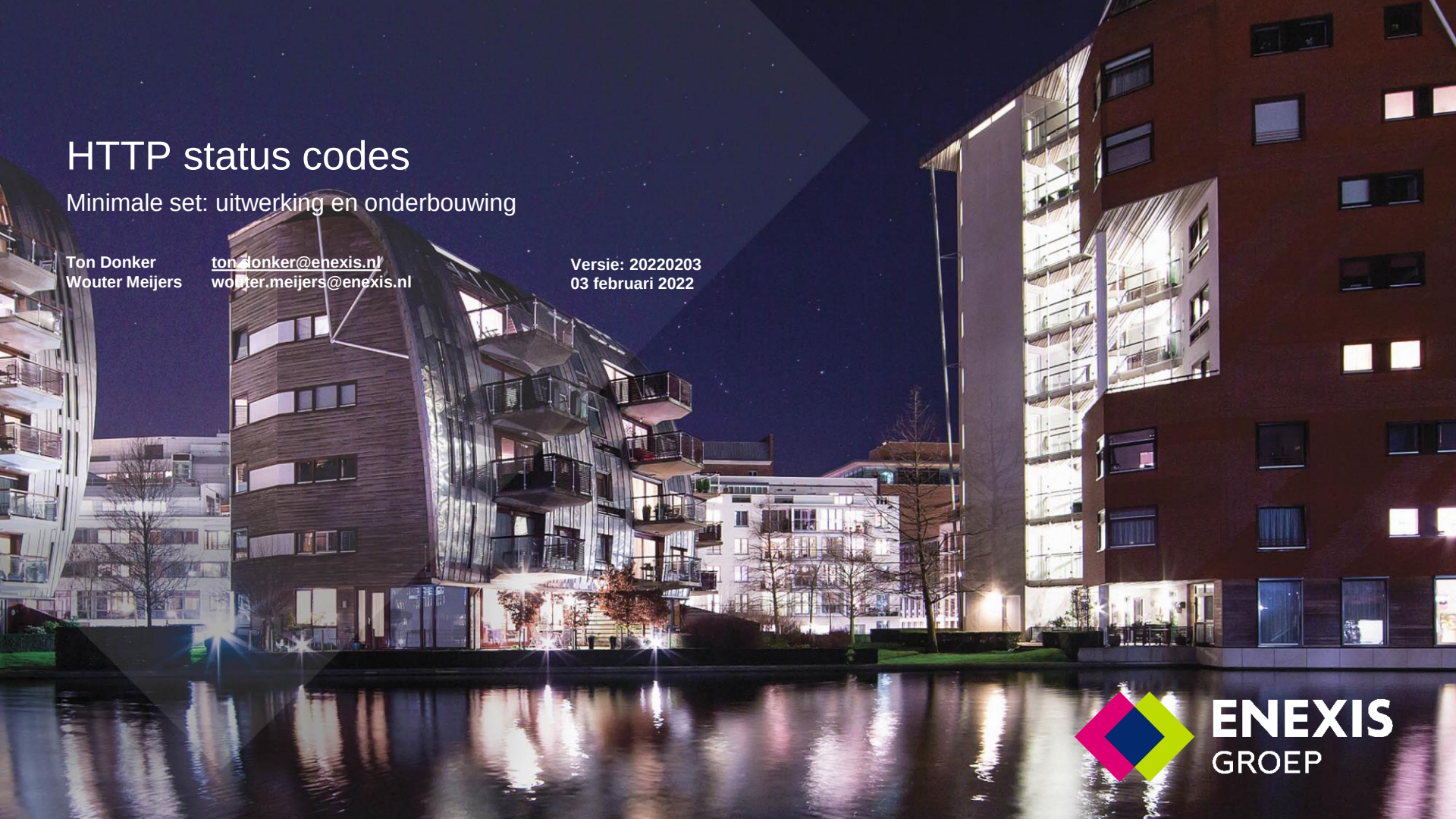

HTTP status codes

Minimale set: uitwerking en onderbouwing

Ton Donker ton.donker@enexis.nl
Wouter Meijers wouter.meijers@enexis.nl

Versie: 20220203
03 februari 2022

ENEXIS GROEP

Change history

Version	Changes	Document
20211216	Eerste release Ton Donker	HTTP-status-codes-min-set-20211216.pptx
20220203	Review door Wouter Meijers	HTTP-status-codes-min-set-20220203.pptx

