

API ontwerprichtlijnen

Titel : API ontwerprichtlijnen
Datum : 27-09-2023
Versie : 4.0
Status : Vastgesteld
Opdrachtgever : MFF

Inhoudsopgave

Colofon 3

Versiegeschiedenis..... 3

Verspreiding..... 5

Gerefereerde documenten 6

Begrippen- en afkortingenlijst..... 7

1 Inleiding..... 8

2 API ontwerprichtlijnen 9

2.1 Standaarden voor API ontwerp 9

01 - API strategie voor de Nederlandse overheid [2] 9

02 - Gestandaardiseerde informatie modellen voor definitie van de API..... 9

2.2 Energiesector specifieke richtlijnen API ontwerp..... 10

01 - Definitie van het koppelvlak 10

02 - Versiebeheer..... 10

03 - Query parameter voor sorteren 11

04 - Query parameter voor zoeken 11

05 - Fouten en uitzonderingssituaties 11

06 - Gebruik van ODATA 12

07 - Definitie attributen Info object..... 13

08 - Gebruik van REST operaties..... 15

09 - Complex search..... 15

10 - HTTP status codes..... 16

11 - Gebruik van JSON Schema Specification 17

12 - Quality of Service specificatie 17

13 - Definiëren van datum/tijd 17

14 - Definiëren van een string 18

15 - Autorisatie 18

16 - Definiëren van veldnamen..... 19

17 - HAL voor het gebruik van hypermedia controls 19

18 - Expanderen van gelinkte resources..... 20

19 - Representatie op maat 20

20 - HTTP headers 20

21 - Enumeraties..... 21

22 - Gebruik van gewone persoonsgegevens in de URL van REST API's..... 22

23 - API specificatie 23

24 - Gebruik van xxxOf constructies 24

Bijlage A: API Strategie Algemeen (Nederlandse API Strategie I) 25

Bijlage B: REST-API Design Rules (Nederlandse API Strategie IIa)..... 26

Bijlage C: API Designrules Extensions (Nederlandse API Strategie IIb)..... 27

Colofon

Versiegeschiedenis

Versienummer	Status	Markering/wijzigingen
0.1	Concept	Initiële versie opgesteld door de NEDU TC werkgroep API strategie.
0.2	Concept	Addendum toegevoegd.
0.3	Concept	Review op v0.2 verwerkt.
0.4	Concept	Initiële versie “NEDU API strategie en ontwerprichtlijnen” opgesplitst in twee documenten: NEDU API strategie; NEDU API ontwerprichtlijnen.
0.5	Concept	Security standaarden toegevoegd.
0.6	Concept	Review op v0.5 verwerkt.
0.9	Concept	Laatste versie van API strategie voor de Nederlandse overheid toegevoegd. Terugkoppeling uit praktijk toegevoegd: - Definieer voor strings in de response op een API call alleen de maximale lengte; - Richtlijn voor waarborgen van eenduidige interpretatie van datum en tijd; - Indien het gehanteerde referentie model een definitie heeft voor veldnamen, dan prevaleert deze definitie boven richtlijn API-26 uit ASNO.
0.91	Concept	Review SI's verwerkt.
0.92	Concept	Review API werkgroep verwerkt.
0.93	Concept	BAG als gestandaardiseerd informatie model opgenomen.
0.94	Concept	HAL-standaard, expand en fields parameters opgenomen.
0.95	Ter vaststelling	Review API werkgroep verwerkt.
1.0	Vastgesteld	Versie na vaststelling in ALV NEDU van 16 december 2020.
1.1	Ter goedkeuring	Header parameters opgenomen (richtlijn nummer 20).
1.9	Ter vaststelling	Versie ter vaststelling door ALV.
2.0	Vastgesteld	Versie na vaststelling in ALV NEDU van 26 mei 2021.
2.1	Vastgesteld	ID03 en ID04 aangepast: parameters voorzien van een underscore prefix. ID07 aangepast: 'x-date' vervangen door 'x-releaseDate' en nieuw voorbeeld Info Object opgenomen. ID20 aangepast: cardinaliteit (verplicht of optioneel) van de HTTP headers is context afhankelijk. ID21 toegevoegd: gebruik van enumeraties in REST API's. ID22 toegevoegd: privacy aspecten voor gebruik van gewone persoonsgegevens in de URL van een REST API. Toelichting opgenomen bij de query parameter sort (ID03) en search (ID04) over het gebruik van de underscore als prefix.
2.2	Concept	ID05 aangepast: RFC7807 als basis voor standaard foutmeldingsformaten ID10 aangepast: specificeren van de minimale set van HTTP status codes
2.2	Definitief	Review werkgroep verwerkt en als versie voor werkgroep opgeleverd
2.3	Concept	ID23 richtlijnen API specificatie toegevoegd.

Versienummer	Status	Markering/wijzigingen
		ID24 richtlijnen gebruik van xxxOf constructies toegevoegd ID05 richtlijnen RFC7807 aangevuld Template gewijzigd in MFF BAS
2.9	Ter vaststelling	Na verwerken review WG API-strategie
3.0	Vastgesteld	Versie na vaststelling in MFF BAS ALV van 10 november 2022
3.1	Concept	ID01 aangepast: API documentatie in Engels ID02: NEN 5825 standaard voor adresnotatie toegevoegd ID07: voorbeelden aangepast naar Engels ID20 aangepast: HTTP header definities
3.9	Ter vaststelling	Na verwerken review WG API-strategie
4.0	Vastgesteld	Versie na vaststelling in MFF BAS ALV van 27 september 2023

