

1xx – Informationen [\[Bearbeiten \]](#)

Die Bearbeitung der Anfrage dauert noch an.

Code	Nachricht	Bedeutung
100	<i>Continue</i>	Die laufende Anfrage an den Server wurde noch nicht zurückgewiesen. (Wird im Zusammenhang mit dem „Expect 100-continue“-Header-Feld verwendet. ^[1]) Der Client kann nun mit der potentiell sehr großen Anfrage fortfahren.
101	<i>Switching Protocols</i>	Wird verwendet, wenn der Server eine Anfrage mit gesetztem „Upgrade“-Header-Feld empfangen hat und mit dem Wechsel zu einem anderen Protokoll einverstanden ist. Anwendung findet dieser Status-Code beispielsweise im Wechsel von HTTP zu WebSocket .
102	<i>Processing</i>	Wird verwendet, um ein Timeout zu vermeiden, während der Server eine zeitintensive Anfrage bearbeitet. ^[2]

2xx – Erfolgreiche Operation [\[Bearbeiten \]](#)

Die Anfrage war erfolgreich, die Antwort kann verwertet werden.

Code	Nachricht	Bedeutung
200	<i>OK</i>	Die Anfrage wurde erfolgreich bearbeitet und das Ergebnis der Anfrage wird in der Antwort übertragen.
201	<i>Created</i>	Die Anfrage wurde erfolgreich bearbeitet. Die angeforderte Ressource wurde vor dem Senden der Antwort erstellt. Das „Location“-Header-Feld enthält eventuell die Adresse der erstellten Ressource.
202	<i>Accepted</i>	Die Anfrage wurde akzeptiert, wird aber zu einem späteren Zeitpunkt ausgeführt. Das Gelingen der Anfrage kann nicht garantiert werden.
203	<i>Non-Authoritative Information</i>	Die Anfrage wurde bearbeitet, das Ergebnis ist aber nicht unbedingt vollständig und aktuell.
204	<i>No Content</i>	Die Anfrage wurde erfolgreich durchgeführt, die Antwort enthält jedoch bewusst keine Daten.
205	<i>Reset Content</i>	Die Anfrage wurde erfolgreich durchgeführt; der Client soll das Dokument neu aufbauen und Formulareingaben zurücksetzen.
206	<i>Partial Content</i>	Der angeforderte Teil wurde erfolgreich übertragen (wird im Zusammenhang mit einem „Content-Range“-Header-Feld oder dem Content-Type multipart/byteranges verwendet). Kann einen Client über Teil-Downloads informieren (wird zum Beispiel von Wget genutzt, um den Downloadfortschritt zu überwachen oder einen Download in mehrere Streams aufzuteilen).
207	<i>Multi-Status</i>	Die Antwort enthält ein XML-Dokument, das mehrere Statuscodes zu unabhängig voneinander durchgeführten Operationen enthält.
208	<i>Already Reported</i>	WebDAV RFC 5842 [↗] – Die Mitglieder einer WebDAV-Bindung wurden bereits zuvor aufgezählt und sind in dieser Anfrage nicht mehr vorhanden.
226	<i>IM Used</i>	RFC 3229 [↗] – Der Server hat eine GET-Anforderung für die Ressource erfüllt, die Antwort ist eine Darstellung des Ergebnisses von einem oder mehreren Instanz-Manipulationen, bezogen auf die aktuelle Instanz.

3xx – Umleitung [\[Bearbeiten \]](#)

Um eine erfolgreiche Bearbeitung der Anfrage sicherzustellen, sind weitere Schritte seitens des Clients erforderlich.

