



Gebruikersdocumentatie Vektis services
Generieke afspraken Vektis services
REST API uitgave 13 december 2022



vektis

Colofon

Uitgave 13-12-2022

Auteurs

Vektis

Meer informatie

Voor veel gestelde vragen en om je specifieke vraag te stellen kun je terecht op de website van Vektis: www.vektis.nl.

Vektis

Sparrenheuvel 18
Gebouw B
3708 JE Zeist

Postbus 703
3700 AS ZEIST

T: 030 8008 300

F: 030 8008 320

E: info@vektis.nl

© 2020 Vektis

Wij hebben de inhoud van dit rapport met de grootste zorgvuldigheid samengesteld. We doen er alles aan deze actueel en juist te houden. Komt u desondanks toch iets tegen dat niet correct is ten opzicht van de Vektis services, dan stellen wij uw reactie bijzonder op prijs. De Vektis services worden ten tijde van deze publicatie verder ontwikkeld. Bij een nieuwe versie van de generieke afspraken wordt de publicatie hierop aangepast.

Vektis aanvaardt geen enkele aansprakelijkheid voor directe of indirecte schade als gevolg van het gebruik van de hier aangeboden informatie.

U mag alles uit deze uitgave kopiëren zolang u gebruikmaakt van bronvermelding.

Revisiehistorie

Datum	Aard/reden wijziging
19-05-2020	<ul style="list-style-type: none">• Aanpassing aan Vektis huisstijl• Samenvoegen functionele en technische documentatie
20-08-2020	Generieke afspraken Vektis services in plaats van Generieke afspraken AGB services
10-03-2021	P3.3 Antwoord 429 toegevoegd en 503 aangepast
22-09-2021	P2.3 Beveiligingsprotocollen (TLS) toegevoegd
30-03-2022	P2.4 Versiebeheer op meer concrete breaking changes aangepast
13-12-2022	P2.6 Antwoord 429 in plaats van 503

Inhoudsopgave

1.	Inleiding	5
1.1.	Leeswijzer	5
1.2.	Doelgroep	5
1.3.	Vektis services	5
1.4.	Gebruik Vektis services	6
2.	Vektis services principes	7
2.1.	Algemeen	7
2.2.	Omgevingen	7
2.3.	Koppelvlak en beveiliging	8
2.4.	Versiebeheer API	9
2.5.	Logging	10
2.6.	Fair Use Policy	10
3.	REST API Richtlijnen	11
3.1.	RESTful Methods	11
3.2.	Header	11
3.3.	Response code – HTTPStatus	12
3.4.	Request/response body	12
4.	Bijlage definities	14

1. Inleiding

1.1. Leeswijzer

De generieke afspraken Vektis services en REST API is de overkoepelende paraplu waar meerdere services binnen Vektis onder vallen. Te denken valt aan bijvoorbeeld ABG Raadpleegdienst, AGB Wijzigingendienst, de Vragenlijstendienst en de RIZ Bulkdienst – dit is geen limitatieve lijst – . Voor iedere service is het volgende pakket aan informatie beschikbaar:

- Een Quick guide van de Vektis service
- Een beknopte uitleg over de specifieke service
- De specificatie van de Vektis service met een procesbeschrijving en een beschrijving van de verschillende REST API's
- Testcases waar testgevallen in de testomgeving voor de verschillende REST API's beschreven zijn.

De generieke afspraken REST API voor Vektis services beschrijft

- H2 – Onder welke algemene principes en condities de Vektis services worden afgenomen.
- H3 – Richtlijnen van Vektis services binnen de REST API architectuur.

1.2. Doelgroep

Dit document is geschreven voor afnemers – of toekomstige afnemers – van Vektis services. Dit zijn over het algemeen klanten en softwareleveranciers die hun software systeem automatisch willen aansluiten op een of meerdere van de Vektis services die Vektis aanbiedt door middel van een koppelvlak met REST API's.

Om aan de slag te gaan met Vektis services is het noodzakelijk om, naast kennis over REST API, ook goede IT-kennis te hebben van de software waar je mee werkt. Daarnaast is het raadzaam om kennis op te doen van de werking van het specifieke product, deze informatie vindt je in beknopte uitleg over de specifieke service.

1.3. Vektis services

Vektis heeft een aantal Vektis services ontwikkeld voor de buitenwereld om geautomatiseerd aan te kunnen sluiten op haar producten, i.e het AGB-register. Deze zijn ontwikkeld op basis van een koppelvlak met een aantal REST API-calls per service. Onze services maken het mogelijk om met de eigen software contact te maken met het product. Dit is een zogenoemde machine-to-machine-verbinding.

