

2.5 HTTP-Statuscodes und Fehlermeldungen

HTTP-Statuscodes zeigen an, wie ein Server eine Anfrage verarbeitet hat.

Der Statuscode besteht aus drei Ziffern.

Beispiele:

- 200 – Anfrage erfolgreich
- 404 – Ressource nicht gefunden
- 500 – interner Serverfehler

Der Statuscode wird in der Response des Servers übertragen.

Beispiel:

```
HTTP/1.1 200 OK

{
  "id": 15,
  "name": "Max Mustermann"
}
```

“ Der Statuscode beschreibt das Ergebnis einer HTTP-Anfrage.

Lernziele

Nach dieser Seite solltest du erklären können:

- wie HTTP-Statuscodes aufgebaut sind
- was die Klassen 1xx bis 5xx bedeuten
- was wichtige Statuscodes aussagen
- wie sich 401 und 403 unterscheiden
- wie Fehler in API-Workflows behandelt werden
- wie HTTP-Fehler systematisch eingegrenzt werden

Statuscode-Klassen

Die erste Ziffer bestimmt die Statusklasse.

Klasse	Bedeutung
1xx	Information
2xx	Anfrage erfolgreich
3xx	Weiterleitung
4xx	Fehler in der Anfrage
5xx	Fehler auf der Serverseite

Merksatz:

“ 2xx bedeutet Erfolg, 4xx bedeutet meistens Clientfehler und 5xx bedeutet Serverfehler.

1xx - Information

Statuscodes der Klasse 1xx zeigen an, dass die Anfrage empfangen wurde und die Verarbeitung fortgesetzt wird.

Beispiele:

Code	Bedeutung
100	Continue
101	Switching Protocols

Diese Codes sind bei normalen API-Aufrufen weniger häufig sichtbar.

2xx - Erfolgreiche Anfrage

Statuscodes der Klasse 2xx zeigen an, dass eine Anfrage erfolgreich verarbeitet wurde.

Code	Bedeutung	Typisches Beispiel
200	OK	Daten erfolgreich abgerufen
201	Created	neue Ressource erstellt
202	Accepted	Anfrage angenommen, Verarbeitung folgt später
204	No Content	erfolgreich, aber keine Antwortdaten

200 OK

`200 OK` ist ein allgemeiner Erfolgsstatus.

Beispiel:

```
GET /users/15
```

Mögliche Response:

Status: 200 OK

```
{
  "id": 15,
  "name": "Max Mustermann",
  "active": true
}
```

👍 Die Anfrage wurde erfolgreich verarbeitet.

201 Created

`201 Created` wird häufig nach dem erfolgreichen Erstellen einer Ressource zurückgegeben.

Beispiel:

```
POST /users
```

Mögliche Response:

Status: 201 Created

```
{
  "id": 15,
  "name": "Max Mustermann"
}
```

Eine neue Ressource wurde erfolgreich erstellt.

202 Accepted

202 Accepted bedeutet:

“ Die Anfrage wurde angenommen, aber noch nicht vollständig verarbeitet.

Beispiel:

Ein umfangreicher Datenimport wird gestartet.

POST /imports

Status: 202 Accepted

Die Verarbeitung kann anschließend im Hintergrund des Zielsystems weiterlaufen.

Der Client sollte später prüfen, ob der Vorgang abgeschlossen wurde.

204 No Content

204 No Content bedeutet:

“ Die Anfrage war erfolgreich, aber die Response enthält keinen Body.

Typisches Beispiel:

DELETE /users/15

Status: 204 No Content

Der Client darf nicht automatisch erwarten, dass JSON-Daten zurückgegeben werden.

3xx - Weiterleitungen

Statuscodes der Klasse 3xx zeigen an, dass eine Ressource unter einer anderen Adresse erreichbar ist oder eine Weiterleitung erforderlich ist.

	Code	Bedeutung
	301	Moved Permanently
	302	Found
	304	Not Modified
	307	Temporary Redirect
	308	Permanent Redirect

Bei APIs können Weiterleitungen problematisch sein, wenn der Client sie nicht unterstützt oder dabei Authentifizierungsdaten verloren gehen.