Code	Nachricht	Bedeutung
300	<i>Multiple Choices</i>	Die angeforderte Ressource steht in verschiedenen Arten zur Verfügung. Die Antwort enthält eine Liste der verfügbaren Arten. Das „Location“-Header-Feld enthält eventuell die Adresse der vom Server bevorzugten Repräsentation.
301	<i>Moved Permanently</i>	Die angeforderte Ressource steht ab sofort unter der im „Location“-Header-Feld angegebenen Adresse bereit (auch Redirect genannt). Die alte Adresse ist nicht länger gültig.
302	<i>Found</i>	Die angeforderte Ressource steht vorübergehend unter der im „Location“-Header-Feld angegebenen Adresse bereit. ^[3] Die alte Adresse bleibt gültig. Die Browser folgen meist mit einem GET, auch wenn der ursprüngliche Request ein POST war. Wird in HTTP/1.1 je nach Anwendungsfall durch die Statuscodes 303 bzw. 307 ersetzt. 302-Weiterleitung ist aufgrund eines Suchmaschinen-Fehlers, dem URL-Hijacking , in Kritik geraten.
303	<i>See Other</i>	Die Antwort auf die durchgeführte Anfrage lässt sich unter der im „Location“-Header-Feld angegebenen Adresse beziehen. Der Browser soll mit einem GET folgen, auch wenn der ursprüngliche Request ein POST war.
304	<i>Not Modified</i>	Der Inhalt der angeforderten Ressource hat sich seit der letzten Abfrage des Clients nicht verändert und wird deshalb nicht übertragen. Zu den Einzelheiten siehe Browser-Cache-Versionsvergleich .
305	<i>Use Proxy</i>	Die angeforderte Ressource ist nur über einen Proxy erreichbar. Das „Location“-Header-Feld enthält die Adresse des Proxy.
306	<i>(reserviert)</i>	306 wird nicht mehr verwendet, ist aber reserviert. Es wurde für „Switch Proxy“ verwendet.
307	<i>Temporary Redirect</i>	Die angeforderte Ressource steht vorübergehend unter der im „Location“-Header-Feld angegebenen Adresse bereit. Die alte Adresse bleibt gültig. Der Browser soll mit <i>derselben</i> Methode folgen wie beim ursprünglichen Request (d. h. einem POST folgt ein POST). Dies ist der wesentliche Unterschied zu 302/303.
308	<i>Permanent Redirect</i>	Experimentell eingeführt via RFC; die angeforderte Ressource steht ab sofort unter der im „Location“-Header-Feld angegebenen Adresse bereit, die alte Adresse ist nicht länger gültig. Der Browser soll mit <i>derselben</i> Methode folgen wie beim ursprünglichen Request (d. h. einem POST folgt ein POST). Dies ist der wesentliche Unterschied zu 302/303. ^[4]

4xx – Client-Fehler [Bearbeiten]

Die Ursache des Scheiterns der Anfrage liegt (eher) im Verantwortungsbereich des Clients.

