

HTTP-Statuscodes: 200, 301, 404, 500 erklärt

PRAXISLEITFADEN | Stand: 27. Oktober 2026 | Kostenlos

1. 2xx: Erfolg

Vorgehen: 200 bedeutet erfolgreiche Antwort, 201 neu angelegte Ressource, 204 Erfolg ohne Body.

Beispiel: POST /api/items → 201 Created mit Location-Header.

Prüfkriterium: API-Vertrag passt zu Status und Response-Body.

Vermeiden: Bei 204 dennoch JSON-Body erwarten.

2. 3xx: Umleitung

Vorgehen: 301/308 stehen für permanente, 302/307 für temporäre Weiterleitung; 307/308 bewahren die HTTP-Methode.

Beispiel: GET /alt → 301 Location: /neu

Prüfkriterium: Redirect endet auf der gewünschten Ressource.

Vermeiden: Lange Redirect-Ketten oder Schleifen.

3. 4xx: Clientfehler

Vorgehen: 400 ungültige Anfrage, 401 nicht authentifiziert, 403 verboten, 404 nicht gefunden, 409 Konflikt, 429 zu viele Anfragen.

Beispiel: GET /api/private → 401 mit passender Authentifizierungsinformation.

Prüfkriterium: Client erhält einen verständlichen, nicht sensitiven Fehler.

Vermeiden: 401 und 403 verwechseln oder interne Details verraten.

4. 5xx: Serverfehler

Vorgehen: 500 unerwarteter Serverfehler, 502 Gatewayfehler, 503 temporär nicht verfügbar, 504 Gateway-Timeout.

Beispiel: 503 mit Retry-After bei geplanter Wartung.

Prüfkriterium: Monitoring und Logs erfassen Fehler ohne personenbezogene Details zu leaken.

Vermeiden: Alle Exceptions pauschal als 500 zurückgeben.

5. Diagnose mit curl

Vorgehen: Untersuche Status, Header und Weiterleitungen getrennt.

Beispiel: curl -I https://example.org
curl -L -I https://example.org/alt

Prüfkriterium: Erwarteter Endstatus und Redirect-Pfad sind nachvollziehbar.

Vermeiden: Nur Browseranzeige statt HTTP-Antwort prüfen.

Fachquellen

MDN: HTTP status – <https://developer.mozilla.org/en-US/docs/Web/HTTP/Reference/Status>

Onlineversion: <https://www.cyberknecht.de/wissen/cheatsheets/http-statuscodes-cheatsheet.html>