



ZORG-AB Implementatiehandleiding

Release: 2.7.0

Documentversie: 2.6

Status: Final

Product : ZORG-AB
Release : 2.7.0
Auteur : DXC Technology
Documentversie : 2.6
Status : Final
Uitgiftedatum : 24-06-2021
Bestandsnaam : ZORG-AB Implementatiehandleiding 2.7.0 20210624 Final.docx

Revisie historie

Release	Document Versie	Document Datum	Document Status	Auteur	Revisie Omschrijving
1.0.0	0.2	21-11-2016	Final	M. Siliakus (DXC)	Initiële versie
(Tussenliggende revisies verwijderd)					
2.6.3	2.3	27-07-2020	Final	M. Siliakus (DXC)	<ul style="list-style-type: none"> Endpoints read-only API's toegevoegd par. 1.2 Beschrijving toegevoegd van niet-ondersteunde codes bij zorgaanbiedertypequery REST interface par. 3.11.1, FHIR interface par. 4.8.2 Voorbeeld EDIFACT toegevoegd REST interface par. 3.18, FHIR interface par. 4.14 Beschrijving van koppeling met Vektis AGB Services toegevoegd par. 5.1
2.6.5	2.4	18-11-2020	Final	M. Siliakus (DXC)	<ul style="list-style-type: none"> Datamodel bijgewerkt par. 2.1 Toegevoegd par. 2.4 Correcties/toevoegingen par. 3.1, par. 3.4 en par. 4.7
2.6.6	2.5	18-02-2021	Final	M. Siliakus (DXC)	<ul style="list-style-type: none"> Beschrijving gecombineerde query's toegevoegd REST interface par. 3.6, FHIR interface par. 4.10 Par. 4.9 bijgewerkt i.v.m. dat vrijetekst query nu kan worden uitgevoerd via '_text' parameter, 'custom query' gemarkeerd als Deprecated /spOrganizations toegevoegd bijlage 1
2.7.0	2.6	24-06-2021	Final	M. Siliakus (DXC)	<ul style="list-style-type: none"> Datamodel bijgewerkt par. 2.1 Bijgewerkt Nictiz waardelijsten URL's par. 2.1 Toegevoegd par. 2.5 Bijgewerkt par. 3.19 Bijgewerkt par. 5.2 Toegevoegd par. 5.5 Toegevoegd /applicationId-v2 aan Bijlage 1

Verspreiding

Release	Document Versie	Ontvangers
1.0.0	0.2	DXC
(Tussenliggende revisies verwijderd)		
2.6.3	2.3	DXC, VZVZ, XIS-leveranciers
2.6.5	2.4	DXC, VZVZ, XIS-leveranciers
2.6.6	2.5	DXC, VZVZ, XIS-leveranciers
2.7.0	2.6	DXC, VZVZ, XIS-leveranciers

Inhoudsopgave

1	Inleiding	6
1.1	Gebruikte Standaarden/Frameworks	6
1.2	Verbinden met ZORG-AB	7
1.2.1	Endpoints	7
1.2.2	Certificaat	7
1.2.3	Testen van de Verbinding (REST Interface)	7
1.2.4	Testen van de Verbinding (FHIR Interface)	8
1.3	OAS (Swagger) Web Service Definities	8
1.4	Voorbeeldquery's en -berichten	8
2	Datamodel	9
2.1	Entiteiten en Attributen	9
2.2	Toelichting op de Entiteit 'Party'	20
2.3	Hoofdletters en Kleine Letters	20
2.4	Author	20
2.4.1	Doeleinden	20
2.4.2	Author van Hoofdobject / Author van Gegevensobject	20
2.4.3	Mogelijke Authors	22
2.4.4	Wijzigen van Gegevens	23
2.4.5	Toevoegen van Gegevens	23
2.5	Suborganisaties	24
2.5.1	Suborganisatie Zoeken o.b.v. LSP Applicatie ID	28
3	Gebruik van de Native REST Interface	29
3.1	Resources	29
3.2	HTTP-methoden	32
3.3	Queryparameters	32
3.4	Voorgedefinieerde Query's (\$filter)	32
3.5	Vrije Tekst Query's (\$search)	35
3.6	Gecombineerde Query's (\$search i.c.m. \$filter)	37
3.7	\$select	39
3.8	Content Type	39
3.9	Validatie en Restricties	39
3.10	Voorbeeld van het Uitvoeren van een Query	39
3.11	Voorbeeld van het Uitvoeren van de Zorgaanbiedertype-query ('JGZ-query')	42
3.11.1	Niet-Ondersteunde Vektis Zorgsoortcodes	44
3.12	Voorbeeld van het Gebruik van de 'StartsWith' Functie in een Query	45
3.13	Voorbeeld van het Toevoegen van een Organisatie (POST-Request)	46
3.14	Voorbeeld van het Toevoegen van een Suborganisatie (POST-Request)	46
3.15	Voorbeeld van het Toevoegen van een Individu (POST-Request)	47
3.16	Voorbeeld van het Toevoegen van Contactgegevens die Alleen Zichtbaar Mogen Zijn voor Collega-Zorgaanbieders	48
3.17	Voorbeeld van het Toevoegen van Beschikbaarheidsgegevens van een Zorgverlener	49
3.18	Voorbeeld van het Toevoegen van EDIFACT Emailadres	51
3.19	MedMij Gegevensdiensten	52
3.19.1	Voorbeeld Response Native REST Interface	52
3.19.2	Zoeken o.b.v. MedMij Naam via Native REST Interface (OData Interface)	53
3.19.3	Voorbeeld Response FHIR Interface	53
3.19.4	Zoeken o.b.v. MedMij Naam via FHIR Interface	54
4	Gebruik van de FHIR Interface	56
4.1	Algemeen	56
4.2	Resources / Interacties	56
4.3	Content Type	56