Code	Nachricht	Bedeutung
400	<i>Bad Request</i>	Die Anfrage-Nachricht war fehlerhaft aufgebaut.
401	<i>Unauthorized</i>	Die Anfrage kann nicht ohne gültige Authentifizierung durchgeführt werden. Wie die Authentifizierung durchgeführt werden soll, wird im „WWW-Authenticate“-Header-Feld der Antwort übermittelt.
402	<i>Payment Required</i>	Übersetzt: Bezahlung benötigt. Dieser Status ist für zukünftige HTTP-Protokolle reserviert.
403	<i>Forbidden</i>	Die Anfrage wurde mangels Berechtigung des Clients nicht durchgeführt, bspw. weil der authentifizierte Benutzer nicht berechtigt ist, oder eine als HTTPS konfigurierte URL nur mit HTTP aufgerufen wurde.
404	<i>Not Found</i>	Die angeforderte Ressource wurde nicht gefunden. Dieser Statuscode kann ebenfalls verwendet werden, um eine Anfrage ohne näheren Grund abzuweisen. Links, welche auf solche Fehlerseiten verweisen, werden auch als Tote Links bezeichnet.
405	<i>Method Not Allowed</i>	Die Anfrage darf nur mit anderen HTTP-Methoden (zum Beispiel GET statt POST) gestellt werden. Gültige Methoden für die betreffende Ressource werden im „Allow“-Header-Feld der Antwort übermittelt.
406	<i>Not Acceptable</i>	Die angeforderte Ressource steht nicht in der gewünschten Form zur Verfügung. Gültige „Content-Type“-Werte können in der Antwort übermittelt werden.
407	<i>Proxy Authentication Required</i>	Analog zum Statuscode 401 ist hier zunächst eine Authentifizierung des Clients gegenüber dem verwendeten Proxy erforderlich. Wie die Authentifizierung durchgeführt werden soll, wird im „Proxy-Authenticate“-Header-Feld der Antwort übermittelt.
408	<i>Request Timeout</i>	Innerhalb der vom Server erlaubten Zeitspanne wurde keine vollständige Anfrage des Clients empfangen.
409	<i>Conflict</i>	Die Anfrage wurde unter falschen Annahmen gestellt. Im Falle einer PUT-Anfrage kann dies zum Beispiel auf eine zwischenzeitliche Veränderung der Ressource durch Dritte zurückgehen.
410	<i>Gone</i>	Die angeforderte Ressource wird nicht länger bereitgestellt und wurde dauerhaft entfernt.
411	<i>Length Required</i>	Die Anfrage kann ohne ein „Content-Length“-Header-Feld nicht bearbeitet werden.
412	<i>Precondition Failed</i>	Eine in der Anfrage übertragene Voraussetzung, zum Beispiel in Form eines „If-Match“-Header-Felds, traf nicht zu.
413	<i>Request Entity Too Large</i>	Die gestellte Anfrage war zu groß, um vom Server bearbeitet werden zu können. Ein „Retry-After“-Header-Feld in der Antwort kann den Client darauf hinweisen, dass die Anfrage eventuell zu einem späteren Zeitpunkt bearbeitet werden könnte.
414	<i>Request-URL Too Long</i>	Die URL der Anfrage war zu lang. Ursache ist oft eine Endlosschleife aus Redirects.
415	<i>Unsupported Media Type</i>	Der Inhalt der Anfrage wurde mit ungültigem oder nicht erlaubtem Medientyp übermittelt.
416	<i>Requested range not satisfiable</i>	Der angeforderte Teil einer Ressource war ungültig oder steht auf dem Server nicht zur Verfügung.
417	<i>Expectation Failed</i>	Verwendet im Zusammenhang mit einem „Expect“-Header-Feld. Das im „Expect“-Header-Feld geforderte Verhalten des Servers kann nicht erfüllt werden.
418	<i>I'm a teapot</i>	Dieser Code ist als Aprilscherz der IETF zu verstehen, welcher näher unter RFC 2324 ↗ , Hyper Text Coffee Pot Control Protocol , beschrieben ist. Innerhalb eines scherzhaften Protokolls zum Kaffeekochen zeigt er an, dass fälschlicherweise eine Teekanne anstatt einer Kaffeekanne verwendet wurde. Dieser Statuscode ist allerdings kein Bestandteil von HTTP, sondern lediglich von HTCPCP ^[5] (Hyper Text Coffee Pot Control Protocol). Trotzdem ist dieser Scherz-Statuscode auf einigen Webseiten zu finden, real wird aber der Statuscode 200 gesendet. ^[6]
420	<i>Policy Not Fulfilled</i>	In W3C PEP (Working Draft 21. November 1997) ^[7] wird dieser Code vorgeschlagen, um mitzuteilen, dass eine Bedingung nicht erfüllt wurde.
421	<i>Misdirected Request</i>	Die Anfrage wurde an einen Server gesendet, der nicht in der Lage ist, eine Antwort zu senden. Eingeführt in HTTP/2 .
422	<i>Unprocessable Entity</i>	Verwendet, wenn weder die Rückgabe von Statuscode 415 noch 400 gerechtfertigt wäre, eine Verarbeitung der Anfrage jedoch zum Beispiel wegen semantischer Fehler abgelehnt wird. ^[8]
423	<i>Locked</i>	Die angeforderte Ressource ist zurzeit gesperrt. ^[8]
424	<i>Failed Dependency</i>	Die Anfrage konnte nicht durchgeführt werden, weil sie das Gelingen einer vorherigen Anfrage voraussetzt. ^[8]

425	<i>Unordered Collection</i>	In den Entwürfen von WebDav Advanced Collections definiert, aber nicht im „Web Distributed Authoring and Versioning (WebDAV) Ordered Collections Protocol“. ^[8]
426	<i>Upgrade Required</i>	Der Client sollte auf Transport Layer Security (TLS/1.0) umschalten. ^[9]
428	<i>Precondition Required</i>	Für die Anfrage sind nicht alle Vorbedingungen erfüllt gewesen. Dieser Statuscode soll Probleme durch Race Conditions verhindern, indem eine Manipulation oder Löschen nur erfolgt, wenn der Client dies auf Basis einer aktuellen Ressource anfordert (Beispielsweise durch Mitliefern eines aktuellen ETag-Header). ^[10]
429	<i>Too Many Requests</i>	Der Client hat zu viele Anfragen in einem bestimmten Zeitraum gesendet. ^[10]
431	<i>Request Header Fields Too Large</i>	Die Maximallänge eines Headerfelds oder des Gesamtheaders wurde überschritten ^[10]