Introductie

Zowel Geonovum[1] als DSO[2] beschrijven een verplichte set van HTTP status codes die minimaal moet worden ondersteund in een API definitie. Er zijn echter wat verschillen zichtbaar tussen DSO en Geonovum in de beschrijving en toepassing van deze HTTP codes. Voorbeeld: de 404 wordt wel benoemd door Geonovum[1], niet door DSO[2], en zien we de 412 als verplichte code wel bij DSO en niet bij Geonovum. Ook in de issue discussies op Github van Kennisplatform APIs[3] zien we veel interpretatieverschillen en meningen over het juiste gebruik van HTTP status codes.

Doelstelling van dit document is het vaststellen van een minimale set van HTTP status codes die door een API moeten worden ondersteund en gespecificeerd. Merk op dat dezelfde HTTP method (bijv. ‘POST’) afhankelijk van de toepassing verschillende status codes kan teruggeven: een 201 in geval POST een resource genereert, een 200 in het geval de POST als complexe query wordt uitgevoerd. Het is dus zinvol om een overzicht te geven van codes per operatie.

In dit document zijn naast NL-overheids APIs en Github discussies ook internationale ‘best practices’ onderzocht en is er intensief gekeken naar API richtlijnen van bijv. Zalando en Bol.com.

1. https://docs.geostandaarden.nl/api/API-Strategie-ext/#http-status-codes
2. https://iplo.nl/digitaal-stelsel/aansluiten/standaarden/api-en-uri-strategie/ (API B-49)
3. https://github.com/Geonovum/KP-APIs en https://github.com/VNG-Realisatie

Inhoudsopgave

Om de minimale set te kunnen vaststellen, moeten ook de afzonderlijke status codes en hun functies worden onderzocht. Wanneer treedt een 401 – *Unauthorized* op en wanneer een 403 – *Forbidden*? Moet een zoekopdracht zonder resultaat worden beantwoord met een 204 – *No Content* of een 404 - *Not Found*? Deze vragen worden in dit document per thema geadresseerd en beantwoord.

Allereerst worden het vraagstuk rondom toepasbaarheid van codes 404, 200 en 204 behandeld. Dit vraagstuk wordt afgesloten met een concrete toepassingsmatrix voor dit drietal codes.

Vervolgens worden de 401 en 403 codes toegelicht en met elkaar vergeleken. Aansluitend een aanbeveling voor het gebruik van dit tweetal. Codes 400 en 422 vereisen eveneens nadere toelichting en uitwerking en advies voor gebruik.

Het vaststellen van de minimale set is het resultaat van:

1. Onderzoek naar specifieke functie van afzonderlijke foutcodes;
2. Analyse van ontwerprichtlijnen van NL-overheids APIs;
3. (Inter)nationale ‘best practices’;
4. Statistisch onderzoekje naar foutcodes in NL-overheids APIs – zie appendices;

404 vs 200[ ] vs 204 (1)

Over het gebruik van 404, vs 200[ ] vs 204 is de internet opinie verdeeld. Bij de werkgroepen van de NL-overheid heeft men voor de standaardwerking gekozen een 200[ ] (lege collectie) terug te geven indien geen zoekresultaten worden gevonden via query parameters. Voorbeeld[1]:

GET /kadastraalonroerendezaken/zakelijkgerechtigden?type=erfpachter - geeft 200 'OK' terug met lege collectie