1.4. Gebruik Vektis services

Dankzij de zorgverzekeraars stelt Vektis informatie uit de verschillende services momenteel kosteloos beschikbaar voor het zorgveld. Voorwaarde daarbij is dat wordt gehouden aan de gemaakte afspraken zoals deze zijn vastgelegd in de overeenkomst met bijbehorende voorwaarden.

Vooralsnog zijn de enige kosten voor de klant de kosten die men zelf maakt om aan te sluiten op de services en het aankopen van het certificaat.

2. Vektis services principes

2.1. Algemeen

- Uitwisseling is op basis van REST API architectuur, voor de Vektis specifieke richtlijnen zie het volgende hoofdstuk.
- De Vektis services worden direct bij Vektis afgenomen, zonder tussenpartijen.
- De Vektis services hebben een synchrone beleving (op elke vraag volgt direct een antwoord).
- De Vektis services leveren altijd de meest actuele stand van de gegevens: elke uitvraag is in principe een momentopname.
- De Vektis services koppelen alleen actieve gegevens terug (daar waar van toepassing op basis van peildatum).
- Als een Vektis service een functionele fout teruggeeft dan is dat een functioneel antwoord op de vraag: er is geen noodzaak om dezelfde vraag nog een keer te stellen.
- Als een Vektis service een technische fout teruggeeft dan is er een storing: van de aanroepende partij wordt verwacht dat die ervoor zorgt dat de Vektis service op een later moment, na 10 minuten nog een keer wordt benaderd. Als de fout dan nog steeds optreedt, neem dan contact op om de storing te melden.
- Vektis hanteert een fair use policy om misbruik te voorkomen, zie paragraaf 2.6.

2.2. Omgevingen

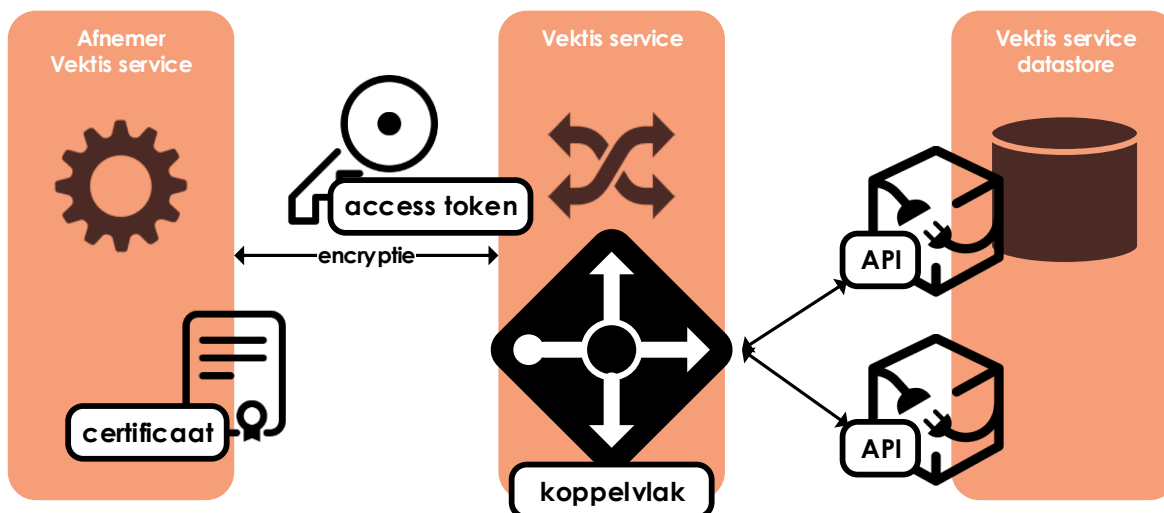
De Vektis services zijn beschikbaar via een productie- en een testomgeving. Beide omgevingen worden ondersteund door Swagger UI. Swagger UI is toegevoegd aan de Vektis service. Swagger UI biedt documentatie en de mogelijkheid om berichten handmatig te testen.

Per Vektis service is in een apart document beschreven:

- Met welk adres – Request URLs, inclusief Swagger UI - de omgevingen te benaderen zijn
- De testcases in de testomgeving

2.3. Koppelvlak en beveiliging

Onderstaand afbeelding illustreert op hoofdniveau de koppelingen tussen de afnemer en de Vektis service datastore door middel van het koppelvlak Vektis services.



