status code 204

Category: Online-Marketing
geschrieben von Tobias Hager | 31. Januar 2026



Status Code 204: Erfolgreich ohne sichtbaren Inhalt

Kein Inhalt und trotzdem ein Erfolg? Willkommen in der Welt von HTTP 204 – dem unsichtbaren Helden der Webkommunikation. Während Marketingabteilungen immer noch auf flashy Banner und klickgeile Headlines setzen, gibt es da draußen einen Statuscode, der genau das Gegenteil tut: Er sagt nichts – und sagt damit alles. In diesem Artikel nehmen wir den HTTP-Statuscode 204 auseinander. Technisch. Präzise. Unbarmherzig ehrlich. Und ja, du wirst danach deinen Servercode mit anderen Augen sehen.

- Was der HTTP-Statuscode 204 bedeutet – und wann du ihn brauchst
- Warum „kein Inhalt“ nicht „keine Wirkung“ heißt
- Wie HTTP 204 in RESTful APIs und modernen Webanwendungen eingesetzt wird
- Welche Fehler beim Einsatz von 204 regelmäßig gemacht werden – und wie du sie vermeidest
- Technische Unterschiede zwischen 204, 200, 304 und 204 No Content
- Warum 204 ein Muss für saubere UX und effiziente Ladezeiten ist

- Wie du 204 im SEO-Umfeld richtig einsetzt – oder besser die Finger davon lässt
- Best Practices für Entwickler, SEOs und Performance-Freaks

HTTP-Statuscode 204: Bedeutung, Definition und technischer Kontext

Der Statuscode 204, auch bekannt als „No Content“, ist ein HTTP-Response-Code, der dem Client signalisiert: „Alles ist okay, aber ich habe nichts zurückzugeben.“ Klingt nutzlos? Ist es nicht. Der 204-Status ist ein essenzielles Werkzeug in der Kommunikation zwischen Client und Server – insbesondere in APIs, bei PUT- oder DELETE-Requests, oder bei asynchronen Hintergrundprozessen, bei denen keine neue Information zurückgegeben werden muss.

Technisch gesehen ist HTTP 204 ein 2xx-Code, was bedeutet, dass die Anfrage erfolgreich war. Aber anders als ein 200 OK enthält die Antwort keinen Body. Kein HTML, kein JSON, kein XML – einfach nichts. Und das ist genau der Punkt. Der Server sagt: „Ich habe verstanden, was du willst, ich habe es erledigt, aber ich habe dir nichts Neues mitzuteilen.“

Die RFC 9110 (HTTP Semantics) definiert klar: Ein Server, der einen 204-Status zurückgibt, darf keinen Nachrichtentext und keine Body-Daten senden. Der Client wiederum darf die aktuelle Darstellung der Ressource beibehalten. Kein Reload. Kein DOM-Update. Kein unnötiger Render-Zyklus. Eine effiziente, schlanke Kommunikation – wie sie in modernen Architekturen immer wichtiger wird.

Gerade in RESTful APIs ist der Statuscode 204 ein fester Bestandteil sauberer Schnittstellenlogik. Wenn du beispielsweise ein Objekt per DELETE löschst und keine weiteren Informationen nötig sind, ist 204 die elegante, semantisch korrekte Antwort. Alles andere ist unnötiger Datenbalast – und genau das wollen wir im Zeitalter der Microservices und Single Page Applications vermeiden.

204 vs. 200 vs. 304 – wo liegt der Unterschied?

Viele Entwickler – und leider auch einige SEOs – werfen HTTP-Statuscodes gerne durcheinander. 204 ist eben nicht einfach eine „leere 200“ und hat auch nichts mit dem 304 Not Modified zu tun. Wer den Unterschied nicht kennt, riskiert nicht nur technische Fehler, sondern auch Performance-Einbußen und SEO-Katastrophen.

Der 200 OK Statuscode signalisiert, dass eine Anfrage erfolgreich war und der Server Inhalte liefert. Das kann HTML, JSON, CSS oder sonst was sein – aber es kommt immer ein Body zurück. 204 hingegen sagt: Anfrage erfolgreich, aber es gibt nichts zurückzugeben. Kein Body. Kein DOM-Update. Kein sichtbarer Unterschied für den User, aber eine große Ersparnis an Bandbreite und Render-Zeit.

Der 304 Not Modified wiederum ist ein ganz anderes Biest. Er wird verwendet, wenn der Client über ein Caching-System verfügt und den Server fragt: „Hat sich etwas geändert?“ Der Server antwortet mit 304, wenn alles beim Alten ist – und der Client verwendet seine gecachten Daten. Auch hier gibt es keinen Body, aber der semantische Kontext ist komplett anders als bei 204.

Wer 204 mit 304 verwechselt oder aus Bequemlichkeit immer 200 zurückgibt, sabotiert sich selbst – technisch, performance-technisch und unter Umständen auch aus SEO-Sicht. Denn Suchmaschinen interpretieren Statuscodes sehr präzise. Und was wie eine Kleinigkeit aussieht, kann über Indexierung oder Ignorierung entscheiden.

Wann HTTP 204 sinnvoll ist – und wann nicht

HTTP 204 ist kein Allheilmittel für „ich weiß nicht, was ich antworten soll“. Es ist ein präzises Werkzeug, das in klar definierten Use Cases Sinn ergibt – und außerhalb davon eher für Irritation sorgt. Deshalb ist es entscheidend, den richtigen Moment für seine Anwendung zu kennen.