Stelling is dat als er niets gevonden wordt, er geen 404 als fout mag worden teruggegeven. Geen zoekresultaten hoeft niet per definitie als foutief te worden aangemerkt, vandaar de 200 *OK* status code met lege collectie. Er wordt immers via query paramaters op een collectie gefilterd. Een resource identifier is idealiter geen query parameter, maar een path parameter, in feite het deel van de URL zonder query parameters. Een mogelijk zoekresultaat kan leeg zijn, als er geen waarden worden gevonden die aan de filtercriteria voldoen. Dit lijkt ook internationaal de meest gangbare 'best practice'[2] te zijn. De 404 wordt toegepast in foutsituaties wanneer de path parameter of collectie die wordt opgevraagd, überhaupt niet bestaat. In dit geval 'bestaat de URI niet' – er wordt een onjuiste unieke resource identificatie gebruikt en dit kan als foutief worden aangemerkt. Voorbeeld[3]:

GET /kadastraalonroerendezaken/1234 - geeft 404 'Not Found' – resource met id '1234' bestaat niet

1. Bronnen:
   https://github.com/VNG-Realisatie/Haal-Centraal-BRK-bevragen/issues/231
   https://github.com/VNG-Realisatie/Haal-Centraal-BRK-bevragen/issues/464
   https://github.com/VNG-Realisatie/Haal-Centraal-BAG-bevragen/issues/245
   HAL voorbeeld met lege '_embedded': https://github.com/VNG-Realisatie/Haal-Centraal-BAG-bevragen/issues/88
2. Interessant artikel dat de toepasbaarheid van deze 3 codes beschrijft : https://apihandyman.io/empty-lists-http-status-code-200-vs-204-vs-404/
3. Bronnen:
   https://github.com/VNG-Realisatie/Haal-Centraal-BRK-bevragen/issues/231
   https://github.com/VNG-Realisatie/Haal-Centraal-BRK-bevragen/issues/801

404 vs 200[ ] vs 204 (2)

De 204 - *No Content* verdient in dit opzicht ook aandacht en wordt als alternatief voor de 200[ ] aangedragen indien geen zoekresultaten worden gevonden.

Opgemerkt dient te worden dat het gebruik van status code 204 summier wordt toegepast in de NL-overheids APIs[1] en in alle gevallen exclusief wordt gebruikt door de DELETE methode. Zie hiervoor ook Appendix A: Status codes vergelijkingsmatrix.

Onderzoek naar de 204[2] levert op dat er hier met name gaat om een specialistische status code waarin aan de user agent of browser wordt medegedeeld de view van het aangepaste document niet te wijzigen, zodat het document beschikbaar blijft voor verdere aanpassingen Alleen de meta informatie wordt aangepast. Dit in tegenstelling tot de 205, waarin een reset juist nodig is[3]. De 204 is ook wel bekend als: *don't change the view response code*[4].

Best practice is de 204 te gebruiken voor de DELETE methode, omdat er bij verwijdering van een resource nooit content terug gegeven kan/hoeft worden. Voor de PUT en PATCH geldt hetzelfde indien geen content terug gegeven kan/hoeft te worden. Dit kan worden getriggerd door de 'Prefer' header parameter in de request.

Voor de GET met query parameters kan de 204 vermeden worden. Vanuit client perspectief biedt dit het voordeel dat in dit geval altijd een 200 met content (lege of gevulde collectie) wordt teruggegeven.

1. https://github.com/VNG-Realisatie/Haal-Centraal-BAG-bevragen/issues/255
2. Bronnen:
   https://apihandyman.io/empty-lists-http-status-code-200-vs-204-vs-404/
   https://datatracker.ietf.org/doc/html/rfc7231#section-6.3.5
   https://stackoverflow.com/questions/65961464/404-vs-204-on-get-put-delete-methods ('stay with the same view')
3. https://benramsey.com/blog/2008/05/http-status-204-no-content-and-205-reset-content/
4. https://newbedev.com/what-is-the-proper-rest-response-code-for-a-valid-request-but-an-empty-data & https://developer.mozilla.org/en-US/docs/Web/HTTP/Status/204

404 vs 200[ ] vs 204 (3)

Voorbeelden[1]:

Methode	URL	Status	Response
GET	/users	200	[rob, jan]
GET	/users/rob	200	rob
GET	/uzerz-bestaat-niet	404	Not Found
GET	/users/arthur	404	Not Found
GET	/users/arthur/friends?name=rob	404	Not Found
GET	/users?name=arthur	200	[ ] or { }
DELETE	/users/rob	204	No Content

1. https://stackoverflow.com/questions/11746894/what-is-the-proper-rest-response-code-for-a-valid-request-but-an-empty-data

401 vs 403 (1)

De internet opinie is het erover eens dat de beschrijving *Unauthorized* misleidend is voor status code 401[1]. Code 401 geeft aan dat er niet geauthentiseerd kan worden, bijvoorbeeld omdat de API-key niet bekend of onjuist is[2]. Het gaat hier dus om *authenticatie* i.p.v. *autorisatie*. Ook met de header parameter *Authorization* wordt bedoeld dat het hier gaat om verstrekken van authenticatie credentials[3].