Neben den standardisierten Codes nutzt [proprietäre](#) Software oftmals auch eigene Codes. Diese Codes können von anderen Browsern nur als allgemeiner unbekannter Fehler dem Benutzer angezeigt werden; nicht aber eine Übersetzung und Hinweise zum weiteren Vorgehen. Teilweise können die Server den Begleitumständen der Anfrage bereits entnehmen, dass es sich um die zugehörige Spezialsoftware handelt, und geben nur dann die privaten Codes zurück. Einige Beispiele sind nachstehend aufgelistet.

444	<i>No Response</i>	In Nginx -Logs verwendet, um anzuzeigen, dass der Server keine Informationen zum Client zurückgesendet und die Verbindung geschlossen hat.
449	<i>The request should be retried after doing the appropriate action</i>	Genutzt in Antworten des Microsoft Exchange Servers . ^[11]
451	<i>Unavailable For Legal Reasons</i>	Vom Google -Mitarbeiter Tim Bray vorgeschlagener Statuscode. Dieser Code soll darauf hinweisen, dass die angeforderte Ressource aufgrund von gesetzlichen Bestimmungen (Copyrighteinschränkungen, Zensur etc., eventuell beschränkt auf ein bestimmtes Land) nicht verfügbar ist. ^[12] Eine Danksagung am Ende dieses Vorschlages verweist auf den Autor Ray Bradbury dessen Werk Fahrenheit 451 eine Erklärung für die Wahl des vorgeschlagenen Status-Codes gerade auch für zensierte Ressourcen gibt.

5xx – Server-Fehler [Bearbeiten]

Nicht klar von den so genannten [Client](#)-Fehlern abzugrenzen. Die Ursache des Scheiterns der Anfrage liegt jedoch eher im Verantwortungsbereich des Servers.

Code	Nachricht	Bedeutung
500	<i>Internal Server Error</i>	Dies ist ein „Sammel-Statuscode“ für unerwartete Serverfehler.
501	<i>Not Implemented</i>	Die Funktionalität, um die Anfrage zu bearbeiten, wird von diesem Server nicht bereitgestellt. Ursache ist zum Beispiel eine unbekannte oder nicht unterstützte HTTP-Methode.
502	<i>Bad Gateway</i>	Der Server konnte seine Funktion als Gateway oder Proxy nicht erfüllen, weil er seinerseits eine ungültige Antwort erhalten hat.
503	<i>Service Unavailable</i>	Der Server steht temporär nicht zur Verfügung, zum Beispiel wegen Überlastung oder Wartungsarbeiten. Ein „Retry-After“-Header-Feld in der Antwort kann den Client auf einen Zeitpunkt hinweisen, zu dem die Anfrage eventuell bearbeitet werden könnte.
504	<i>Gateway Time-out</i>	Der Server konnte seine Funktion als Gateway oder Proxy nicht erfüllen, weil er innerhalb einer festgelegten Zeitspanne keine Antwort von seinerseits benutzten Servern oder Diensten erhalten hat.
505	<i>HTTP Version not supported</i>	Die benutzte HTTP-Version (gemeint ist die Zahl vor dem Punkt) wird vom Server nicht unterstützt oder abgelehnt.
506	<i>Variant Also Negotiates</i>	Die Inhaltsvereinbarung der Anfrage ergibt einen Zirkelbezug . ^[13]
507	<i>Insufficient Storage</i>	Die Anfrage konnte nicht bearbeitet werden, weil der Speicherplatz des Servers dazu zurzeit nicht mehr ausreicht. ^[8]
508	<i>Loop Detected</i>	Die Operation wurde nicht ausgeführt, weil die Ausführung in eine Endlosschleife gelaufen wäre. Definiert in der Binding-Erweiterung für WebDAV gemäß RFC 5842 ↗ , weil durch Bindings zyklische Pfade zu WebDAV-Ressourcen entstehen können.
509	<i>Bandwidth Limit Exceeded</i>	Die Anfrage wurde verworfen, weil sonst die verfügbare Bandbreite überschritten würde (inoffizielle Erweiterung einiger Server).
510	<i>Not Extended</i>	Die Anfrage enthält nicht alle Informationen, die die angefragte Server-Extension zwingend erwartet. ^[14]