Verspreiding

Naam	Datum	Versie
NEDU Technische Commissie	17-12-2019	0.6
NEDU Technische Commissie	10-02-2020	0.9
API werkgroep	10-02-2020	0.91
NEDU TC, SI's, API werkgroep	20-05-2020	0.92
API werkgroep	09-09-2020	0.93
API werkgroep	30-09-2020	0.94
NEDU	01-10-2020	0.95
NEDU	16-12-2020	1.0
API werkgroep	31-03-2021	1.1
NEDU Technische Commissie	31-03-2021	1.1
NEDU Technische Commissie	12-04-2021	1.1
NEDU ALV	11-05-2021	1.9
NEDU ALV	26-05-2021	2.0
API werkgroep, NEDU TC	08-11-2021	2.1
NEDU ALV	01-12-2021	2.1
API werkgroep	09-03-2022 (concept)	2.2
API werkgroep	04-05-2022 (definitief)	2.2
API werkgroep	13-06-2022 (concept)	2.3
MFF BAS API werkgroep	24-08-2022 (ter vaststelling)	2.9
CoP Secure Architecture	28-10-2022 (ter vaststelling)	2.9
MFF BAS ALV	10-11-2022 (vastgesteld)	3.0
MFF BAS API werkgroep	09-05-2023 (concept)	3.1
MFF BAS API werkgroep	31-05-2023 (ter vaststelling)	3.9
CoP Secure Architecture	02-06-2023 (ter vaststelling)	3.9
MFF BAS ALV	27-09-2023 (vastgesteld)	4.0

Gerefereerde documenten

Nummer	Omschrijving	Datum	Versie	Auteur
1.	NEDU EDSN Ontwerpkeuzes	15-04-2021	3.9	NEDU/EDSN
2.	API strategie voor de Nederlandse overheid:			
	• API Strategie Algemeen (Nederlandse API Strategie I)	20-12-2021	Handreiking	Geonovum e.a.
	• REST-API Design Rules (Nederlandse API Strategie IIa)	09-07-2020	1.0 Vastgesteld (Logius Standaard)	Geonovum e.a.
	• API Designrules Extensions (Nederlandse API Strategie IIb)	13-10-2021	Handreiking	Geonovum e.a.
3.	TC021 Vastlegging datum/tijd conventies	28-09-2011	1.0	NEDU TC
4.	TC028 Veldlengtes in berichten	28-10-2015	1.0	NEDU TC
5.	NEDU API strategie	16-12-2020	1.0	NEDU API werkgroep
6.	NEDU API URI richtlijnen	16-12-2020	1.0	NEDU API werkgroep
7.	W3C DTF Note ISO 8601	15-09-1997	N.v.t.	W3C
8.	API Strategie Digitaal Stelsel Omgevingswet: • https://aandeslagmetdeomgevingswet.nl/digitaal-stelsel/aansluiten/standaarden/api-en-uri-strategie/	26-03-2020	2.0 Vastgesteld	A.J. Sloos e.a.

Begrippen- en afkortingenlijst

Begrip	Omschrijving
API	Application Programming Interface
ASNO	API strategie voor de Nederlandse overheid
JSON	JavaScript Object Notation
ODATA	Open Data Protocol
OAS	Open API Specification
OAuth	Open Authorization
REST	REpresentational State Transfer
QoS	Quality of Service specificatie
WFS	Web Feature Service
HAL	Hypertext Application Language

1 Inleiding

De huidige generatie applicaties worden volgens een service georiënteerde architectuur gerealiseerd. Het doel hierbij is de flexibiliteit in dienstverlening te vergroten. Het ontsluiten van deze services wordt gerealiseerd door webservices en sinds enige tijd ook door REST API's¹.

Binnen de context van service georiënteerde architectuur is het ontwerp van webservices en API's van groot belang. Een slecht ontworpen webservice of API kan de ontwikkeling van applicaties, die deze services aanroepen,odeloos complex maken. Vanuit een business perspectief kan een slecht ontworpen webservice of API een negatief effect hebben op flexibiliteit van business processen. Daarentegen kunnen goed ontworpen webservices en API's de ontwikkelsnelheid van applicaties en de flexibiliteit van business processen verbeteren.

Voor het ontwerpen en ontwikkelen van webservices heeft de voormalige NEDU TC, in nauwe samenwerking met EDSN, de ontwerpkeuzes voor webservices [1] opgesteld. De partijen in de Nederlandse energiemarkt, verenigd binnen de NEDU, hebben de behoefte uitgesproken om eveneens richtlijnen te definiëren voor het ontwerpen van API's. In opdracht van de voormalige Technische Commissie NEDU (TC) zijn door een werkgroep de volgende documenten opgesteld:

1. API strategie [5];
2. API ontwerprichtlijnen (dit document);
3. API URI richtlijnen [6].

Uitgangspunt voor de API ontwerprichtlijnen is de *API-Strategie voor de Nederlandse overheid* [2]. Daarnaast zijn de energiesector specifieke ontwerprichtlijnen vastgelegd in paragraaf **Fout! Verwijzingsbron niet gevonden.** van de API ontwerprichtlijnen.

1. Waar in dit document over API's wordt gesproken, worden REST API's verstaan.

2 API ontwerprichtlijnen

2.1 Standaarden voor API ontwerp

Dit zijn de standaarden die door andere organen zijn vastgesteld en die richtinggevend zijn voor de API's die door de energiesector worden ontworpen en ontwikkeld.

ID	01
Titel	01 - API strategie voor de Nederlandse overheid [2]
Strategie	Bij het ontwerpen en ontwikkelen van API's is de <i>API strategie voor de Nederlandse overheid</i> (ASNO) [2] richtinggevend.
Toelichting	Deze API strategie is door de MFF bestempeld als algemeen uitgangspunt voor het ontwerpen en ontwikkelen van API's. Energiesector specifieke ontwerprichtlijnen zijn opgenomen in paragraaf Fout! Verwijzingsbron niet gevonden..
Onderbouwing	De API strategie van de Nederlandse overheid is behoorlijk doordacht/volwassen. Tevens bestaat de mogelijkheid dat de energie sector ook gebruik gaat maken van overheids API's.
Afwijking	Niet alle service interacties zijn geschikt voor een REST-API. Bij het nemen van een besluit om een REST-API te ontwikkelen wordt overwogen of een REST-API geschikt is voor het specifieke doel.