Momenteel is er een discussie gaande of de naamgeving moet worden aangepast. Roy Fielding geeft aan dat een 401 zowel voor *unauthorized* of *unauthenticated* kan worden gebruikt[4].

Google CloudAPIs[5] beschrijft de 401 als UNAUTHENTICATED:

401 UNAUTHENTICATED	*Request not authenticated due to missing, invalid, or expired OAuth token*

De 401 te reserveren voor authenticatie fouten lijkt de best practice te zijn.

Ook is er veel discussie over de toepassing van status code 403 – *Forbidden*. De gangbare tendens is dat de geauthenticeerde user (dus voorbij de 401 fout situatie) geen rechten heeft, of niet geautoriseerd is tot een bepaalde resource[6].

1. Bronnen:
   - https://datatracker.ietf.org/doc/html/rfc7235#section-3.1 '401 (Unauthorized) … it lacks valid authentication credentials…
   - https://github.com/Geonovum/KP-APIs/issues/103
   - https://apihandyman.io/hands-off-that-resource-http-status-code-401-vs-403-vs-404/
2. https://github.com/VNG-Realisatie/Haal-Centraal-BAG-bevragen/issues/105
3. https://en.wikipedia.org/wiki/List_of_HTTP_header_fields
4. https://github.com/httpwg/http-core/issues/745
5. https://cloud.google.com/apis/design/errors#error_codes
6. https://github.com/Geonovum/KP-APIs/blob/master/API-strategie-extensies/ext-error-handling.md

401 vs 403 (2)

Google CloudAPIs[1] beschrijft de 403 code als PERMISSION_DENIED:

403 PERMISSION_DENIED	*Client does not have sufficient permission. This can happen because the OAuth token does not have the right scopes, the client doesn't have permission, or the API has not been enabled*

De internet standaard RFC7231[2] omschrijft 403 met het volgende zinsfragment:

> *…If authentication credentials were provided in the request, the server considers them insufficient to grant access…*

Identieke toepassing zien we bij de NL overheid[3]: de 403 wordt gekoppeld aan de oauth scope. Het oauth token is wel geldig (anders was een 401 teruggegeven), maar de 403 geeft aan dat niet geautoriseerd kan worden in de gegeven scope. In lijn met de bovenstaande definitie van Google CloudAPIs.

De 403 te reserveren voor autorisatie fouten lijkt te best practice te zijn.

Vermeld dient te worden dat de omschrijving van de 403 momenteel wordt bediscussieerd in de HTTP core werkgroep[4]. Aanleiding is het 403 voorbeeld in RFC7807[5]. In dit voorbeeld wordt een 403 teruggegeven omdat de GET request verboden is, gezien de huidige staat van de resource en niet vanwege een autorisatiefout. Een mogelijke aanpassing van de 403 omschrijving is nog niet officieel vastgesteld[6].

1. https://cloud.google.com/apis/design/errors#error_codes
2. https://datatracker.ietf.org/doc/html/rfc7231#section-6.5.3
3. https://github.com/Geonovum/KP-APIs/issues/103
4. https://github.com/httpwg/http-core/issues/218
5. https://datatracker.ietf.org/doc/html/rfc7807#section-3
6. https://github.com/httpwg/http-core/commit/2ceecdbaa90c0888348adeba8c4ff64f336d2072

400

De 400 response code geeft aan dat het request een syntactische fout bevat. Merk op dat foutieve *waarden* van query parameters hier ook toe behoren. Voorbeeld[1]:

GET /adresseerbareobjecten?bbox=12&oppervlakte[min]=-1&oppervlakte[max]=-1

In werkelijkheid kan dit niet, want de oppervlakte is cf business rules een natuurlijk getal ‘tussen 1 en 999999’. Dus hoewel deze query geen syntactische fouten bevat, maar wel semantisch: nl. de inhoud van de parameterwaarde, moet een 400 status code worden teruggegeven. Nog een voorbeeld [2]:

GET /adresseerbareobjecten?bbox=12&oppervlakte[min]=50&oppervlakte[max]=10

In dit voorbeeld is oppervlakte[min] > oppervlakte[max] wat in werkelijkheid niet kan. Op basis van deze business rule moet hier een 400 code worden teruggegeven.

Indien een onjuiste waarde voor een query parameter of combinatie ervan wordt gegeven, dan dient dit te worden afgekeurd met een 400 status code. Dit geldt bijv. ook voor paginering. Wanneer een niet bestaande pagina wordt opgevraagd, dient een 400 te worden teruggegeven, omdat de waarde voor query page parameter onjuist is. Voorbeeld[3] – waarbij page ’12’ niet bestaat:

GET /adressen?pandIdentificatie=0014100022188747&page=12&pageSize=50

1. https://github.com/VNG-Realisatie/Haal-Centraal-BAG-bevragen/issues/477
2. https://github.com/VNG-Realisatie/Haal-Centraal-BAG-bevragen/issues/476
3. Bronnen:
   https://github.com/VNG-Realisatie/Haal-Centraal-BAG-bevragen/issues/376
   https://github.com/VNG-Realisatie/Haal-Centraal-BAG-bevragen/issues/270

400 – 500

De 400 en 500 status codes worden ook wel gezien als generieke foutcodes voor foutscenario's waarvoor geen andere 4xx of 5xx code toepasbaar is[1]. In geval van twijfel kunnen deze codes worden gebruikt. De 400 en 500 worden internationaal gezien als *fallback option*.

Microsoft beschrijft het als volgt[2] expliciet voor de 400 status code:

> *400 BadRequest indicates that the request could not be understood by the server. BadRequest is sent when no other error is applicable, or if the exact error is unknown or does not have its own error code.*

En de 500 status code:

> *500 InternalServerError indicates that a generic error has occurred on the server*

De 400 en 500 te reserveren voor generieke 'default' fouten lijkt de best practice te zijn.

Dit betekent dat deze codes toepasbaar zijn op elke HTTP-methode. Uitgangspunt is wel om zo optimaal mogelijk gebruik te maken van de 4XX range om de foutsituatie zo precies mogelijk te beschrijven.

1. Bronnen:
   https://restapilinks.com/wp-content/uploads/2021/04/REST-API-Design-Rulebook.pdf (pagina 30)
   https://restapilinks.com/wp-content/uploads/2021/04/RESTful_Web_APIs.pdf (pagina 306)
   https://datatracker.ietf.org/doc/html/rfc7231#section-6.5.1
   https://www.mnot.net/blog/2017/05/11/status_codes
2. https://docs.microsoft.com/en-us/dotnet/api/system.net.httpstatuscode?view=net-6.0#System_Net_HttpStatusCode_BadRequest

422

De 422 status code - *Unprocessable Entity* – wordt steeds vaker toegepast in met name internationale API definities[1]. Oorspronkelijk werd deze status code gespecificeerd in RFC4918[2] en was deze voorbehouden voor HTTP extensie WebDAV. Dit staat voor: *Web Distributed Authoring and* Versioning. Dit is een uitbreiding op het HTTP protocol voor webgebaseerd en verdeeld auteurschap met versiebeheer[3]*.* Dit suggereert een specifiek gebruik, en dat betekent dat de 422 niet zou behoren tot de standaard set van HTTP status codes.

De huidige tendens is dat 422 uit de extensie van WebDAV wordt gehaald en opgenomen wordt in de herziene HTTP specificaties. De *title* van 422 wordt daarbij gewijzigd naar: *Unprocessable Payload*[4]. Dit betekent dat de 422 toegepast kan worden op semantische validatie van de request payload[5], te denken valt aan de PUT, POST (search) en PATCH. In de specificatie van de PATCH wordt de 422 ook expliciet genoemd[6].