4xx - Fehler in der Anfrage

Statuscodes der Klasse 4xx zeigen normalerweise an, dass die Anfrage fehlerhaft ist oder nicht ausgeführt werden darf.

Häufige Ursachen:

- falsche Daten
- fehlende Authentifizierung
- fehlende Berechtigung
- ungültiger Endpoint
- falsche HTTP-Methode
- ungültiges Datenformat
- zu viele Anfragen

400 Bad Request

400 Bad Request bedeutet:

“ Die Anfrage ist fehlerhaft und kann nicht verarbeitet werden.

Mögliche Ursachen:

- ungültiges JSON
- fehlender Pflichtwert
- falscher Datentyp
- fehlerhafte Syntax
- ungültiger Parameter

Beispiel:

```
{
  "name": "Max Mustermann",
  "active":
}
```

Der Wert für `active` fehlt.

Mögliche Response:

Status: 400 Bad Request

```
{
  "error": "Invalid request body"
}
```

401 Unauthorized

`401 Unauthorized` bedeutet meistens:

“ Die Authentifizierung fehlt oder ist ungültig.

Mögliche Ursachen:

- Token fehlt
- Token ist abgelaufen
- API-Key ist ungültig
- Zugangsdaten sind falsch
- Authentifizierungsheader ist fehlerhaft

Beispiel:

Authorization: Bearer ungültiges-token

Mögliche Response:

Status: 401 Unauthorized

Der Server weiß nicht sicher, wer der Client ist.

403 Forbidden

403 Forbidden bedeutet:

“ Die Identität ist bekannt, aber die Aktion ist nicht erlaubt.

Beispiel:

Ein Token darf Benutzer lesen, aber keine Benutzer löschen.

```
DELETE /users/15
```

Mögliche Response:

Status: 403 Forbidden

“ Der Client ist authentifiziert, besitzt aber nicht die benötigte Berechtigung.

401 und 403 im Vergleich

Statuscode	Bedeutung
401	Identität fehlt oder konnte nicht bestätigt werden
403	Identität ist bekannt, aber die Berechtigung fehlt

Merksatz:

“ 401 bedeutet: Wer bist du?
403 bedeutet: Du bist bekannt, darfst das aber nicht.

404 Not Found

404 Not Found bedeutet:

“ Die angeforderte Ressource wurde nicht gefunden.

Mögliche Ursachen:

- falscher Endpoint
- falsche ID
- Ressource wurde gelöscht
- API-Version ist falsch
- Schreibfehler in der URL

Beispiel:

```
GET /users/99999
```

Mögliche Response:

Status: 404 Not Found

```
{  
  "error": "User not found"  
}
```

405 Method Not Allowed

405 Method Not Allowed bedeutet:

“ Der Endpoint existiert, aber die verwendete HTTP-Methode ist dort nicht erlaubt.

Beispiel:

```
DELETE /reports
```

Der Endpoint erlaubt möglicherweise nur GET.

Mögliche Response:

Status: 405 Method Not Allowed

409 Conflict

409 Conflict bedeutet:

“ Die Anfrage steht im Konflikt mit dem aktuellen Zustand des Systems.

Beispiele:

- E-Mail-Adresse existiert bereits
- Benutzername ist schon vergeben
- Ressource wurde zwischenzeitlich geändert
- doppelter Datensatz soll erstellt werden

Beispiel:

```
POST /users
```

```
{
  "email": "max.mustermann@example.com"
}
```

Mögliche Response:

```
Status: 409 Conflict
```

```
{
  "error": "Email already exists"
}
```

415 Unsupported Media Type

415 Unsupported Media Type bedeutet:

“ Das übertragene Datenformat wird nicht unterstützt.

Mögliche Ursache:

Der Body enthält JSON, aber der Header fehlt oder ist falsch.

Falsch:

```
Content-Type: text/plain
```

Richtig:

```
Content-Type: application/json
```

422 Unprocessable Content

422 Unprocessable Content bedeutet:

“ Die Anfrage ist technisch lesbar, enthält aber fachlich ungültige Daten.