ID	02
Titel	02 - Gestandaardiseerde informatie modellen voor definitie van de API
Strategie	Voor de definitie van het koppelvlak (de API) zijn de gestandaardiseerde informatie modellen zoals IEC CIM, ebIX UML Model en ebIX EFET ENTSO-E Role Model leidend voor naamgeving van resources en attributen.
Toelichting	Deze gestandaardiseerde modellen worden door energiesector bestempeld als de standaard voor de definitie van assets en marktprocessen: • IEC Common Information Model (CIM); • ebIX UML Model for the European Energy Market; • OASIS Universal Business Language (UBL); • CMF Markt Model; • ebIX EFET ENTSO-E Harmonised Electricity Market Role Model; • NEN 5825 standaard voor adresnotatie; • BAG (Basisregistratie Adressen en Gebouwen). Dit overzicht van modellen is niet eindig en mag worden aangevuld met andere gestandaardiseerde modellen indien vereist voor de definitie van het koppelvlak.
Onderbouwing	Door het hanteren van deze ontwerprichtlijn volgt de definitie van de API de (internationale) standaarden IEC CIM en ebIX®.
Afwijking	Niet elke informatie behoefte wordt afgedekt door de standaarden. Eigen uitbreidingen van een bestaande gestandaardiseerde modellen heeft de voorkeur (in plaats van zelf verzinnen). Deze uitbreidingen kunnen teruggegeven worden aan de werkgroepen die de gestandaardiseerde modellen onderhouden.

2.2 Energiesector specifieke richtlijnen API ontwerp

Dit zijn de richtlijnen die opgesteld zijn op basis van *best practices* en *lessons learned* en zijn richtinggevend voor de API's die door de energiesector worden ontworpen en ontwikkeld. Deze richtlijnen zijn aanvullend en/of prevalerend op de *API strategie voor de Nederlandse overheid* (ASNO) [2].

ID	01
Titel	01 - Definitie van het koppelvlak
Strategie	Definitie van het koppelvlak (de API) in het UK-Engels.
Toelichting	Documentatie (zoals <i>description</i> en <i>example</i>) in de API definitie in het Engels en de definitie van het koppelvlak energiesector API's (resource en attributen) in het Engels.
Onderbouwing	ASNO schrijft Nederlands voor (API-04). Ontwerprichtlijn voor de energiesector voor de definitie van het koppelvlak is UK-Engels omdat deze taal beter aansluit bij (internationale) standaarden zoals IEC CIM en ebIX®.
Afwijking	Interne APIs: keuze aan bedrijf.

ID	02
Titel	02 - Versiebeheer
Strategie	API's zijn altijd voorzien van een versie
Toelichting	Regels voor versionering: • major: Niet backward compatible, daardoor grote impact voor de service consumer; • minor: Wel backward compatible, beperkte impact voor de service consumer; • patch: Wel backward compatible, zonder impact voor service consumer. Is bedoeld voor kleine fixes en mag zonder verdere specifieke berichtgeving uitgerold worden. Welk versie-kenmerk in de naamgeving van de API wordt opgenomen, is beschreven in de <i>NEDU API URI richtlijnen</i> [6].
Onderbouwing	Afspraken welke versie(s) in productie is/zijn en of er sprake is van een transitieperiode tussen de versies worden, in tegenstelling tot de richtlijnen uit ASNO (API Principe 4.4/API-20), niet als een generieke richtlijn voor de energiesector opgelegd. Daarvoor in de plaats worden per API afspraken gemaakt en vastgelegd waarbij de voorkeur uit gaat naar scenario 2: <u>Scenario 1: big bang</u> Een nieuwe versie van de API vervangt de productieversie, er is geen sprake van transitieperiode.

ID	02
	<u>Scenario 2: transitie</u> Nieuwe versies van de API en productieversie worden gedurende een transitieperiode ondersteund. Het aantal versies die gelijktijdig in productie mogen zijn is gelimiteerd tot drie óf zoals voorgeschreven in de invoeringsstrategie van de betreffende MFF Topic.
Afwijking	

ID	03
Titel	03 - Query parameter voor sorteren
Strategie	Gebruik 'sort' of '_sort' voor sorteren.
Toelichting	Conform ASNO API-31.
Onderbouwing	Ontwerprichtlijn voor de energiesector voor de definitie van het koppelvlak is Engels omdat deze taal beter aansluit bij (internationale) standaarden zoals IEC CIM en ebIX®.
Afwijking	Om verwarring of "botsingen" met resource-properties te voorkomen wordt een underscore als prefix geadviseerd (_sort).

ID	04
Titel	04 - Query parameter voor zoeken
Strategie	Gebruik 'search' of '_search' voor full-text zoeken.
Toelichting	Conform ASNO API-32.
Onderbouwing	Ontwerprichtlijn voor de energiesector voor de definitie van het koppelvlak is Engels omdat deze taal beter aansluit bij (internationale) standaarden zoals IEC CIM en ebIX®.
Afwijking	Om verwarring of "botsingen" met resource-properties te voorkomen wordt een underscore als prefix geadviseerd (_search).

ID	05
Titel	05 - Fouten en uitzonderingssituaties
Strategie	API's dienen fouten en uitzonderingssituaties uniform en volgens standaarden af te handelen.
Toelichting	Foutmeldingen dienen te zijn gespecificeerd conform het gestandaardiseerde foutmeldingsformaat van RFC 7807 (https://www.rfc-editor.org/rfc/rfc7807), zoals voorgeschreven in API-46 in ASNO. Voor iedere foutmelding zijn de volgende attributen verplicht: • type: bevat een URI referentie naar het type fout; • title: generieke titel voor het type fout; • status: de HTTP status code;