Merk op dat de 422 in de richtlijnen de NL-overheid wel als verplicht wordt aangemerkt[7], maar we zien deze code niet vaak geïmplementeerd in de APIs

De 422 te reserveren voor semantische validatie van de request payload lijkt de best practice te zijn.

1. Bronnen:
   https://stackoverflow.com/questions/16133923/400-vs-422-response-to-post-of-data
   https://www.codetinkerer.com/2015/12/04/choosing-an-http-status-code.html (indrukwekkende ‘decision chart’)!
2. https://datatracker.ietf.org/doc/html/rfc4918
3. https://nl.wikipedia.org/wiki/WebDAV
4. https://datatracker.ietf.org/doc/html/draft-ietf-httpbis-semantics-05#section-9.5.20
5. https://pawelpluta.com/why-should-you-return-http-422/
6. https://datatracker.ietf.org/doc/html/rfc5789#section-2.2
7. https://docs.geostandaarden.nl/api/API-Strategie-ext/#http-status-codes

Minimale set foutcodes (1)

Method	200	201	204	400	401	403	404	422	500	503
GET collectie (& query params)	√			√	√	√	√		√	√
GET & path params	√			√	√	√	√		√	√
PUT create		√	opt	√	√	√	√	√	√	√
PUT update	√		opt	√	√	√	√	√	√	√
PATCH	√		opt	√	√	√	√	√	√	√
DELETE	opt		√	√	√	√	√		√	√
POST create		√		√	√	√	√	√	√	√
POST search	√			√	√	√	√	√	√	√

Met *opt* wordt aangegeven dat de foutcode in bijzonder gevallen – optioneel – kan worden gebruikt.
Deze bijzondere situaties worden in de volgende slide uitgewerkt.

Minimale set foutcodes (2)

- De 400 en 500 kunnen als generieke foutcode worden gezien en zijn derhalve toepasbaar op alle methods;
- PUT kan een resource creëren[1]. Dit verdient echter niet de voorkeur. Indien sprake is van creatie, dan kan een 201 met response payload of een 204 zonder payload in worden teruggegeven. In beide gevallen moet de Location header informatie bevatten over de gecreëerde resource.
- De 204 kan ook worden teruggegeven bij de update (PUT of PATCH) van een resource, indien geen response payload teruggegeven hoeft te worden.
- De 200 moet worden teruggegeven door de DELETE als er behoefte is aan een response payload met informatie over de verwijderde resource[2].
- De 422 wordt strikt gebruikt voor semantische validatie van de payload.

1. Bronnen:
   https://opensource.zalando.com/restful-api-guidelines/#put
   https://httpwg.org/specs/rfc7231.html#PUT
   https://iplo.nl/digitaal-stelsel/aansluiten/standaarden/api-en-uri-strategie/ (2.5.4)
2. https://httpwg.org/specs/rfc7231.html#DELETE

Appendix A: Status codes vergelijkingsmatrix

	POST		GET		PUT		PATCH		DELETE	
	Zalando	DSO/GEO	Zalando	DSO/GEO	Zalando	DSO/GEO	Zalando	DSO/GEO	Zalando	DSO/GEO
200 OK	√	√	√	√	√	√	√	√	√	√
201 Created	√	√			√					
204 No Content					√	√	√	√	√	√
400 Bad Request	√		√		√		√		√	
401 Unauthorized	√		√		√		√		√	
403 Forbidden	√		√		√		√		√	
404 Not Found	√		√	√	√	√	√	√	√	√
405 Method Not Allowed	√	√	√		√	√	√	√	√	√
406 Not Acceptable	√		√		√		√		√	
409 Conflict	√	√			√	√	√	√	√	
412 Precondition Failed					√		√		√	
415 Unsupported Media Type	√				√		√		√	
422 Unprocessable Entity										
429 Too Many Requests	√		√		√		√		√	
500 Internal Server Error	√		√		√		√		√	
503 Service Unavailable	√		√		√		√		√	

Appendix B: NL-overheids APIs en status codes (1)

