[前端 - 如何理解HTTP响应的状态码\_个人文章 - SegmentFault 思否](https://segmentfault.com/a/1190000023467319)

HTTP 响应状态代码指示特定 HTTP 请求是否已成功完成。状态代码由 section 10 of RFC 2616定义。响应分为五类：

* 1xx：信息
* 2xx：成功
* 3xx：重定向
* 4xx：客户端错误
* 5xx：服务器错误

### 一般故障排除提示

* 当使用Web浏览器测试Web服务器时，在更改服务器后刷新浏览器
* 检查服务器日志以获取有关服务器如何处理请求的更多详细信息。 例如，网络服务器，如Apache或Nginx的生成两个文件名为access.log和error.log ，可以为相关的信息进行扫描。
* ****请记住：****HTTP状态代码定义是由提供请求的应用程序实现的标准的一部分。 这意味着返回的****实际状态代码取决于服务器软件如何处理特定错误**** – 本指南通常应该指向正确的方向

一些常见的状态码为：

### 1xx（临时响应）表示临时响应并需要请求者继续执行操作的状态代码。

这一类型的状态码，代表请求已被接受，需要继续处理。这类响应只包含状态行和某些可选的响应头信息，并以空行结束。

由于HTTP/1.0协议中没有定义任何1xx状态码，所以除非在某些试验条件下，服务器禁止向此类客户端发送1xx响应。

****100（继续）：****迄今为止的所有内容都是可行的，客户端应该继续请求，如果已经完成，则忽略它。

服务器已经接收到请求头，并且客户端应继续发送请求主体（在需要发送身体的请求的情况下：例如，POST请求），或者如果请求已经完成，忽略这个响应。服务器必须在请求完成后向客户端发送一个最终响应。要使服务器检查请求的头部，客户端必须在其初始请求中发送Expect: 100-continue作为头部，并在发送正文之前接收100 Continue状态代码。

****101 （切换协议）：****该代码是响应客户端的 Upgrade 标头发送的，并且指示服务器也正在切换的协议。

服务器已经理解了客户端的请求，并将通过Upgrade消息头通知客户端采用不同的协议来完成这个请求。在发送完这个响应最后的空行后，服务器将会切换到在Upgrade消息头中定义的那些协议。

只有在切换新的协议更有好处的时候才应该采取类似措施。例如，切换到新的HTTP版本（如HTTP/2）比旧版本更有优势，或者切换到一个实时且同步的协议（如WebSocket）以传送利用此类特性的资源。

****102 Processing（WebDAV；RFC 2518）：****

WebDAV请求可能包含许多涉及文件操作的子请求，需要很长时间才能完成请求。该代码表示服务器已经收到并正在处理请求，但无响应可用。这样可以防止客户端超时，并假设请求丢失。

****103 Early Hints：**** 此状态代码主要用于与Link 链接头一起使用，以允许用户代理在服务器仍在准备响应时开始预加载资源。

### 2xx （成功） 表示请求已成功被服务器接收、理解、并接受。

****200（成功）：****请求已成功，请求所希望的响应头或数据体将随此响应返回。

成功的含义取决于HTTP方法：

* GET：资源已被提取并在消息正文中传输。
* HEAD：实体标头位于消息正文中。
* POST：描述动作结果的资源在消息体中传输。
* TRACE：消息正文包含服务器收到的请求消息

****201（已创建）：****该请求已成功，并因此创建了一个新的资源。这通常是在POST请求，或是某些PUT请求之后返回的响应。

****202（已接受）：****服务器已接受请求，但尚未处理。最终该请求可能会也可能不会被执行，并且可能在处理发生时被禁止。如：需要的资源无法及时创建。

****203（非授权信息）（自HTTP / 1.1起）：****服务器已成功处理了请求，但返回的信息可能来自另一来源。

****204（无内容）：****服务器成功处理了请求，没有返回任何内容。

****205（重置内容）：****服务器成功处理了请求，但没有返回任何内容。

但是与204响应不同，返回此状态码的响应要求请求者重置文档视图。该响应主要是被用于接受用户输入后，立即重置表单，以便用户能够轻松地开始另一次输入。与204响应一样，该响应也被禁止包含任何消息体，且以消息头后的第一个空行结束。