ID	05
	Default waarde 'about:blank' mag gebruikt worden indien geen URI referentie voor attribuut 'type' gegeven kan worden. Voorbeeld: <pre>{ "type": "https://datatracker.ietf.org/doc/html/rfc7231#section-6.5.3", "title": "Forbidden" "status": 403 }</pre> Naast de bovengenoemde verplichte attributen specificeert RFC7807 een aantal optionele attributen: • detail: extra gedetailleerde informatie van de fout; • instance: URI van de aanroep die de fout heeft veroorzaakt. RFC7807 staat toe dat de foutmelding kan worden uitgebreid met context specifieke attributen om extra informatie mee te sturen in het antwoord naar de consumer. Voorbeeld hiervan is het meesturen van een zgn. 'errorCode' door de betreffende applicatie welke refereert aan een <i>business rule</i> van deze applicatie waardoor het ingediende verzoek niet kan worden afgehandeld. Fouten en uitzonderingssituaties dienen conform de volgende richtlijnen afgehandeld te worden: 1. De foutmelding dient abstract en functioneel geformuleerd te zijn zodat iedere afnemer dit begrijpt; 2. De (functionele) foutmelding moet in de payload (body) van het bericht staan; 3. De foutmelding mag in geen geval (technische) implementatiedetails (bijv. stacktrace) bevatten.
Onderbouwing	Hiermee wordt een consistent en duidelijk gedrag bij foutmeldingen mogelijk gemaakt.
Afwijking	

ID	06
Titel	06 - Gebruik van ODATA
Strategie	API's dienen te worden gespecificeerd in OAS
Toelichting	In principe geen ondersteuning voor ODATA in de NEDU API strategie.
Onderbouwing	Een uitgewerkte use-case zal bepalen welk type API het beste voldoet. Voor API's waar datasets op vele manieren opgevraagd kunnen worden is bijvoorbeeld ODATA een mogelijk optie. ODATA of WFS houden als aparte standaard, niet dwingen in OAS/API vorm.

ID	06
Afwijking	

ID	07
Titel	07 - Definitie attributen Info object
Strategie	Elke API (vanaf OpenAPI 3.0 en hoger) dient te zijn voorzien van een Info object met informatie over de API.
Toelichting	De volgende attributen dienen opgenomen te zijn in het Info object: <pre>openapi: 3.0.3 info: title: description: termsOfService: contact: name: email: license: name: url: version: x-releaseDate:</pre> Het attribuut <i>version</i> specificeert het versienummer van de API zoals die wordt opgenomen in de URI (URI-component <versie>). De opbouw is de letter “v” gevolgd door het volledige versienummer van de API. Het attribuut <i>x-releaseDate</i> specificeert de datum (ISO 8601 - Wikipedia) waarop de REST API is vrijgegeven voor gebruik. Betreft een specificatie-extensies en begint daarom met een x- (zie https://swagger.io/docs/specification/openapi-extensions/).
Onderbouwing	Deze velden geven context aan de API voor gebruikers. <u>Voorbeeld JSON:</u>

ID	07
	<pre> openapi: 3.0.3 info: title: connection-details description: # API Version - 2.0.1 ##### Generated by ECDM Toolkit version 2.7.1 This API offers functionality for the retrieval of connection details ## Changelog ## 1.0.11 (8/15/2022) * FIX - Filter method attribute 'type' renamed to 'modelName' * FIX - Relation 'Assets to PowerElectronicsConnection' changed to 'Assets to PowerSystemResource' (conform CIM) ## 1.0.10 (7/14/2022) * CHG - In the enum list in case of an 422 IMMUTABLE_FIELD_CHANGED added termsOfService: 'https://www.example.com/terms' contact: name: EDSN url: https://edsn.nl email: servicedesk@edsn.nl license: name: APACHE2.0 url: https://www.apache.org/licenses/LICENSE-2.0 version: 2.0.1 x-releaseDate: 8/15/2022 </pre> Voorbeeld in de Swagger Editor: connection-details 2.0.1 OAS3 API Version - 2.0.1 Generated by ECDM Toolkit version 2.7.1 This API offers functionality for the retrieval of connection details Changelog 1.0.11 (8/15/2022) - FIX - Filter method attribute 'type' renamed to 'modelName' - FIX - Relation 'Assets to PowerElectronicsConnection' changed to 'Assets to PowerSystemResource' 1.0.10 (7/14/2022) - CHG - In the enum list in case of an 422 IMMUTABLE_FIELD_CHANGED added Terms of service EDSN - Website Send email to EDSN APACHE2.0

ID	07
	Voor de formattering van teksten van attribuut 'description' dient de CommonMark syntax gebruikt te worden conform de OAS specificatie. Voor CommonMark syntax zie https://spec.commonmark.org/0.27/
Afwijking	

ID	08
Titel	08 - Gebruik van REST operaties²
Strategie	De REST operaties POST, GET, PUT, PATCH en DELETE worden gebruikt in de context waar ze voor bedoeld zijn.
Toelichting	POST: Wordt gebruik als <i>create</i> voor resources en voegt een nieuwe resource toe aan de collectie. GET: Wordt gebruikt als <i>read</i> om een resource op te vragen van de server. Data wordt alleen opgevraagd en niet gewijzigd. PUT: Wordt gebruikt als <i>update</i> om een specifieke resource te vervangen. Is óók een <i>create</i> wanneer de resource op aangegeven identifier/URI nog niet bestaat. PATCH: Wordt gebruikt als <i>update</i> om een bestaande resource gedeeltelijk bij te werken. Het verzoek bevat de gegevens die gewijzigd of toegevoegd moeten worden aan de betreffende resource. DELETE: Wordt gebruikt als <i>delete</i> om een specifieke resource te verwijderen.
Onderbouwing	Dit zijn de meest gangbare operaties.
Afwijking	Privacy gevoelige informatie mag niet meegestuurd worden in de GET URI string. Gebruik POST indien er privacy gevoelige informatie wordt meegestuurd.

ID	09
Titel	09 - Complex search
Strategie	Gebruik POST uitgebreide zoekopdracht (complex search)
Toelichting	Voor een uitgebreid zoekopdracht (complex search) mag gebruik gemaakt van een POST. Voor het gebruik hiervan is consensus in de markt.
Onderbouwing	ASNO API-36 maakt een uitzondering op regel API-03 voor geometrische queries. API-37 maakt een uitzondering op regel API-03 voor geometrische queries in combinatie van andere query parameters. Echter, het toestaan van een POST met query parameters in de JSON body voor een complex search moet los gezien kunnen worden van een geometrie query.
Afwijking	

2. Alhoewel deze richtlijn ook in de API strategie voor de Nederlandse overheid is beschreven, is bewust deze richtlijn met eigen bewoordingen opgenomen in deze ontwerprichtlijn.