Beispiele:

- E-Mail-Adresse besitzt ein ungültiges Format
- Startdatum liegt vor dem Geburtsdatum
- Pflichtwert ist leer
- Wert liegt außerhalb des erlaubten Bereichs

Beispiel:

```
{  
  "email": "keine-gueltige-email"  
}
```

429 Too Many Requests

429 Too Many Requests bedeutet:

“ Der Client hat in kurzer Zeit zu viele Anfragen gesendet.

Dies wird als Rate Limit bezeichnet.

Mögliche Maßnahmen:

- Wartezeit einbauen
- Anzahl der Anfragen reduzieren
- mehrere Datensätze gesammelt übertragen
- Retry-Mechanismus verwenden
- `Retry-After`-Header auswerten

Beispiel:

Status: 429 Too Many Requests

Retry-After: 60

Bedeutung:

⚡ Vor einem neuen Versuch 60 Sekunden warten.

5xx - Serverfehler

Statuscodes der Klasse 5xx zeigen an, dass die Anfrage grundsätzlich empfangen wurde, aber der Server sie nicht erfolgreich verarbeiten konnte.

Mögliche Ursachen:

- interner Softwarefehler
- Datenbank nicht erreichbar
- überlasteter Server
- abhängiger Dienst ausgefallen
- Wartungsarbeiten
- Gateway- oder Proxyfehler

500 Internal Server Error

`500 Internal Server Error` ist ein allgemeiner Serverfehler.

Mögliche Ursachen:

- Programmfehler
- fehlerhafte Serverkonfiguration
- Datenbankfehler
- nicht behandelter Ausnahmefall

Der Client kann die Ursache meistens nicht direkt beheben.

502 Bad Gateway

502 Bad Gateway bedeutet:

“ Ein Gateway oder Proxy hat von einem nachgelagerten System eine ungültige Antwort erhalten.

Beispiel:

Client

|

v

Reverse Proxy

|

v

Anwendung nicht erreichbar

Mögliche Ursachen:

- Zielcontainer ist gestoppt
- falscher interner Port
- falscher DNS-Name
- Anwendung antwortet fehlerhaft
- Proxy-Konfiguration ist falsch

503 Service Unavailable

503 Service Unavailable bedeutet:

“ Der Dienst ist vorübergehend nicht verfügbar.

Mögliche Ursachen:

- Wartung
- Überlastung
- Serverstart
- abhängiger Dienst nicht verfügbar

- zu wenige Ressourcen

Ein späterer erneuter Versuch kann erfolgreich sein.

504 Gateway Timeout

504 Gateway Timeout bedeutet:

“ Ein Gateway oder Proxy hat nicht rechtzeitig eine Antwort vom Zielsystem erhalten.

Mögliche Ursachen:

- Zielsystem reagiert zu langsam
- Datenbankabfrage dauert zu lange
- Netzwerkproblem
- Timeout-Wert ist zu niedrig
- Anwendung hängt

Wichtige Statuscodes im Überblick

Code	Bedeutung	Typische Ursache
200	OK	Anfrage erfolgreich
201	Created	Ressource erstellt
202	Accepted	Verarbeitung erfolgt später
204	No Content	erfolgreich ohne Body
400	Bad Request	fehlerhafte Anfrage
401	Unauthorized	Authentifizierung fehlt
403	Forbidden	Berechtigung fehlt
404	Not Found	Ressource nicht gefunden
405	Method Not Allowed	falsche HTTP-Methode
409	Conflict	Datensatzkonflikt
415	Unsupported Media Type	falsches Datenformat
422	Unprocessable Content	fachlich ungültige Daten
429	Too Many Requests	Rate Limit erreicht
500	Internal Server Error	allgemeiner Serverfehler

Code	Bedeutung	Typische Ursache
502	Bad Gateway	nachgelagerter Dienst fehlerhaft
503	Service Unavailable	Dienst vorübergehend nicht verfügbar
504	Gateway Timeout	Zielsystem antwortet zu langsam

HTTP-Fehler und Netzwerkfehler

Nicht jeder Fehler besitzt einen HTTP-Statuscode.

Ein HTTP-Statuscode kann nur empfangen werden, wenn der Client den Server erreicht und eine HTTP-Response zurückkommt.