****206 （部分内容）：****服务器成功处理了部分 GET 请求。实现断点续传或者将一个大文档分解为多个下载段同时下载。如：视频播放时的数据加载请求。

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，并且可能包含 If-Range 来作为请求条件。

****207 （多状态）：****由WebDAV(RFC 2518)扩展的状态码，代表之后的消息体将是一个XML消息，并且可能依照之前子请求数量的不同，包含一系列独立的响应代码。

****208 (已报告) （WebDAV；RFC 5842）：****在 DAV 里面使用: propstat 响应元素以避免重复枚举多个绑定的内部成员到同一个集合。

****226（IM Used）（RFC 3229）：****服务器已经完成了对资源的 GET 请求，并且响应是对当前实例应用的一个或多个实例操作结果的表示。

### 3xx （重定向） 表示要完成请求，需要进一步操作。

表示要完成请求，需要进一步操作。 通常，这些状态代码用来重定向。

当且仅当后续的请求所使用的方法是GET或者HEAD时，用户浏览器才可以在没有用户介入的情况下自动提交所需要的后续请求。客户端应当自动监测无限循环重定向（例如：A→B→C→……→A或A→A），因为这会导致服务器和客户端大量不必要的资源消耗。按照HTTP/1.0版规范的建议，****浏览器不应自动访问超过5次的重定向。****

****300（多种选择）：****被请求的资源有一系列可供选择的回馈信息，每个都有自己特定的地址和浏览器驱动的商议信息。用户或浏览器能够自行选择一个首选的地址进行重定向。

如果服务器本身已经有了首选的回馈选择，那么在Location中应当指明这个回馈的URI；浏览器可能会将这个Location值作为自动重定向的地址。此外，除非额外指定，否则这个响应也是可缓存的。

****301 （永久移除）：****请求的网页已永久移动到新位置。服务器返回此响应（对 GET 或 HEAD 请求的响应）时，会自动将请求者转到新位置。

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。除非额外指定，否则这个响应也是可缓存的。

新的永久性的URI应当在响应的Location域中返回。除非这是一个HEAD请求，否则响应的实体中应当包含指向新的URI的超链接及简短说明。如果这不是一个GET或者HEAD请求，因此浏览器禁止自动进行重定向，除非得到用户的确认，因为请求的条件可能因此发生变化。

对于某些使用HTTP/1.0协议的浏览器，当它们发送的POST请求得到了一个301响应的话，接下来的重定向请求将会变成GET方式。

****302（临时移动）：****服务器目前从不同位置的网页响应请求，但请求者应继续使用原有位置来进行以后的请求。

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。

新的临时性的URI应当在响应的Location域中返回。除非这是一个HEAD请求，否则响应的实体中应当包含指向新的URI的超链接及简短说明。

如果这不是一个GET或者HEAD请求，那么浏览器禁止自动进行重定向，除非得到用户的确认，因为请求的条件可能因此发生变化。

虽然RFC 1945和RFC 2068规范不允许客户端在重定向时改变请求的方法，但是很多现存的浏览器将302响应视作为303响应，并且使用GET方式访问在Location中规定的URI，而无视原先请求的方法。因此状态码303和307被添加了进来，用以明确服务器期待客户端进行何种反应。