API	Org	GET		PUT	PATCH	POST		DELETE
		Collectie + query params	Resource – path param			Creatie	Search	
BAG API Huidige Bevragingen[1]	Kadaster	200, 400, 401, 403, 406, 500, 503	200, 400, 401, 403, 404, 406, 500, 503					
BAG API Individuele Bevragingen[2]	Kadaster	200, 400, 401, 403, 405, 406, 412, 500, 503, default	200, 400, 401, 403, 404, 405, 406, 412, 500, 503, default				200, 400, 401, 403, 405, 406, 412, 415, 500, 503	
Terugmelding API[3]	Kadaster	200, 204, 400, 401	200, 400, 401, 404			200, 400, 401		
BRT API[4]	Kadaster	200, 400, 401, 406, 503	200, 400, 401, 404, 406, 503				200, 400, 401, 406, 503	
Bevraging Ingeschreven Persoon[5]	VNG	200, 400, 401, 403, 406, 500, 501, 503	200, 400, 401, 403, 404, 406, 500, 501, 503					

1. Basisregistratie Adressen en Gebouwen: https://developer.overheid.nl/apis/00000001802327497000-kadaster-bag-current
2. Basisregistratie Adressen en Gebouwen: https://developer.overheid.nl/detail/00000001802327497000-kadaster-bag-individual/production/specification
3. Terugmelding: https://developer.overheid.nl/detail/kadaster-terugmeldingen/production/specification
4. Basisregistratie Topografie: https://developer.overheid.nl/detail/kadaster-brt/production/specification
5. https://developer.overheid.nl/detail/vng-bevraging-ingeschreven-persoon/production/specification

Appendix B: NL-overheids APIs en status codes (2)

API	Org	GET		PUT	PATCH	POST		DELETE
		Collectie + query params	Resource - path param			Creatie	Search	
Authorisatie component[1]	VNG	200, 400, 401, 403, 406, 409, 410, 415, 429, 500	200, 401, 403, 404, 406, 409, 410, 415, 429, 500	200, 400, 401, 403, 404, 406, 409, 410, 415, 429, 500	200, 400, 401, 403, 404, 406, 409, 410, 415, 429, 500			204, 401, 403, 404, 406, 409, 410, 415, 429, 500
Besluitregistratie component[2]	VNG	200, 400, 401, 403, 406, 409, 410, 415, 429, 500	200, 401, 403, 404, 406, 409, 410, 415, 429, 500	200, 400, 401, 403, 404, 406, 409, 410, 415, 429, 500	200, 400, 401, 403, 404, 406, 409, 410, 415, 429, 500	201, 400. 401, 403, 406, 409, 410, 415, 429, 500		204, 401, 403, 404, 406, 409, 410, 415, 429, 500
Notificatie routeringscomponent[3]	VNG	200, 401. 403, 406, 409, 410, 415, 429, 500	200, 401, 403, 404, 406, 409, 410, 415, 429, 500	200, 400, 401, 403, 404, 406, 409, 410, 415, 429, 500	200, 400, 401, 403, 404, 406, 409, 410, 415, 429, 500			204, 401, 403, 404, 406, 409, 410, 415, 429, 500
Centrale OIN Raadpleegvoorziening[4]	Logius	200, 400, 404, 405, 406, 500, 503	200, 400, 404, 405, 406, 500, 503					

1. https://developer.overheid.nl/detail/vng-ac/production/specification
2. https://developer.overheid.nl/detail/vng-brc/production/specification
3. https://developer.overheid.nl/detail/vng-nrc/production/specification
4. https://developer.overheid.nl/detail/logius-cor/production/specification

Appendix C: Bol.com & Zalando APIs

API	Org	GET		PUT		PATCH	POST		DELETE
		path param	Collectie query	Update	Create		Creatie	Search	
v6 – Retailer API[1]	Bol.com	200, 400, 404	200, 400	202, 400			202, 400	200, 400, 404	202, 400
v7 – Advertising API[2]	Bol.com	200, 400, 404	200, 400, 404	202, 400			202, 400		202, 400
V7 – Shared API[3]	Bol.com	200, 400	200, 400					200, 400	
Orders API[4]	Zalando	200, 403, 404, 429, default				204, 400, 403, 404, 429, default			
Products API[5]	Zalando	200, 409, default			204, 400, 403, 404, 409, 429 default				
Product Attributes[6]	Zalando	200, default							