Beispiele ohne HTTP-Statuscode:

- DNS-Auflösung schlägt fehl
- Netzwerkverbindung besteht nicht
- Port ist geschlossen
- TLS-Zertifikat ist ungültig
- Verbindung wird zurückgesetzt
- Timeout ohne Serverantwort

Unterschied:

Fehlerart	Beispiel
HTTP-Fehler	Server antwortet mit 404
Netzwerkfehler	Server ist nicht erreichbar
DNS-Fehler	Hostname kann nicht aufgelöst werden
TLS-Fehler	Zertifikat wird nicht akzeptiert

“ Keine Response bedeutet auch keinen HTTP-Statuscode.

Response-Body auswerten

Der Statuscode zeigt nur die allgemeine Fehlerklasse.

Der Response-Body enthält häufig genauere Informationen.

Beispiel:

Status: 400 Bad Request

```
{  
  "error": "Missing required field",  
  "field": "email"  
}
```

Für die Fehlersuche sollten geprüft werden:

- Statuscode
- Fehlermeldung
- betroffenes Feld
- Zeitpunkt
- Request-ID
- Endpoint
- verwendete Methode

Fehlerbehandlung in n8n

Ein Workflow sollte nicht nur prüfen, ob eine Node ausgeführt wurde.

Er sollte auch bewerten:

- welcher Statuscode zurückgegeben wurde
- ob erwartete Daten vorhanden sind
- ob ein erneuter Versuch sinnvoll ist
- ob der Fehler dauerhaft oder vorübergehend ist
- ob ein Administrator informiert werden muss

Beispiel:

API-Anfrage

|

v

Statuscode prüfen

/ | \

2xx 4xx 5xx

| | |

v v v

weiter Fehler Retry

melden und melden

Dauerhafte und vorübergehende Fehler

Fehler	Typische Behandlung
400	Request korrigieren
401	Token oder Anmeldung prüfen
403	Berechtigungen prüfen
404	Endpoint oder ID prüfen
409	vorhandene Ressource prüfen
429	warten und erneut versuchen
500	Fehler protokollieren und melden
503	später erneut versuchen
504	Timeout und Zielsystem prüfen

Nicht jeder Fehler sollte automatisch wiederholt werden.

Beispiel:

Ein `400 Bad Request` wird durch denselben unveränderten Request nicht behoben.

Ein `503 Service Unavailable` kann nach einer Wartezeit verschwinden.

Retry

Retry bedeutet:

“ Eine fehlgeschlagene Anfrage wird erneut ausgeführt.

Retry ist sinnvoll bei vorübergehenden Fehlern:

- `429 Too Many Requests`
- `502 Bad Gateway`
- `503 Service Unavailable`
- `504 Gateway Timeout`
- kurzfristigem Netzwerkfehler

Retry ist normalerweise nicht sinnvoll bei:

- `400 Bad Request`
- `401 Unauthorized`
- `403 Forbidden`

- 404 Not Found

Diese Fehler benötigen zuerst eine Änderung an Anfrage, Zugang oder Berechtigung.

Wartezeit zwischen Wiederholungen

Anfragen sollten nicht sofort unbegrenzt wiederholt werden.

Beispiel:

1. Versuch sofort
2. Versuch nach 5 Sekunden
3. Versuch nach 15 Sekunden
4. Versuch nach 30 Sekunden
5. anschließend Administrator informieren

Eine schrittweise steigende Wartezeit wird als Backoff bezeichnet.

Vorteile:

- Zielsystem wird nicht zusätzlich überlastet
 - Rate Limits werden berücksichtigt
 - kurzfristige Störungen können sich beheben
 - Endlosschleifen werden vermieden
-

Logging

Bei einem Fehler sollten mindestens folgende Informationen protokolliert werden:

- Zeitpunkt
- Workflow
- betroffener Schritt
- HTTP-Methode
- Endpoint
- Statuscode
- Fehlermeldung
- betroffene Ressource
- Anzahl der Versuche
- Ergebnis der Fehlerbehandlung

Geheime Daten dürfen nicht vollständig im Log gespeichert werden.