****303（查看其他位置）：****对应当前请求的响应可以在另一个 URI 上被找到，而且客户端应当采用 GET 的方式访问那个资源。这个方法的存在主要是为了允许由脚本激活的POST请求输出重定向到一个新的资源。

同时，303响应禁止被缓存。当然，第二个请求（重定向）可能被缓存。

新的URI应当在响应的Location域中返回。除非这是一个HEAD请求，否则响应的实体中应当包含指向新的URI的超链接及简短说明。

许多HTTP/1.1版以前的浏览器不能正确理解303状态。如果需要考虑与这些浏览器之间的互动，302状态码应该可以胜任，因为大多数的浏览器处理302响应时的方式恰恰就是上述规范要求客户端处理303响应时应当做的。

****304 （未修改）：****表示资源未被修改，因为请求头指定的版本If-Modified-Since或If-None-Match。在这种情况下，由于客户端仍然具有以前下载的副本，因此不需要重新传输资源。

304 响应禁止包含消息体，因此始终以消息头后的第一个空行结尾。

****305（使用代理）：****被请求的资源必须通过指定的代理才能被访问。

Location 域中将给出指定的代理所在的 URI 信息，接收者需要重复发送一个单独的请求，通过这个代理才能访问相应资源。只有原始服务器才能建立305响应。

RFC 2068中没有明确305响应是为了重定向一个单独的请求，而且只能被原始服务器建立。忽视这些限制可能导致严重的安全后果。

****306（临时重定向）：****在最新版的规范中，306状态码已经不再被使用。

****307（临时重定向）：**** 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。

****308（永久重定向）：****这意味着资源现在永久位于由 Location: HTTP Response 标头指定的另一个 URI。

这意味着资源现在永久位于由 Location: HTTP Response 标头指定的另一个 URI。 这与 301 Moved Permanently HTTP 响应代码具有相同的语义，但用户代理不能更改所使用的 HTTP 方法：如果在第一个请求中使用 POST，则必须在第二个请求中使用 POST。

### 4xx（客户端错误）表示请求可能出错，妨碍了服务器的处理。

这类的状态码代表了****客户端看起来可能发生了错误****，妨碍了服务器的处理。除非响应的是一个HEAD请求，否则服务器就应该返回一个解释当前错误状况的实体，以及这是临时的还是永久性的状况。

如果错误发生时客户端正在传送数据，那么使用TCP的服务器实现应当仔细确保在关闭客户端与服务器之间的连接之前，客户端已经收到了包含错误信息的数据包。如果客户端在收到错误信息后继续向服务器发送数据，服务器的TCP栈将向客户端发送一个重置数据包，以清除该客户端所有还未识别的输入缓冲，以免这些数据被服务器上的应用程序读取并干扰后者。

****400（错误请求）：****由于明显的客户端错误，服务器不能或不会处理该请求。

* 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。比如：传参的数据结构不对
* 请求格式有误。如：由于浏览器故障，导致请求格式错误
* 参数数据量太大
* 无效的请求消息或欺骗性路由请求
* 与该网站相关联的用户Cookie已损坏。 清除浏览器的缓存和Cookie可以解决此问题
* 恶意代码的请求由于人为错误手动形成HTTP请求时（例如，使用curl错误）

****401（未授权）：**** 请求要求身份验证。 对于需要登录的网页，服务器可能返回此响应。

当前请求需要用户验证。该响应必须包含一个适用于被请求资源的 WWW-Authenticate 信息头用以询问用户信息。客户端可以重复提交一个包含恰当的 Authorization 头信息的请求。如果当前请求已经包含了 Authorization 证书，那么401响应代表着服务器验证已经拒绝了那些证书。如果401响应包含了与前一个响应相同的身份验证询问，且浏览器已经至少尝试了一次验证，那么浏览器应当向用户展示响应中包含的实体信息，因为这个实体信息中可能包含了相关诊断信息。

