

本篇文章来看看重定向那些事，即我们经常遇到的301和302状态码，本文利用实际例子进行辅助说明，希望能把重定向那些事讲明白。

在搬出主角前我们先简单说下状态码，稍微补下之前没有来得及介绍的这部分内容，状态码有很多类型，我们先看看有哪些。

一、响应状态码

我们之前学习过响应头起始行是：一般由【协议版本】、【状态码如200】及其【描述】组成，比如 HTTP/1.1 200 OK

200我们十分熟悉，表示成功，状态码的作用就十分明显了：通过此状态码可知道服务器对请求的处理结果。

还有其他一些状态码，我们简单来过一遍吧！来，八股文走起。

- **1xx 消息**：Informational（信息性状态码），表示请求已被服务器接收，继续处理。
- **2xx 消息**：Success（成功状态码），表示请求已成功被服务器接收、理解、并接受。
 - 200 OK：表示成功
 - 204 No Content：服务器成功处理了请求，但不需要返回任何实体内容。如果客户端是浏览器的话，那么用户浏览器应保留发送了该请求的页面，而不产生任何文档视图上的变化。
 - 206 Partial Content：服务器已经成功处理了部分 GET 请求。类似于 FlashGet 或者迅雷这类的 HTTP 下载工具都是使用此类响应实现断点续传或者将一个文档分解为多个下载段同时下载。该请求必须包含 Range 头信息来指示客户端希望得到的内容范围。
- **3xx 消息**：Redirection（重定向状态码），表示需要后续操作才能完成这一请求。
 - 301 Moved Permanently：被请求的资源已永久移动到新位置，并且将来任何对此资源的引用都应该使用本响应返回的若干个 URI 之一。如果可能，拥有链接编辑功能的客户端应当自动把请求的地址修改为从服务器反馈回来的地址。
 - 302 Found：请求的资源现在临时从不同的 URI 响应请求。由于这样的重定向是临时的，客户端应当继续向原有地址发送以后的请求。只有在Cache-Control或Expires中进行了指定的情况下，这个响应才是可缓存的。
- **4xx 消息**：Client Error（客户端错误状态码），表示请求含有语法错误或者无法被执行。
 - 400 Bad Request：语义有误，当前请求无法被服务器理解。除非进行修改，否则客户端不应该重复提交这个请求。请求参数有误。
 - 401 Unauthorized：当前请求需要用户验证。
 - 403 Forbidden：服务器已经理解请求，但是拒绝执行它。

- 404 Not Found: 请求失败, 请求所希望得到的资源未被在服务器上发现。
- **5xx 消息:** Server Error (服务器错误状态码), 表示服务器在处理某个正确请求时发生错误。
 - 500 Internal Server Error: 服务器遇到了不知道如何处理的情况。这个时候一般是服务出现BUG了。
 - 502 Bad Gateway: 此错误响应表明服务器作为网关需要得到一个处理这个请求的响应, 但是得到一个错误的响应。
 - 503 Service Unavailable: 表明服务器暂时处于超负载或正在进行停机维护, 无法处理请求。
 - 504 Gateway Timeout: 当服务器作为网关, 不能及时得到响应时返回此错误代码。

简单总结为:

分类	分类描述
1**	信息, 服务器收到请求, 需要请求者继续执行操作
2**	成功, 操作被成功接收并处理
3**	重定向, 需要进一步的操作以完成请求
4**	客户端错误, 请求包含语法错误或无法完成请求
5**	服务器错误, 服务器在处理请求的过程中发生了错误

 幕后哈士奇

以上的状态码都比较重要, 常见的有: 200、301、302、400、403、404、500、502、504等。

程序员最想看到的: 200-OK

程序员不想看到的: 500-Internal-Server-Error...

用户不想看到的: 401-Unauthorized、403-Forbidden、404-Not-Found...