Voor de Vektis services zijn een aantal lagen beveiliging geïmplementeerd.

Transport Level Security

Om verbinding te maken met de dienst wordt een beveiligde verbinding tussen de afnemer en de Vektis diensten vereist. Deze dient standaard TLS1.2 of hoger te zijn.

Certificaat

De omgevingen van de diensten zijn te bereiken via een basis URL en zijn alleen beschikbaar met een geldig SSL client certificaat. De communicatie met de diensten heeft encryptie doormiddel van HTTPS.

Voor de beveiliging van de verbinding wordt gebruik gemaakt van X.509-certificaten uitgegeven onder de Vektis-root: één voor de testomgeving (VEKTIS Test) en één voor de productieomgeving (VEKTIS).

Testomgeving

Je krijgt van Vektis een certificaat voor de testomgeving. Dit wordt opgestuurd na aanvraag.

Productieomgeving

Zorgverzekeraars en zorgaanbieders die een certificaat van VECOZO hebben kunnen die gebruiken.

Voor overige partijen verwijzen we uitsluitend naar Xolphin als reseller van certificaten.

Vektis ontvangt via Xolphin het certificaat dat is aangevraagd via

<https://www.digitalehandtekeningen.nl/email#3> (van Xolphin).

Access token

In de aanroep van de API wordt tevens een unieke access token meegestuurd, waarmee de identiteit van de externe bron wordt bepaald.

Het access token is door Vektis geleverd aan partijen waarvan zij een geldig certificaat hebben.

Testomgeving

Je krijgt van Vektis naast het certificaat een access token voor de test.

Productieomgeving

Je krijgt van Vektis een productie token na ondertekening van de overeenkomst. In verband met de veiligheid zal het productie token periodiek vervangen worden.

2.4. Versiebeheer API

Versiebeheer API wordt gedaan om transparant wijzigingen op de API te managen. Uitgangspunt voor een nieuwe versie zijn zogenaamde “**breaking changes**” voor de afnemers.

Het versiebeheer van de services is ingericht op hoofdniveau.

Een nieuwe versie is beschikbaar bij de volgende “breaking changes”:

- Het wijzigen van het request/response formaat (b.v. van JSON naar XML)
- Het wijzigen van een property naam (b.v. van Aanvang naar Begindatum) of data type (b.v. van een string naar integer)
- Het toevoegen van een verplicht veld aan het request (b.v. nieuwe verplichte header of een property in de request body)
- Het verwijderen van een property uit de response (b.v. StatutaireNaam uit Vestiging)

Een nieuwe versie zal vroegtijdig gecommuniceerd worden. Ook zal er duidelijk gecommuniceerd worden of bestaande versie blijft bestaan naast de nieuwe versie of dat deze wordt vervangen. Hierdoor is er voldoende tijd om de implementaties aan te passen.

Een nieuwe versie is *niet* beschikbaar bij de volgende wijzigingen:

- Het oplossen van technische fouten of eventuele beveiligingsrisico's krijgen de services geen nieuwe versie en kan in specifieke situaties de functionaliteit pas weer beschikbaar zijn als de oplossing aan twee kanten doorgevoerd is.
- Het toevoegen van nieuwe features of ingangen.
- Het toevoegen van een property aan de response (b.v. Beheerder aan Onderneming)

Er zal gecommuniceerd worden wanneer de functionaliteit toegevoegd is en deze kan vanaf dat moment voor de gebruiker geïmplementeerd worden.

2.5. Logging

Voor de Vektis services is de algemene richtlijn dat we minimaal loggen voor raadplegen. Minimaal loggen houdt in dat de services een log bijhouden van verzoeken die binnenkomen en van fouten die optreden tijdens de uitwisseling.

De inhoud van de berichten wordt niet bewaard bij raadpleegacties (GET). Bij het doorgeven van informatie (POST) worden de berichten wel vastgelegd voordat ze verwerkt worden.

2.6. Fair Use Policy

Per access token kan er een maximum van 10 calls per seconde in behandeling genomen worden. Wanneer een grote hoeveelheid data wordt bevraagd kan het antwoord daarom wat langer duren. Calls die boven de limiet zitten krijgen als antwoord 429 (too many requests).

Het is niet toegestaan om alle mogelijke cijfercombinaties binnen de range opeenvolgend te bevragen. We helpen je graag verder om de juiste informatie op te halen.