注意：当网站（通常是网站域名）禁止IP地址时，有些网站状态码显示的401，表示该特定地址被拒绝访问网站。

****402 （要求付费）：****此响应码保留以便将来使用，创造此响应码的最初目的是用于数字支付系统，然而现在并未使用。

如果特定开发人员已超过请求的每日限制，Google Developers API会使用此状态码。

****403（禁止）：****服务器已经理解请求，但是拒绝执行它。

与 401 响应不同的是，身份验证并不能提供任何帮助，而且这个请求也不应该被重复提交。如果这不是一个 HEAD 请求，而且服务器希望能够讲清楚为何请求不能被执行，那么就应该在实体内描述拒绝的原因。当然服务器也可以返回一个 404 响应，假如它不希望让客户端获得任何信息。

* 执行访问被禁止。
* 读访问被禁止。
* 写访问被禁止。
* 要求 SSL。
* 要求 SSL 128。
* IP 地址被拒绝。
* 要求客户端证书。
* 站点访问被拒绝。
* 用户数过多。
* 配置无效。
* 密码更改。
* 拒绝访问映射表。
* 客户端证书被吊销。
* 拒绝目录列表。
* 超出客户端访问许可。
* 客户端证书不受信任或无效。
* 客户端证书已过期或尚未生效。
* 在当前的应用程序池中不能执行所请求的 URL。
* 不能为这个应用程序池中的客户端执行 CGI。
* Passport 登录失败。

****404（未找到）：****请求失败，请求所希望得到的资源未被在服务器上发现。

没有信息能够告诉用户这个状况到底是暂时的还是永久的。假如服务器知道情况的话，应当使用410状态码来告知旧资源因为某些内部的配置机制问题，已经永久的不可用，而且没有任何可以跳转的地址。404这个状态码被广泛应用于当服务器不想揭示到底为何请求被拒绝或者没有其他适合的响应可用的情况下。

* 资源的链接是否有打字错误？
* 用户键入的网址有误吗？
* 文件是否存在于服务器上的正确位置？ 资源是否已在服务器上移动或删除？
* 服务器配置是否具有正确的文档根位置？
* 拥有Web服务器工作进程的用户是否有权限遍历到请求的文件所在的目录？ （提示：目录需要访问读取和执行权限）
* 访问的资源是否是符号链接？ 如果是，请确保Web服务器配置为遵循符号链接

****405（方法禁用）：****请求行中指定的请求方法（post、get、put、delete等）不能被用于请求相应的资源。

该响应必须返回一个Allow 头信息用以表示出当前资源能够接受的请求方法的列表。 鉴于 PUT，DELETE 方法会对服务器上的资源进行写操作，因而绝大部分的网页服务器都不支持或者在默认配置下不允许上述请求方法，对于此类请求均会返回405错误。

****406（（不接受） ）：****请求的资源的内容特性无法满足请求头中的条件，因而无法生成响应实体。

除非这是一个HEAD请求，否则该响应就应当返回一个包含可以让用户或者浏览器从中选择最合适的实体特性以及地址列表的实体。实体的格式由Content-Type头中定义的媒体类型决定。浏览器可以根据格式及自身能力自行作出最佳选择。但是，规范中并没有定义任何作出此类自动选择的标准。

****407（（需要代理授权））：****与401响应类似，只不过客户端必须在代理服务器上进行身份验证。

代理服务器必须返回一个 Proxy-Authenticate 用以进行身份询问。客户端可以返回一个 Proxy-Authorization 信息头用以验证。

****408（（请求超时） ）：****请求超时。客户端没有在服务器预备等待的时间内完成一个请求的发送。客户端可以随时再次提交这一请求而无需进行任何更改。

****409（冲突）：****由于和被请求的资源的当前状态之间存在冲突，请求无法完成。

这个代码只允许用在这样的情况下才能被使用：用户被认为能够解决冲突，并且会重新提交新的请求。该响应应当包含足够的信息以便用户发现冲突的源头。如：多个同步更新之间的编辑冲突。

****410（已删除）：****被请求的资源在服务器上已经不再可用，而且没有任何已知的转发地址。

这样的状况应当被认为是永久性的。如果可能，拥有链接编辑功能的客户端应当在获得用户许可后删除所有指向这个地址的引用。如果服务器不知道或者无法确定这个状况是否是永久的，那么就应该使用 404 状态码。除非额外说明，否则这个响应是可缓存的。