本篇文章主要看下重定向中最常用的两个状态码: **301永久重定向**和**302临时重定向**。

二、为什么要重定向

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。

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

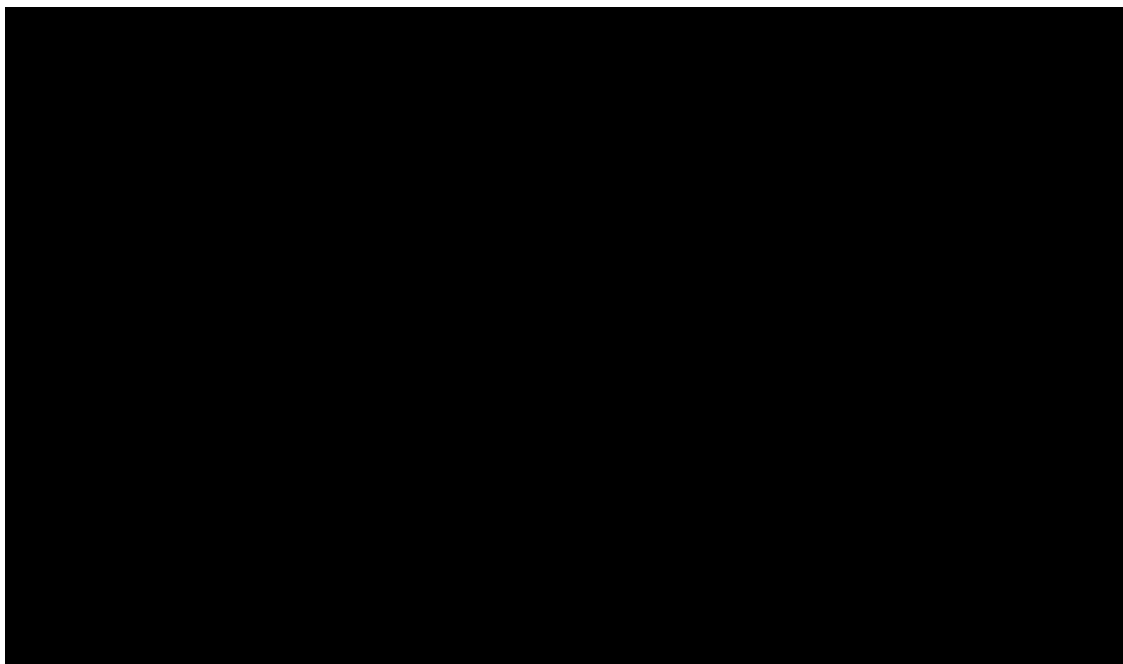
如果域名、服务器、网站架构发生了大幅度的改变，比如启用了新域名、服务器切换到了新机房、网站目录层次重构，这些都算是“永久性”的改变。原来的 URI 已经不能用了，必须用 301“永久重定向”，通知浏览器和搜索引擎更新到新地址，这也是搜索引擎优化（SEO）要考虑的因素之一

302 的含义是“临时”的

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

三、301永久重定向

拿我的一个商城项目为例，我将其部署到了云服务器上，并且从HTTP协议演变到了HTTPS，当我访问 <http://www.oursnail.cn/> 时，会出现301状态码。



301状态码对应的名称是Moved Permanently，即永久重定向，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。

301比较常用的场景是使用域名跳转。

比如我们这里的 <http://www.oursnail.cn/> 跳转到 <https://www.oursnail.cn/>，当发起请求后，就会返回301状态码。

这里出现了一个新的头字段 “Location: <https://www.oursnail.cn/>”，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“<https://www.oursnail.cn>”。

我是如何实现的呢？

实际上我的商城项目是前后端分离项目，通过nginx进行代理转发，可以对我访问的请求进行配置：

```
server {  
    listen      80;  
    server_name www.oursnail.cn;  
    rewrite ^(.*)$ https://www.oursnail.cn/$1 permanent;  
}
```

当我访问<http://www.oursnail.cn/>后，就会自动匹配到这里，将会进行 permanent 的重定向。

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，下次可能就会直接用新的 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。

四、302临时重定向

302即临时重定向，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。

在nginx中也很好配置，例如：

```
server {  
    listen      80;  
    server_name www.oursnail.cn;  
    rewrite ^(.*)$ https://www.oursnail.cn/$1 redirect;  
}
```

