吗？比如浏览器，是不是浏览器解析了返回状态码，解析到301或302然后解析出响应头/15-1地址然后又发起了一次http请求？能举例那些客户端不支持重定向吗？

2.我觉得看文章的同时，我们是不是结合着老师提供的实验源码，来了解整个来龙去脉，这样会更容易理解？虽然老师用的lua，但还是可以勉强看懂的

作者回复：

1.浏览器为了方便用户访问网页，肯定是要自动跳转到新的uri的。如果你用Python等语言自己实现客户端，那就可以自己定义处理策略了。

2.实验环境的lua源码都很简单，只有最小的服务器逻辑，如果能参照着看是最好的。



1

领资料





信信

2019-07-24

文中提到的所有链接都返回200，和访问<http://www.chrono.com/>一个效果。。。。。。

作者回复: 建议打开开发者工具，看看uri是如何处理的。

比如<http://www.chrono.com/18-1?dst=/15-1?name=a.json>，应该是跳转到15-1。



1



一步

2019-07-18

对于ajax请求，网上好多资料说不能处理重定向的请求比如321,302,303等，不能进行跳转，这个是因为什么呢？

作者回复: 一直做后端，对前端的技术不是太了解，很抱歉无法回答你的问题。

重定向本质上是服务器向客户端发出的一个指示，如何做取决于客户端。

共 3 条评论 >



1

领资料