****411（需要有效长度）：****服务器拒绝在没有定义 Content-Length 头的情况下接受请求。在添加了表明请求消息体长度的有效 Content-Length 头之后，客户端可以再次提交该请求。

****412（前提条件失败）：****服务器在验证在请求的头字段中给出先决条件时，没能满足其中的一个或多个。这个状态码允许客户端在获取资源时在请求的元信息（请求头字段数据）中设置先决条件，以此避免该请求方法被应用到其希望的内容以外的资源上。

****413（请求实体过大）：****服务器拒绝处理当前请求，因为该请求提交的实体数据大小超过了服务器愿意或者能够处理的范围。

此种情况下，服务器可以关闭连接以免客户端继续发送此请求。如果这个状况是临时的，服务器应当返回一个 Retry-After 的响应头，以告知客户端可以在多少时间以后重新尝试。

****414（请求的 URI 过长）：****请求的URI 长度超过了服务器能够解释的长度，因此服务器拒绝对该请求提供服务。

* 本应使用POST方法的表单提交变成了GET方法，导致查询字符串（Query String）过长。
* 重定向URI“黑洞”，例如每次重定向把旧的URI作为新的URI的一部分，导致在若干次重定向后URI超长。
* 客户端正在尝试利用某些服务器中存在的安全漏洞攻击服务器。这类服务器使用固定长度的缓冲读取或操作请求的URI，当GET后的参数超过某个数值后，可能会产生缓冲区溢出，导致任意代码被执行。

****415（不支持的媒体类型） ：****对于当前请求的方法和所请求的资源，请求中提交的实体并不是服务器中所支持的格式，因此请求被拒绝。

****416（请求范围不符合要求） （RFC 7233）：****如果请求中包含了 Range 请求头，并且 Range 中指定的任何数据范围都与当前资源的可用范围不重合，同时请求中又没有定义 If-Range 请求头。

****417（未满足期望值）：****此响应代码意味着服务器无法满足 Expect 请求标头字段指示的期望值。

****418 （I'm a teapot）（RFC 2324）：****本操作码是在1998年作为IETF的传统愚人节笑话，在RFC 2324超文本咖啡壶控制协议中定义的，并不需要在真实的HTTP服务器中定义。当一个控制茶壶的HTCPCP收到BREW或POST指令要求其煮咖啡时应当回传此错误。