4.4	Validatie en Restricties	57
4.5	Voorgedefinieerde Query's	57
4.6	Voorbeeld van een Query	58
4.7	Voorbeeld van Zoeken op URA- of UZI- of KVK-nummer of AGB-code	59
4.8	Voorbeeld van de Zorgaanbiedertype-query ('JGZ-query')	62
4.8.1	Toelichting LSP-applicatie ID's	65
4.8.2	Niet-Ondersteunde Nictiz Zorgaanbiedertypecode 'Z3' en Niet-Ondersteunde Vektis Zorgsoortcodes	66
4.8.3	XML niet Ondersteund bij Zorgaanbiedertypequery	67
4.9	Vrije Tekst Query's	67
4.10	Gecombineerde Query's (LSP-query)	68
4.11	Algemene Queryparameters	70
4.12	PATCH	70
4.13	Voorbeeld van het Toevoegen van Beschikbaarheidsgegevens van een Zorgverlener	71
4.14	Voorbeeld van het Toevoegen van EDIFACT Emailadres	72
5	Koppelingen met Externe Gegevensbronnen	74
5.1	Vektis AGB Services	74
5.2	VZVZ Clientèlebestand (BI-bestand)	75
5.3	UZI LDAP	76
5.4	Live Koppeling met het LSP	76
5.5	Koppeling met MedMij ZKL (ZorgaanbiedersKoppelLijst)	76
	Bijlage 1 Lijst van Geldige URI's (Native REST Interface)	78
	Bijlage 2 Lijst van Geldige URI's (FHIR Interface)	85
	Bijlage 3 Configuratie SoapUI (Native REST Interface)	88

1 Inleiding

Dit document bevat informatie, instructies en voorbeelden ten behoeve van de implementatie van (applicaties/systemen die gebruik maken van) de web API's van ZORG-AB.

ZORG-AB bevat gegevens van zowel zorgaanbieders (organisaties) als zorgverleners (specialisten). Zorgaanbieders worden in ZORG-AB Organizations genoemd. Zorgverleners worden in ZORG-AB Individuals of (in de context van de HL7 FHIR interface) Practitioners genoemd.

ZORG-AB biedt onder andere de volgende functionaliteit:

- Twee op standaarden gebaseerde web API's:
 - Een native REST interface met ondersteuning voor JSON
 - Een HL7 FHIR interface met ondersteuning voor JSON en XML
- Koppelingen met verschillende externe gegevensbronnen (zie hoofdstuk 5 voor details):
 - Vektis AGB Services
 - Het cliëntèlebestand (BI-bestand) van VZVZ
 - UZI LDAP
 - Een live koppeling met het applicatieregister van het LSP
 - MedMij ZKL (ZorgaanbiedersKoppelijst)
- Twee geavanceerde zoekmogelijkheden:
 - Gericht en efficiënt zoeken met voorgedefinieerde query's (zie par. 3.4 en par. 4.5)
 - Zoeken op een zoekmachine-achtige manier met vrije tekst query's (zie par. 3.5 en par. 4.9)
 - ZORG-AB maakt gebruik van Soundex en 'fuzzy search' zodat een zorgaanbieder bij een afwijking qua spelling toch gevonden kan worden (bijvoorbeeld zoeken op 'Apotheek Jansen' levert ook een match op met 'Apotheek Janssen')
 - De vrije tekst query's ondersteunen het gebruik van wild cards
- Voor zowel de native REST interface als FHIR interface wordt volledige ondersteuning geboden voor het muteren van gegevens (CRUD). ZORG-AB heeft een geavanceerd 'Author' model waarmee geborgd wordt dat gegevens alleen gemuteerd kunnen worden door de eigenaar van de gegevens (dit kan de zorgaanbieder zijn maar ook een externe gegevensbron zoals het UZI register)
- Overige mogelijkheden zoals de JGZ-query (zie par. 3.11 en par. 4.8), zoeken op AGB-code en KVK-nummer (zie par. 3.4 en par. 4.7)