302用的还是比较广泛的，比如在流媒体领域，如苹果推出的大名鼎鼎的HLS协议。

（我们将单独开辟一节来介绍HLS协议，简单来说他是依靠HTTP协议的流媒体传输协议）

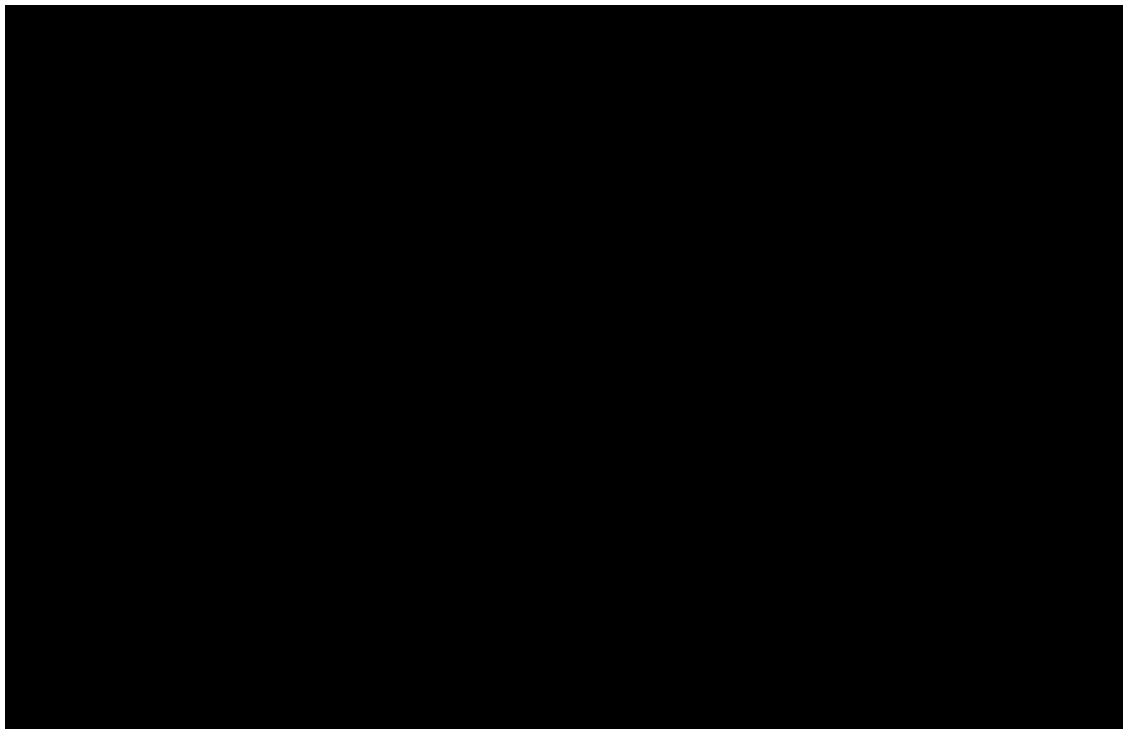
抛开这个协议本身不谈，我们来用VLC等播放器播放以下这么一条链接：

```
http://vod.jjc.cntlcdn.com/ceph_050/0099prcpaj0181161070360801837199/playlist.m3u8?CONTENTID=0099prcpaj0181161070360801837199&AUTHINFO=I5CL8h%2FBcEDNbHnR55rDy9oeRfzN5Hle1ak9%2BCLQk64sjlStiSjHMB1dTUV EJ0lqobfT9UraoYYZZS0tukVE15C0udAUfndXYIPDcUHXkfdHgTn4q1%2BwoU4D7uHps7%2FTyPBSuUIHVJJyqAtedVQMZyaZEDcpFfp6xjX6wG8mwZ6ZkKF63lldLltXTuc uC9IJ%2BFb13l2IZlQ6FtEl9ABybEYxEF%2BoPuSAiuugWg%2FwGw07IYCx5zVsB4yohjXGbW1F1S0%2FufnSGB9LC%2FklCVh2tQ%3D%3D&USERTOKEN=6j9bVCgXbG4PNLUDl8sTyxDU3Avb0BfE
```