这个HTTP状态码在某些网站（包括Google.com）与项目（如Node.js、ASP.NET和Go语言）中用作彩蛋。

****420（保持冷静）：****Twitter Search与Trends API在客户端被限速的情况下返回。

****421（错误请求）（RFC 7540）：****该请求针对的是无法产生响应的服务器。 这可以由服务器发送，该服务器未配置为针对包含在请求 URI 中的方案和权限的组合产生响应。

****422 （无法处理的实体）（WebDAV；RFC 4918 ）：****请求格式良好，但由于语义错误而无法遵循。

****423 （锁定）（WebDAV；RFC 4918）：****正在访问的资源被锁定。

****424 （依赖）（WebDAV；RFC 4918）：****由于先前的请求失败，所以此次请求失败。

****425 （太早了）：****服务器不愿意冒着风险去处理可能重播的请求。如：安全验证未完成。

****426 （需要升级）（RFC 2817）：****服务器拒绝使用当前协议执行请求，但可能在客户机升级到其他协议后愿意这样做。

****428 （需要前提条件） (RFC 6585)：****原始服务器要求该请求是有条件的。 旨在防止“丢失更新”问题，即客户端获取资源状态，修改该状态并将其返回服务器，同时第三方修改服务器上的状态，从而导致冲突。

****429 （请求过多） （RFC 6585）：****用户在给定的时间内发送了太多请求（“限制请求速率”）。

****431 （请求头过大）（RFC 6585）：****服务器不愿意处理请求，因为它的 请求头字段太大（ Request Header Fields Too Large）。 请求可以在减小请求头字段的大小后重新提交。

****444 (不响应)：****Nginx上HTTP服务器扩展。服务器不向客户端返回任何信息，并关闭连接（有助于阻止恶意软件）。

****450（被Windows家长控制系统阻止）：****这是一个由Windows家庭控制（Microsoft）HTTP阻止的450状态代码的示例，用于信息和测试。

****451（由于法律原因无法使用）：****用户请求非法资源，例如：由政府审查的网页。

### 5xx（服务器错误） 表示服务器在尝试处理请求时发生内部错误。

这类状态码代表了服务器在处理请求的过程中有错误或者异常状态发生，也有可能是服务器意识到以当前的软硬件资源无法完成对请求的处理。除非这是一个HEAD请求，否则服务器应当包含一个解释当前错误状态以及这个状况是临时的还是永久的解释信息实体。浏览器应当向用户展示任何在当前响应中被包含的实体。这些状态码适用于任何响应方法。

****500（内部服务器错误）：****通用错误消息，服务器遇到了一个未曾预料的状况，导致了它无法完成对请求的处理。

这此错误最常见的原因是服务器配置错误（例如，一个畸形.htaccess文件）或缺失的软件包（如试图在没有安装正确PHP执行的PHP文件）。

****501（尚未实施）：****此请求方法不被服务器支持且无法被处理。只有GET和HEAD是要求服务器支持的，它们必定不会返回此错误代码。

****502（错误网关）：****这意味着服务器的网关或代理服务器，它没有收到来自实际应该履行请求的后端服务器的有效响应。

如果有问题的服务器是逆向代理服务器，例如负载均衡器，则需要检查以下几项：

* 后端服务器（HTTP请求转发到的）是正常的
* 正确配置逆向代理，指定适当的后端
* 后端服务器和逆向代理服务器之间的网络连接是正常的。 如果服务器可以在其他端口上通信，请确保防火墙允许它们之间的流量
* 如果Web应用程序配置为侦听套接字，请确保套接字存在于正确的位置，并且具有适当的权限

****503（服务不可用）：**** 表示服务器超载或正在维护。 这个错误意味着服务应该在某个时候可用。

请注意，与此响应一起，应发送解释问题的用户友好页面。 这个响应应该用于临时条件和 Retry-After：如果可能的话，HTTP头应该包含恢复服务之前的估计时间。 网站管理员还必须注意与此响应一起发送的与缓存相关的标头，因为这些临时条件响应通常不应被缓存。

如果服务器未处于维护状态，则这可以指示服务器没有足够的CPU或内存资源来处理所有传入的请求，或者Web服务器需要配置为允许更多的用户，线程或进程

****504（网关超时）：****当服务器作为网关，不能及时得到响应时返回此错误代码。

* 服务器之间的网络连接很差
* 由于性能不佳，履行请求的后端服务器太慢
* 网关或代理服务器的超时持续时间太短

****505 （HTTP 版本不受支持）：****服务器不支持请求中所使用的HTTP协议版本。

****506 （协商引起的异常）（RFC 2295）：****对请求的透明内容协商导致循环引用。

****507 （存储不足）（WebDAV；RFC 4918）：****服务器无法存储完成请求所必须的内容。这个状况被认为是临时的。

****508 （循环检测）（WebDAV；RFC 5842）：****服务器在处理请求时陷入死循环。 可代替 208状态码

****510 （扩展不足）（RFC 2774）：****获取资源所需要的策略并没有被满足，客户端需要对请求进一步扩展，服务器才能实现它。服务器会回复客户端发出扩展请求所需的所有信息。

****511 （网络身份验证要求） （RFC 6585）：****客户端需要进行身份验证才能获得网络访问权限，旨在限制用户群访问特定网络。如：连接WiFi热点时的强制网络门户