1. https://api.bol.com/retailer/public/apispec/v6
2. https://api.bol.com/retailer/public/apispec/Advertiser%20API%20-%20v7
3. https://api.bol.com/retailer/public/apispec/Shared%20API%20-%20v7
4. https://developers.merchants.zalando.com/docs/openapi/orders.html
5. https://developers.merchants.zalando.com/docs/openapi/products.html
6. https://developers.merchants.zalando.com/docs/openapi/product-attributes.html

Appendix D: gateway vs applicatie codes

Gateway	Applicatie
401 Unauthorized	200 OK
403 Forbidden	204 No Content
429 Too many requests	400 Bad Request
500 Internal Server Error	404 Not found
503 Service Unavailable	406 Not Acceptable
	409 Conflict
	410 Gone
	412 Precondition Failed
	415 Unsupported Media Type
	422 Unprocessable Entity

Dat de gateway deze codes genereert betekent niet dat de applicatie ze niet ook kan genereren. Het is meer een aandachtspunt dat, als de API de codes NIET gebruikt, een gateway dit dus WEL kan doen en ze daardoor in de OAS moeten voorkomen omdat die compleet moet zijn.

Bronnen:
https://github.com/VNG-Realisatie/Haal-Centraal-BAG-bevragen/issues/105
https://github.com/VNG-Realisatie/Haal-Centraal-BRK-bevragen/issues/231
https://github.com/VNG-Realisatie/Haal-Centraal-BAG-bevragen/issues/255

Appendix E: waarden voor *Title*

Status code	Omschrijving	Title
200	OK	The request succeeded
201	Created	The request successfully created a resource
204	No Content	The server successfully fulfilled the request but returns no response body
400	Bad Request	The server cannot or will not process the request due to something that is perceived to be a client error
401	Unauthorized	User failed to authenticate properly
403	Forbidden	Authorization failed due to insufficient permissions
404	Not found	The specified resource does not exist
405	Method Not Allowed	The server does not support the requested HTTP method
406	Not Acceptable	The server does not support the client-request media type to return the response payload
409	Conflict	The request conflicts with current state of the target resource
415	Unsupported Media Type	The server does not support the request payload's media type
422	Unprocessable Entity	The server is unable to process the contained instructions
429	Too many requests	Too many requests. Blocked due to rate limiting
500	Internal Server Error	An internal server error has occurred
503	Service Unavailable	Service unavailable

Appendix F: Handige links

- Mooi overzicht HTTP status codes en uitleg: https://developer.mozilla.org/en-US/docs/Web/HTTP/Status
- Ook API gateway specifieke statuscodes (401, 403, 429, 503) zijn onderdeel van de API specificatie: https://github.com/VNG-Realisatie/Haal-Centraal-BAG-bevragen/issues/255
- Het is onmogelijk een combinatie van meerdere fouten mee te geven, bijv een 406 en 400: https://github.com/VNG-Realisatie/Haal-Centraal-BAG-bevragen/issues/503
- Foutafhandeling feature file van VNG: https://github.com/VNG-Realisatie/Haal-Centraal-common/blob/master/features/foutafhandeling.feature
- HTTP status codes en bijbehorende RFCs: https://webconcepts.info/concepts/http-status-code/
- HTTP status code decision chart: https://www.codetinkerer.com/2015/12/04/choosing-an-http-status-code.html
- Github http decision diagram: https://github.com/for-GET/http-decision-diagram
- Meeste recente HTTP semantics: https://httpwg.org/http-core/draft-ietf-httpbis-semantics-latest.html


SAMEN WERKEN WE AAN EEN
BETROUWBARE EN DUURZAME
ENERGIEVOORZIENING
VOOR VANDAAG EN VOOR DE
TOEKOMST.
For more information, please contact:
CST-Integratie@enexis.nl
ENEXIS
GROEP