注意，这条链接有时效性，记得没错的话，防盗链的有效时间我设置的是到21年12月底。

防盗链机制是指某个资源设置防盗校验，比较常见的是在访问资源url的后面增加加密后的信息，当请求CDN等服务器时，服务端会进行校验，比如是否可以解密？解密后的时间戳是否在有效范围内？有效防止了盗链的存在，否则腾讯的视频被直接搬到爱奇艺上，腾讯会答应吗？

不多说，来播放一下，并且顺便抓一个包来看看：

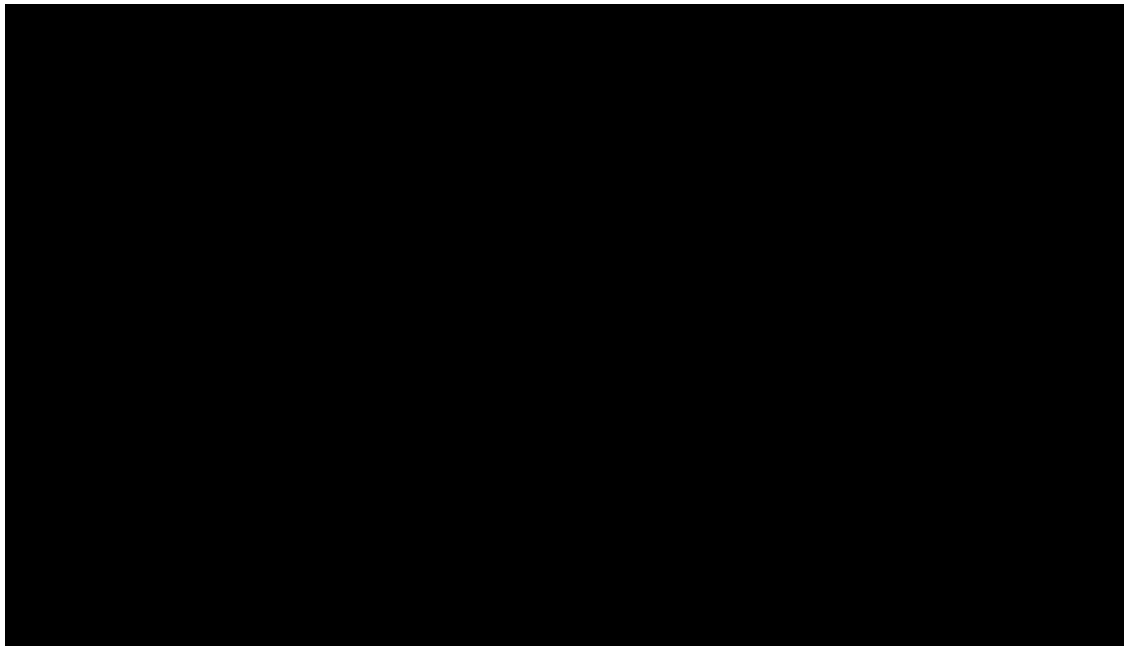


抓包的部分结果为：（这里我只过滤出来了HTTP报文，TCP报文已被隐藏）



可以看到出现了302状态码。

对这条302报文进行放大，可以看到类似上面301的响应报文：



这里面的原理是什么？实际上这里是CDN常用的一个机制，即就近原则响应。

当身处江苏的我访问该视频资源的时候，CDN根据我的地址进行CDN的选择，让离我最近的CDN为我提供服务，这是非常有意义的，试想，新疆的用户访问跟江苏用户访问到的CDN节点是一样的话，CDN加速能快吗？

所以这里302的结果往往是根据实际情况进行不同的临时选择，有可能这个CDN节点损坏，会将你临时调配到另外一个备份或稍远的节点上。

所以，在这个场景下，302并不是说因为系统维护临时重定向，多少是带着“故意为之”的意味啦。