Typische Einsatzszenarien für HTTP 204 sind:

- DELETE-Requests, bei denen kein Rückgabewert erforderlich ist
- PUT- oder PATCH-Requests, bei denen der Client bereits alle Informationen hat
- Formularübermittlungen, bei denen ein UI-Element asynchron aktualisiert wird
- AJAX-Abfragen im Hintergrund, die lediglich Statusänderungen triggern

Wichtig: Ein 204 sollte niemals dann verwendet werden, wenn der Client neue Informationen erwartet. Wenn du beispielsweise eine neue Ressource erstellst (POST), ist ein 201 Created angebracht – inklusive Location-Header. Antwortest du stattdessen mit 204, ist das semantisch falsch und führt zu unklarer Logik im Frontend.

Ein weiterer häufiger Fehler: 204 bei GET-Requests. Technisch erlaubt, aber inhaltlich unsinnig. Denn ein GET-Request signalisiert: „Ich will etwas sehen.“ Wenn du darauf mit „Kein Inhalt“ antwortest, hast du den Request nicht verstanden. Das kann zu UI-Fehlern, SEO-Problemen und frustrierten Usern führen.

Merke: HTTP 204 ist ein skalpellartiges Werkzeug für spezifische Anwendungsfälle – kein Schweizer Taschenmesser für alle unklaren Zustände.

SEO und HTTP 204: Unsichtbar heißt nicht irrelevant

Und jetzt kommt der Teil, bei dem viele SEOs anfangen zu schwitzen: Was macht Google mit einem HTTP 204? Die kurze Antwort: Nichts. Und genau das ist das Problem.

Wenn ein Crawler auf eine Seite trifft, die mit 204 antwortet, wird die Seite nicht indexiert. Kein Inhalt = keine Indexierung. Punkt. Das ist zwar logisch – aber in der Praxis fatal, wenn du versehentlich eine öffentlich zugängliche URL mit 204 auslieferst. Google interpretiert das als „nicht relevant“ oder „nicht existent“ – und nimmt die Seite aus dem Index oder ignoriert sie komplett.

Deshalb gilt: HTTP 204 hat im öffentlichen Web, also bei dokumentenbasierten Seiten, nichts verloren. Landingpages, Blogposts, Produktseiten – all das braucht eine 200-Antwort mit Inhalten. Wer hier mit 204 antwortet, schneidet sich selbst aus dem organischen Spiel heraus.

Im API-Kontext hingegen ist 204 völlig unproblematisch. Google crawlt keine REST-Endpunkte oder Backend-Schnittstellen. Hier zählt Effizienz, nicht Sichtbarkeit. Und genau deshalb ist der Kontext entscheidend. HTTP 204 ist ein SEO-Killer – wenn er an der falschen Stelle eingesetzt wird.

Best Practice: Stelle sicher, dass alle crawlfähigen URLs mit 200 OK und validem HTML antworten. Verwende 204 ausschließlich für technische Endpunkte oder nicht-crawlbare AJAX-Calls. Und kontrolliere regelmäßig deine Logfiles und Statuscodes – sonst fliegen dir deine Rankings schneller um die Ohren, als du “Indexierung” buchstabieren kannst.

Best Practices für HTTP 204 – effizient, sauber, sinnvoll

Du willst HTTP 204 richtig einsetzen? Dann halte dich an diese Grundregeln. Kein Rocket Science, aber konsequente Umsetzung trennt die Profis vom Rest:

- Nur bei Non-Content-Interaktionen verwenden
  Nutze 204 ausschließlich, wenn keine neuen Informationen an den Client zurückgegeben werden müssen.
- Keine Body-Daten senden
  Der Response muss komplett leer sein – kein Body, kein HTML, kein JSON.
- Nur bei nicht-crawlbaren Endpunkten einsetzen
  Vermeide 204 bei öffentlich zugänglichen URLs, die indexiert werden sollen.
- Frontend-Logik klar trennen
  Stelle sicher, dass dein Frontend korrekt mit 204 umgehen kann – z.B. durch keine weiteren UI-Updates oder Reloads.

- Statuscode bewusst setzen, nicht per Default
  Viele Server-Frameworks antworten per Default mit 200 – du musst 204 explizit setzen.

Und noch ein Pro-Tipp: Wenn du 204 einsetzt, logge es. Denn beim Debugging ist ein leerer Response oft schwer zu erkennen. Ein sauberer Log-Eintrag mit Timestamp, Endpoint und Method hilft dir, Fehler schneller zu identifizieren – und zu vermeiden.

Fazit: Weniger ist manchmal mehr – aber nur, wenn du weißt, was du tust

HTTP 204 ist kein Gimmick. Es ist ein präzises Werkzeug für saubere Webkommunikation. Richtig eingesetzt, spart es Bandbreite, reduziert Ladezeiten und verbessert die User Experience – besonders in APIs, AJAX-Interaktionen und modernen Webarchitekturen. Falsch eingesetzt, verursacht es SEO-Super-GAUs, unverständliche UIs und Debugging-Alpträume.

Deshalb: Wenn du 204 verwendest, dann mit Verstand. Mach dir klar, was du dem Client sagen willst – und was nicht. Vermeide semantische Fehlkommunikation, schütze deine indexierbaren Seiten vor versehentlicher Unsichtbarkeit und nutze Statuscodes nicht als Notlösung, sondern als präzise Signale. HTTP ist eine Sprache – und 204 ist ein Satz. Einer ohne Worte, aber mit Wirkung.