Nicht protokollieren:

- Passwörter
- vollständige API-Keys

- vollständige Bearer-Tokens
- vertrauliche personenbezogene Daten

Praxisbeispiel: Benutzer anlegen

Ein n8n-Workflow sendet:

POST /users

Mögliche Ergebnisse:

Status	Bedeutung	Reaktion
201	Benutzer erstellt	Workflow fortsetzen
400	Daten fehlerhaft	Eingabedaten prüfen
401	Token ungültig	Zugangsdaten prüfen
403	keine Berechtigung	API-Rechte prüfen
409	Benutzer existiert bereits	vorhandenen Benutzer suchen
429	zu viele Anfragen	warten und erneut versuchen
503	Dienst nicht verfügbar	später erneut versuchen

Systematische Fehlersuche

Wenn eine API-Anfrage fehlschlägt:

1. Wurde überhaupt eine HTTP-Response empfangen?
2. Welcher Statuscode wurde zurückgegeben?
3. Welche Fehlermeldung steht im Response-Body?
4. Ist der Endpoint korrekt?
5. Ist die HTTP-Methode korrekt?
6. Ist der Request-Body gültig?
7. Ist der `Content-Type` korrekt?
8. Ist das Token vorhanden und gültig?
9. Besitzt das Konto ausreichende Rechte?
10. Existiert die angeforderte Ressource?
11. Wurde ein Rate Limit erreicht?
12. Ist der Fehler dauerhaft oder vorübergehend?
13. Kann der Request sicher erneut ausgeführt werden?
14. Muss ein Administrator informiert werden?

Wichtige Begriffe

Begriff	Bedeutung
Statuscode	Ergebnis einer HTTP-Anfrage
2xx	erfolgreiche Verarbeitung
3xx	Weiterleitung
4xx	Fehler in der Anfrage
5xx	Fehler auf der Serverseite
Rate Limit	Begrenzung der Anzahl von Anfragen
Retry	erneuter Ausführungsversuch
Backoff	steigende Wartezeit zwischen Versuchen
Response-Body	zusätzliche Antwort- oder Fehlerdaten
Timeout	Antwort wurde nicht rechtzeitig empfangen
Gateway	vermittelndes System zwischen Client und Server

Gesamtmerksatz

“ HTTP-Statuscodes zeigen das Ergebnis einer Anfrage. 2xx steht für Erfolg, 4xx für Fehler in Anfrage, Authentifizierung oder Berechtigung und 5xx für Fehler auf der Serverseite. Für eine saubere Fehlerbehandlung müssen Statuscode, Response-Body, Wiederholbarkeit und Logs gemeinsam ausgewertet werden.

Kontrollfragen

Was bedeutet ein Statuscode der Klasse 2xx?

Die Anfrage wurde erfolgreich verarbeitet.

Was ist der Unterschied zwischen 401 und 403?

Bei 401 fehlt eine gültige Authentifizierung. Bei 403 ist der Client bekannt, besitzt aber nicht die benötigte Berechtigung.

Was bedeutet 404?

Die angeforderte Ressource wurde nicht gefunden.

Was bedeutet 409?

Die Anfrage steht im Konflikt mit dem aktuellen Zustand des Systems.

Was bedeutet 429?

Der Client hat zu viele Anfragen gesendet und ein Rate Limit erreicht.

Was bedeutet 500?

Auf dem Server ist ein interner Fehler aufgetreten.

Was bedeutet 503?

Der Dienst ist vorübergehend nicht verfügbar.

Wann ist ein Retry sinnvoll?

Bei vorübergehenden Fehlern wie 429, 502, 503, 504 oder kurzfristigen Netzwerkproblemen.

Warum sollte ein 400-Fehler nicht unverändert wiederholt werden?

Weil die Anfrage selbst fehlerhaft ist und zuerst korrigiert werden muss.

Warum muss zusätzlich zum Statuscode der Response-Body geprüft werden?

Weil er häufig genauere Informationen über Ursache und betroffenes Feld enthält.

Quellen

- [IETF - RFC 9110: HTTP Semantics](#)
 - [MDN Web Docs - HTTP-Statuscodes](#)
 - [MDN Web Docs - 401 Unauthorized](#)
 - [MDN Web Docs - 403 Forbidden](#)
 - [MDN Web Docs - 429 Too Many Requests](#)
 - [n8n-Dokumentation - Error Handling](#)
-