ID	10																																																																																																			
Titel	10 - HTTP status codes																																																																																																			
Strategie	Definiëren van de minimale set van HTTP status codes die door een REST API moeten worden gespecificeerd en ondersteund.																																																																																																			
Toelichting	Minimale set van HTTP status codes per REST operatie: <table><tr><th>REST operatie</th><th>200</th><th>201</th><th>204</th><th>400</th><th>401</th><th>403</th><th>404</th><th>422</th><th>500</th><th>503</th></tr><tr><td>GET collectie (& query params)</td><td>√</td><td></td><td></td><td>√</td><td>√</td><td>√</td><td>√</td><td></td><td>√</td><td>√</td></tr><tr><td>GET & path params</td><td>√</td><td></td><td></td><td>√</td><td>√</td><td>√</td><td>√</td><td></td><td>√</td><td>√</td></tr><tr><td>PUT create</td><td></td><td>√</td><td>opt</td><td>√</td><td>√</td><td>√</td><td>√</td><td>√</td><td>√</td><td>√</td></tr><tr><td>PUT update</td><td>√</td><td></td><td>opt</td><td>√</td><td>√</td><td>√</td><td>√</td><td>√</td><td>√</td><td>√</td></tr><tr><td>PATCH</td><td>√</td><td></td><td>opt</td><td>√</td><td>√</td><td>√</td><td>√</td><td>√</td><td>√</td><td>√</td></tr><tr><td>DELETE</td><td>opt</td><td></td><td>√</td><td>√</td><td>√</td><td>√</td><td>√</td><td></td><td>√</td><td>√</td></tr><tr><td>POST create</td><td></td><td>√</td><td></td><td>√</td><td>√</td><td>√</td><td>√</td><td>√</td><td>√</td><td>√</td></tr><tr><td>POST search</td><td>√</td><td></td><td></td><td>√</td><td>√</td><td>√</td><td>√</td><td>√</td><td>√</td><td>√</td></tr></table> - Een 400 en 500 kunnen als generieke status codes worden gezien en zijn derhalve toepasbaar op alle REST operaties; - Een 422 status code wordt strikt gebruikt voor semantische validatie van de payload. Met <i>opt</i> wordt aangegeven dat de foutcode in bijzonder gevallen – optioneel – kan worden gebruikt: - PUT kan een resource creëren, dit verdient echter niet de voorkeur. Indien er toch 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; - Een 204 status code kan worden teruggegeven bij een update (PUT of PATCH) van een resource indien geen response payload teruggegeven hoeft te worden; - Een 200 status code moet worden teruggegeven bij een DELETE operatie indien er behoefte is aan een response payload met informatie over de verwijderde resource.	REST operatie	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	√			√	√	√	√	√	√	√
REST operatie	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	√			√	√	√	√	√	√	√																																																																																										
Onderbouwing	Zowel Geonovum als DSO 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, niet door DSO, 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 zien we veel interpretatieverschillen en meningen over het juiste gebruik van HTTP status codes. Het specificeren van een minimale set van HTTP status codes die voor een REST operatie moeten worden gebruikt geeft de API ontwerper duidelijkheid bij het ontwerpen en ontwikkelen van de REST API.																																																																																																			
Afwijking																																																																																																				

ID	11
Titel	11 - Gebruik van JSON Schema Specification
Strategie	Elke API call met een body dient te zijn voorzien van een JSON Schema Specification.
Toelichting	ASNO schrijft het opnemen van een JSON Schema in de API niet verplicht voor (API-23). De ontwerprichtlijn voor de energiesector verplicht wel tot het opnemen van een JSON Schema in de API.
Onderbouwing	Door het opnemen van het JSON Schema in de API kan een API call vroegtijdig, reeds in de gateway, worden afgewezen indien de API call niet voldoet aan de syntax. Hiermee wordt voorkomen dat andere applicaties worden belast met het afhandelen van syntactisch niet-valide API calls. Eveneens stelt het opnemen van het JSON Schema de API 'gebruiker' in staat om zijn eigen berichten al te valideren voordat de implementatie van de API beschikbaar is.
Afwijking	

ID	12
Titel	12 - Quality of Service specificatie
Strategie	Elke API heeft een Quality of Service specificatie (QoS).
Toelichting	Aspecten om op te nemen in de QoS: • Integration latency; • Openingstijden; • Beschikbaarheid; • Capaciteit; • Betrouwbaarheid.
Onderbouwing	De <i>API-Strategie voor de Nederlandse overheid</i> beschrijft geen Quality of Service specificatie en moet als informatie toegevoegd worden aan de OAS-specificatie en in de business servicebeschrijving. De informatie die vereist is voor een ontwikkelaar die wil aansluiten op het koppelvlak van de API dient in de OAS-specificatie te zijn opgenomen. Business relevante informatie dient in de betreffende business servicebeschrijving te zijn opgenomen.
Afwijking	

ID	13
Titel	13 - Definieren van datum/tijd
Strategie	De notatie van datum/tijd dient compleet en eenduidig te zijn.
Toelichting	Datum/tijd notatie moet conform W3C DTF Note ISO 8601 [7]. Hierdoor wordt eenduidige interpretatie van datum/tijd internationaal gewaarborgd. Advies:

ID	13
	- Hanteer UTC en met de "Z" als time zone designator; - Specificeer de tijd met seconden en milliseconden. Indien milliseconden niet bekend zijn, gebruik dan "000". Voorbeeld: 2020-02-07T14:55:42.000Z.
Onderbouwing	De notatie van datum/tijd conform de W3C DTF Note ISO 8601 [7] is vastgelegd in TC021 [3].
Afwijking	