3. REST API Richtlijnen

Vektis heeft gekozen om functionaliteit van haar producten via Vektis services beschikbaar te maken. Daarvoor is per Vektis service een koppelvlak beschikbaar met REST API's.

Informatie over REST API's, zie ook [https://www.w3schools.in/restful-web-services/intro/#The REST Architecture](https://www.w3schools.in/restful-web-services/intro/#The_REST_Architecture).

Dit hoofdstuk beschrijft de richtlijnen van Vektis services binnen de REST API architectuur.

3.1. RESTful Methods

Vektis services maakt gebruik van de onderstaande gebruikelijke HTTP-methoden

- GET: Het raadplegen van gegevens.
- POST: Het aanmaken van een nieuwe entiteit.
- PUT: Het wijzigen van een entiteit of het notificeren over een wijziging.

Alle entiteiten worden met HTTP GET benaderd. Als er andere methoden beschikbaar zijn, wordt dit per Resource aangegeven.

3.2. Header

Accept property

Met behulp van de accept property in de header kan het formaat van de response gestuurd worden. De mogelijkheden:

- Accept: application/json
- Accept: application/xml

Content-Type property

In welk formaat de request bij een POST gestuurd is staat in de Content-Type property. De mogelijkheden:

- Content-Type: application/json
- Content-Type: application/xml

Authorization property

Het access token dat door Vektis aangeleverd is, moet meegegeven worden met behulp van een authorization property:

- Authorization: Vektis <AccessToken>

3.3. Response code – HTTPStatus

We beschrijven hier welke response codes wij geïmplementeerd hebben en wat het betekent als een vraagbericht dit als antwoord geeft.

- 200 (ok): Bij een goed bericht dat verwerkt kan worden zal altijd een code 200 gegeven worden.
- 400 (bad request): Als het vraagbericht of de aangeboden parameters niet geldig zijn.
- 401 (unauthorized): Als het access token dat meegestuurd is in de autorisatie header niet geldig is.
- 403 (forbidden): Als de beveiligingsconfiguratie niet goed is of informatie opgevraagd wordt waar de gebruiker geen rechten voor heeft.
- 404 (not found): Als de resource die gevraagd wordt niet bestaat.
- 429 (too many requests): Als er meer dan 10 calls per seconde per token worden uitgevoerd, wordt dit op elk bericht daarboven terug gegeven.
- 500 (server error): Als er een technische fout optreedt. Als reactie hierop stellen wij voor de actie na ongeveer 10 minuten te herhalen. Als de fout dan nog steeds optreedt, neem dan contact op om de storing te melden.
- 503 (service unavailable): de opgevraagde service is (tijdelijk) niet beschikbaar. Dit kan komen door een overload van de server of een service- onderbreking wegens onderhoud.

Specifieke Response codes worden bij de documentatie van de AGB-service vermeld.

3.4. Request/response body

De inhoud van de request en response body heeft formaat JSON of XML. Hieronder is aangegeven hoe de body is gevuld:

- De standaard tekenset is UTF-8.
- Formaat van uitwisseling is JSON of XML.
- Numerieke velden hebben datatype 'String' : bij een request wordt de controle via een reguliere expressie – RegEx – afdgedwongen.
Zo bestaat een AGB-code altijd uit 8 posities, daar waar nodig wordt een voorloopnul gebruikt.
- Voor datumvelden is het standaard datumformaat: (YYYY-MM-DDTHH:mm:ss) (bijv. 2018-01-01T00:00:00)
- Email heeft maximaal 128 posities wat wordt getoetst aan de standaard conventies van een Emailadres
- Alle objecten, eigenschappen en lijsten zijn aanwezig, ook als deze leeg zijn.
 - Een object dat niet gevuld is heeft de waarde null.
 - Een attribuut wat niet gevuld is, heeft een lege string als waarde.
 - Een lege lijst is als een lijst zonder elementen beschikbaar.

4. Bijlage definities

In dit document worden verschillende definities gebruikt. Hieronder wordt per definitie een omschrijving gegeven.

Definitie	Omschrijving
API	Application Programming Interface
HTTP(S)	Hyper Text Transfer Protocol (Secured)
JSON	Javascript Object Notation: manier om gegevens in javascript op te slaan
REST	Representational state transfer: lichtgewicht internet protocol om gegevens tussen partijen uit te wisselen.
SSL	Secure Socket Layer
URL	Uniform Resource Locator
URI	Uniform Resource Identifier
XML	Extensible Markup Language