ID	14
Titel	14 - Definiëren van een string
Strategie	In lijn van TC028 [4] dient de definitie van type string in de API call te zijn voorzien van een lengte attribuut of van een minimale/maximale lengte attribuut waarbij de minimale lengte 1 moet zijn ³ .
Toelichting	<pre>type: string minLength: 1 maxLength: 10</pre> of <pre>type: string length: 10</pre>
Onderbouwing	Om uit te sluiten dat een API call een lege string bevat en daardoor mogelijk tot uitval kan leiden bij service providers of service consumer. Het advies is om voor strings in de response op een API call in ieder geval de maximale lengte te definiëren voor de service consumer .
Afwijking	

ID	15
Titel	15 - Autorisatie
Strategie	Autorisatie op API's dient ingericht te zijn met OAuth2.0 en bij voorkeur met OpenID Connect
Toelichting	OAuth2 wordt gebruikt voor autorisatie op API's. Met een Access token wordt toegang verleent tot een Resource server(API). IETF standaard OAuth2: https://tools.ietf.org/html/rfc6749 OpenID Connect is voor het authentifieren van gebruikers. OpenID Connect een is standaard bovenop OAuth2. OAuth2 beschrijft echter niet hoe authenticatie van gebruikers plaats moet vinden. Daarom wordt OpenID Connect geadviseerd. IETF

3. Minimale lengte 0 is niet toegestaan.

ID	15
	standaard OpenID Connect: https://openid.net/specs/openid-connect-core-1_0.html PKCE is een standaard om de Authorization code flow mogelijk te maken voor public clients. RFC: https://tools.ietf.org/html/rfc7636 OpenID Connect en OAuth2 bevatten meerdere flows en grant types . Het volgende wordt geadviseerd: • Authorization code flow met PKCE wanneer een gebruiker inlogt; • Client credentials grant voor applicatie naar applicatie autorisatie. JWT Access tokens worden geadviseerd om validatie te kunnen doen op Resource server(API) niveau zonder afhankelijkheid van een autorisatie server. IETF draft JWT Access token: https://tools.ietf.org/html/draft-ietf-oauth-access-token-jwt-02
Onderbouwing	Het beveiligen van gevoelige data binnen API's is een essentieel. OpenID Connect en OAuth2 zijn API beveiligingsstandaarden die wereldwijd erkend en ondersteund zijn. Door gebruik te maken van markt standaarden kunnen partijen die toegang willen tot API's hier eenvoudig en veilig op aansluiten.
Afwijking	

ID	16
Titel	16 - Definiëren van veldnamen
Strategie	Hanteer de definitie van veldnamen zoals die in het referentie model worden voorgeschreven.
Toelichting	Het definiëren van veldnamen in camelCase is voorgeschreven in richtlijn API-26 uit ASNO. Echter, indien het gehanteerde referentie model een definitie heeft voor veldnamen, dan prevaleert deze definitie boven richtlijn API-26 uit ASNO.
Onderbouwing	Door het hanteren van deze ontwerprichtlijn volgt de definitie van de veldnaam de (internationale) standaarden IEC CIM en ebIX®.
Afwijking	

ID	17
Titel	17 - HAL voor het gebruik van hypermedia controls
Strategie	Voor het realiseren van hypermedia controls wordt Hypertext Application Language (HAL) gebruikt.
Toelichting	De HAL-standaard is ontworpen voor het bouwen van API's waarin clients door resources navigeren door (hyper)links te volgen. De HAL-representatie voor JSON dient te worden gebruikt indien gerelateerde resources (relaties) worden teruggegeven als hyperlinks.

ID	17
Onderbouwing	De HAL-standaard voor het gebruik van hypermedia controls is vastgelegd in de <i>API Strategie Digitaal Stelsel Omgevingswet [API-H01]</i> [8].
Afwijking	

ID	18
Titel	18 - Expanderen van gelinkte resources
Strategie	Voor het expanderen van gelinkte resources wordt de query parameter ‘_expand’ gebruikt.
Toelichting	Gelinkte resources kunnen op verzoek worden meegeladen als onderdeel van een resourceverzoek (‘eager loading’). Dit dient dan te worden aangegeven via de ‘_expand’ query-parameter.
Onderbouwing	Het gebruik van de ‘_expand’ parameter voor het expanderen van gelinkte resources is vastgelegd in de <i>API Strategie Digitaal Stelsel Omgevingswet [API-B29]</i> [8].
Afwijking	

ID	19
Titel	19 - Representatie op maat
Strategie	Voor het gericht zoeken naar individuele velden (‘representatie op maat’) wordt de ‘_fields’ parameter gebruikt.
Toelichting	Gebruikers van een API hebben niet altijd de volledige representatie (lees alle velden) van een resource nodig. Daarom helpt het als de mogelijkheid bestaat om de gewenste velden te selecteren, we noemen dit ‘representatie op maat’. Hierdoor kan het netwerkverkeer worden beperkt (relevant voor lichtgewicht toepassingen), vereenvoudigt het gebruik van de API en is de uitvoer aanpasbaar (op maat).
Onderbouwing	Het gebruik van de ‘_fields’ parameter voor opvragen van individuele velden is vastgelegd in de <i>API Strategie Digitaal Stelsel Omgevingswet [API-Q01]</i> [8].
Afwijking	

ID	20
Titel	20 - HTTP headers
Strategie	HTTP headers worden gebruikt voor de beveiliging en het toevoegen van metadata aan de request en response.
Toelichting	HTTP headers zijn technische parameters die beveiligingsinformatie en metadata bevatten over de uitgewisselde berichten. Deze technische parameters kunnen opgenomen te worden in de request- of responseheaders van de berichten en zijn <u>geen</u> onderdeel van de functionele payload van het bericht. De cardinaliteit (verplicht of optioneel) van de HTTP headers is context afhankelijk en wordt beschreven in de Business Service en aangeduid in de OAS specificatie. De HTTP headers zijn gestandaardiseerd volgens de HTTP specificatie: HTTP/1.1:

ID	20
	Header Field Definitions (w3.org)
Onderbouwing	De in de HTTP specificatie voorgeschreven set van te ondersteunen technische HTTP headers voldoet aan de eisen van de energiesector. Aan deze set worden onderstaande technische parameters extensies (aangeduid met prefix 'X' conform ASNO) toegevoegd: • X-Object-ID (response): alternatief voor header Location indien geen complete URL in de response teruggegeven dient te worden. X-Object-ID bevat het identificatienummer van de gecreëerde resource; • X-Correlation-ID (request & response): bevat een unieke identificatie waarmee verschillende berichten in een transactie aan elkaar kunnen worden gekoppeld; • X-Request-ID (request): bevat de unieke identificatie van het request; • X-Sender-ID (request): de EAN13 van de marktpartij die de REST API aanroept; • X-Role (request): de rol van de marktpartij die met X-Sender-ID is aangeduid; • X-Principal-Token (request): bevat de unieke identificatie van de aanroepende partij (persoon of systeem) in de context van een request. Hoe de inhoud van het veld precies moet worden geïnterpreteerd is context afhankelijk; het kan b.v. een OAuth access token zijn.
Afwijking	

ID	21
Titel	21 - Enumeraties
Strategie	Enumeraties zijn onder voorwaarden toegestaan voor REST API's.
Toelichting	Enumeraties (opgesomde typen) zijn variabele typen in een dynamische of statische reeks van toegestane waarden: • Dynamische reeks: een reeks die kan wijzigen als gevolg van regelgeving of business requirements; • Statische reeks: een reeks die niet meer wijzigt, bijvoorbeeld alle dagen van de week. Enumeraties zijn populair in API ontwerp omdat ze worden gezien als een eenvoudige manier om die dynamische of statische reeks van toegestane waarden te beschrijven. Het gebruik van enumeraties is alleen toegestaan voor <u>opzichzelfstaande</u> REST API's omdat het aanpassen en uitbreiden van die enumeraties geen ketenproces impact heeft. Indien enumeraties gebaseerd zijn op een internationale standaard (zoals https://www.iso.org/iso-639-language-codes.html), dan moet men in het API ontwerp die notatie volgen. Indien enumeraties niet gebaseerd zijn op een internationale standaard, dan dient men enumeraties als mnemonische codes te definiëren in een UPPER_SNAKE_CASE notatie (conform API-66 van de API

ID	21
	Designrules Extensions [2]).
Onderbouwing	Door het opnemen van enumeraties in het JSON Schema, kan de API call vroegtijdig worden afgewezen indien de enumeraties in de API call niet voldoen aan de syntax.
Afwijking	Enumeraties zijn niet toegestaan in REST API's die onderdeel zijn van een ketenproces omdat een wijziging in de reeks van toegestane waarden voor een ketenproces veelal impact heeft op de gehele informatie-uitwisseling van zo'n proces en dient door de gehele sector te worden geïmplementeerd middels een sector release. De toegestane waarden van enumeraties worden dan beschreven in de bijbehorende servicebeschrijvingen.

ID	22
Titel	22 - Gebruik van gewone persoonsgegevens in de URL van REST API's
Strategie	Gewone persoonsgegevens mogen gebruikt worden in de URL van een REST API.
Toelichting	De privacy officers van de regionale netbeheerders hebben een zienswijze over het gebruik van gewone persoonsgegevens in de URL van REST API's geformuleerd op basis van de PPT van Enexis (Security & Privacy aspecten in REST API's - Enexis PPT CST-integratie dd. 03-06-2021).
Onderbouwing	Privacy Aspecten – Statement Er vanuit gaande dat: • Bij het gebruik van (gewone) persoonsgegevens in de URL het volledige bericht (body en URL) tijdens het versturen wordt versleuteld; • Deze gegevens uitsluitend bij de API Consumer, API Gateway en API Provider kunnen worden ingezien; • Gebruik van persoonsgegevens in de URL uitsluitend mag worden gebruikt, indien het noodzakelijk is dat de versturende én ontvangende partijen inzicht in deze (persoons)gegevens verkrijgen voor de uitvoering van hun taak; • Er geen gevoelige of bijzondere persoonsgegevens uit de URL kunnen worden afgeleid (Categorie 2); ontstaat er voor de Business een acceptabel risico voor het gebruik van een aantal gewone persoonsgegevens in de URL van de REST API's. Met name om het bouwen van API's beheerbaar en beheersbaar te maken. Hierbij valt te denken aan het gebruik van identificerende gegevens, zoals: • BAG; • EAN-code; • Klantnummer; • Meternummer; • Postcode huisnummer. Door een combinatie van bepaalde gewone persoonsgegevens in de URL van een

ID	22
	REST API kan er nieuwe gevoelige informatie ontstaan. Indien het combineren van gewone persoonsgegevens leidt tot gevoelige informatie in de URL, dan is deze combinatie niet toegestaan! Daarnaast zijn er een aantal gevoelige of bijzondere persoonsgegevens die – wanneer deze bekend worden – schade aan kunnen richten voor de klant. Deze gegevens mogen nooit in de URL van REST API's voorkomen. Indien het noodzakelijk is om deze gegevens uit te wisselen, zullen deze uitsluitend in de body worden verstuurd, zodat deze gegevens ook niet uit logging kunnen worden afgeleid. Het lijstje met bijzondere persoonsgegevens is limitatief, namelijk: • Ras of etnische afkomst; • Politieke opvattingen; • Religieuze of levensbeschouwelijke overtuigingen; • Lidmaatschap van een vakbond; • Genetische of biometrische gegevens; • Gezondheidsgegevens; • Gegevens over seksueel geaardheid; • BSN-nummer; • Strafrechtelijke gegevens. Bij gevoelige gegevens moet men denken aan bijvoorbeeld (niet-limitatief): • Fraudegegevens; • Financiële gegevens (zowel bankrekeningnummer/creditcard als betalingsgegevens); • Verbruiksgegevens; • Afsluitgegevens. Het is voor Privacy niet mogelijk om een limitatieve lijst van persoonsgegevens op te leveren en tot welke categorie deze behoren. Of iets een persoonsgegeven is en de gevoeligheid is namelijk sterk afhankelijk van de context waarin de gegevens worden gebruikt.
Afwijking	Het combineren van meerdere gewone persoonsgegevens in de URL van een REST API is niet toegestaan!

ID	23
Titel	23 – API specificatie
Strategie	API's dienen te worden gespecificeerd in de OpenAPI Specificatie standaard vanaf versie 3 of hoger (OAS v3). Bron: https://www.openapis.org/ . Voor de specificatie moet het JSON-formaat worden gehanteerd. Het YAML-formaat wordt niet ondersteund.

ID	23
Toelichting	Conform ASNO API-16
Onderbouwing	Door het afspreken van een éénduidig specificatieformaat (JSON) conform de OAS standaard, wordt de afstemming en samenwerking tussen providers en consumers vereenvoudigd en verbeterd. Het proces dat providers API's publiceren in JSON formaat conform OAS3, leidt tot voorspelbaarheid, verbeterde interoperabiliteit en draagt bij aan een hogere developer experience voor consumers.
Afwijking	

ID	24
Titel	24 – Gebruik van xxxOf constructies
Strategie	Het gebruik van xxxOf (allof, anyOf, oneOf) constructies in API's moet worden vermeden.
Toelichting	Alhoewel de allof, anyOf en oneOf constructies valide OAS3 constructies zijn, leveren deze bij het generen van code en bij het importeren van applicaties problemen op.
Onderbouwing	Diverse code generatoren en applicaties gaan niet goed om met deze constructies en genereren foutieve code en importeer fouten. Mocht dit in de toekomst verholpen zijn, dan is het overwegen waard om deze constructies alsnog toe te passen bij een eerstvolgende breaking change. Daaraan voorafgaand moet wel een marktconsultatie plaatsvinden om te controleren of de marktpartijen met deze constructies kunnen omgaan.
Afwijking	ASNO maakt geen expliciete melding van het gebruik van xxxOf constructies. Op de github site van VNG realisatie (API werkgroep van De Vereniging Nederlandse Gemeenten, komt de aanbeveling om xxxOf niet te gebruiken wel voor. Directe link: Haal-Centraal-common/design_decisions.md at master · VNG-Realisatie/Haal-Centraal-common · GitHub

Bijlage A: API Strategie Algemeen (Nederlandse API Strategie I)

API Strategie Algemeen (Nederlandse API Strategie I)



Geonovum Handreiking
Consultatieversie 20 december 2021

Deze versie:

<https://docs.geostandaarden.nl/api/cv-hr-API-Strategie-20211220/>

Laatst gepubliceerde versie:

<https://docs.geostandaarden.nl/api/API-Strategie/>

Vorige versie:

<https://docs.geostandaarden.nl/api/cv-hr-API-Strategie-20210915/>

Laatste werkversie:

<https://geonovum.github.io/KP-APIs/>

Redacteurs:

Frank Terpstra, [Geonovum](#)

Jan van Gelder, [Geonovum](#)

Auteurs:

Lancelot Schellevis, [Forum Standaardisatie](#)

Han Zuidweg, [Forum Standaardisatie](#)

Friso Penninga, [Geonovum](#)

Matthias Snoei, [Swis](#)

Jasper Roes, [Het Kadaster](#)

Peter Haasnoot, [Logius](#)

Frank van Es, [Logius](#)

Dennis de Wit, [Solventa](#)

Doe mee:

[GitHub geonovum/KP-APIs](#)

[Dien een melding in](#)

[Revisiehistorie](#)

[Pull requests](#)

Rechtenbeleid:



Creative Commons Attribution 4.0 International Public License
(CC-BY)

Bijlage B: REST-API Design Rules (Nederlandse API Strategie Ila)

REST-API Design Rules (Nederlandse API Strategie Ila) 1.0



Logius Standaard

Vastgestelde versie 09 juli 2020

This version:

<https://publicatie.centrumvoorstandaarden.nl/api/adr/1.0>

Latest published version:

<https://publicatie.centrumvoorstandaarden.nl/api/adr/>

Latest editor's draft:

<https://logius-standaarden.github.io/API-Design-Rules/>

Editors:

Frank Terpstra (Geonovum)
Jan van Gelder (Geonovum)

Authors:

Jasper Roes (Het Kadaster)
Joost Farla (Het Kadaster)

Participate:

[GitHub Logius-standaarden/API-Design-Rules](#)
[File a bug](#)
[Commit history](#)
[Pull requests](#)

This document is licensed under a [Creative Commons Attribution 4.0 License](#).

Bijlage C: API Designrules Extensions (Nederlandse API Strategie IIb)

API Designrules Extensions (Nederlandse API Strategie IIb)

Geonovum Handreiking
Vastgestelde versie 13 oktober 2021



This version:

<https://docs.geostandaarden.nl/api/def-hr-API-Strategie-ext-20211013/>

Latest published version:

<https://docs.geostandaarden.nl/api/API-Strategie-ext/>

Vorige versie:

<https://docs.geostandaarden.nl/api/cv-hr-API-Strategie-ext-20210628/>

Latest editor's draft:

<https://geonovum.github.io/KP-APIs/>

Editors:

Jasper Roes, [Het Kadaster](#)
Linda van den Brink, [Geonovum](#)
Frank Terpstra, [Geonovum](#)
Jan van Gelder, [Geonovum](#)
Joost Farla, [Het Kadaster](#)

Authors:

Jasper Roes, [Het Kadaster](#)
Joost Farla, [Het Kadaster](#)
Henri Korver, [VNG Realisatie](#)
Frank van Es, [Logius](#)
Frank Terpstra, [Geonovum](#)
Jaron Azaria, [Logius](#)
Martin Borgman, [Het Kadaster](#)
Peter Haasnoot, [Logius](#)
Leon van der Ree, [Logius](#)
Bob te Riele, [RvIG](#)
Remco Schaar, [Logius](#)
Jan Jaap Zoutendijk, [Rijkswaterstaat](#)
Heiko Hudig, [Politie](#)
Erwin Reinhoud, [Kennisset](#)
Michiel Trimpe, [Gemeente Amsterdam](#)
Tony Sloos, [Architect](#)

Participate:

[GitHub geonovum/KP-APIs](#)
[File a bug](#)
[Commit history](#)
[Pull requests](#)

Rechtenbeleid:



Creative Commons Attribution 4.0 International Public License
(CC-BY)