1.1 Gebruikte Standaarden/Frameworks

De API's en het datamodel van ZORG-AB zijn gebaseerd op standaarden/frameworks van OData, IHE, HL7 en Argonaut. Dit wordt samengevat in de volgende tabel:

ZORG-AB Onderdeel	Standaard/Framework	Versie/Link
Native REST interface queryparameters \$filter, \$search en \$select	OData URL Conventions	4.0
Datamodel	IHE Healthcare Provider Directory (HPD) IT Infrastructure Technical Framework Supplement	1.7

HL7 FHIR interface	HL7 FHIR	3.0.1 Release 3 (STU)
	Argonaut Provider Directory Implementation Guide	1.0.0

1.2 Verbinden met ZORG-AB

1.2.1 Endpoints

Er is zowel een test- als een productieomgeving beschikbaar met beide zowel een native REST interface als een HL7 FHIR interface.

De testomgeving is toegankelijk via zowel ZorgNet als Internet.

In de productieomgeving zijn de standaard API's toegankelijk via ZorgNet en zijn de read-only API's toegankelijk via Internet. Via de standaard API's kan worden gelezen en kunnen ook gegevens worden toegevoegd/gemuteerd. De read-only API's bieden dezelfde functionaliteit als de standaard API's uitgezonderd dat toevoegen/muteren van gegevens niet mogelijk is.

De betreffende endpoints kunnen worden gevonden in onderstaande tabel:

Nr.	Omgeving	Interface	Endpoint
1A	Test ZorgNet/Internet	Native REST	https://zab.test.lsp.aorta-zorg.nl/zab
1B		HL7 FHIR	https://zab.test.lsp.aorta-zorg.nl/zab/fhir
2A	Productie ZorgNet	Native REST	https://zab.prod.lsp.aorta-zorg.nl/zab
2B		HL7 FHIR	https://zab.prod.lsp.aorta-zorg.nl/zab/fhir
2C	Productie Internet	Native REST Read-Only	https://zab.prod.aorta-zorg.nl/zab-ro
2D		HL7 FHIR Read-Only	https://zab.prod.aorta-zorg.nl/zab-ro/fhir

1.2.2 Certificaat

Er moet om te verbinden met ZORG-AB gebruik worden gemaakt van een UZI-servercertificaat of (ander) PKI-certificaat en TLS v1.2.

1.2.3 Testen van de Verbinding (REST Interface)

De verbinding kan worden getest door het volgende request uit te voeren:

Omgeving	Request
Test	<code>GET https://zab.test.lsp.aorta-zorg.nl/zab/organizations</code>
Productie ZorgNet	<code>GET https://zab.prod.lsp.aorta-zorg.nl/zab/organizations</code>
Productie Internet	<code>GET https://zab.prod.aorta-zorg.nl/zab-ro/organizations</code>

Als de volgende response wordt opgeleverd dan is de test succesvol:

```
{"error": "No $filter or $search supplied"}
```

1.2.4 Testen van de Verbinding (FHIR Interface)

De verbinding kan worden getest door het volgende request uit te voeren:

Omgeving	Request
Test	GET https://zab.test.lsp.aorta-zorg.nl/zab/fhir/metadata
Productie ZorgNet	GET https://zab.prod.lsp.aorta-zorg.nl/zab/fhir/metadata
Productie Internet	GET https://zab.prod.aorta-zorg.nl/zab-ro/fhir/metadata

Als er een Capability Statement wordt opgeleverd dan is de test succesvol.

1.3 OAS (Swagger) Web Service Definities

Er is voor de native REST interface een web service definitie beschikbaar in OAS (Swagger) formaat. Hetzelfde geldt voor de read-only versie van de native REST interface. De bestanden zijn beschikbaar in zowel JSON als YAML formaat. Het betreft de volgende bestanden:

- ZORG_AB_2_7_0_OAS_20210603.json
- ZORG_AB_2_7_0_Read_Only_OAS_20210603.json
- ZORG_AB_2_7_0_OAS_20210603.yaml
- ZORG_AB_2_7_0_Read_Only_OAS_20210603.yaml

De bestanden bevatten een volledige specificatie van de native REST interface, uitgezonderd dat de OAS niet beschrijft hoe de queryparameter \$filter gevuld moet worden. Dit wordt beschreven in dit document in par. 3.4.

(En zie voor het gebruik van de queryparameter \$search par. 3.5.)

1.4 Voorbeeldquery's en -berichten

Voor de native REST interface kunnen voorbeeldquery's en -berichten worden gevonden in de volgende paragrafen:

- Zoeken via voorgedefinieerde query's: par. 3.4
- Zoeken via vrije tekst query's: par. 3.5
- Muteren van gegevens: bijv. par. 3.18

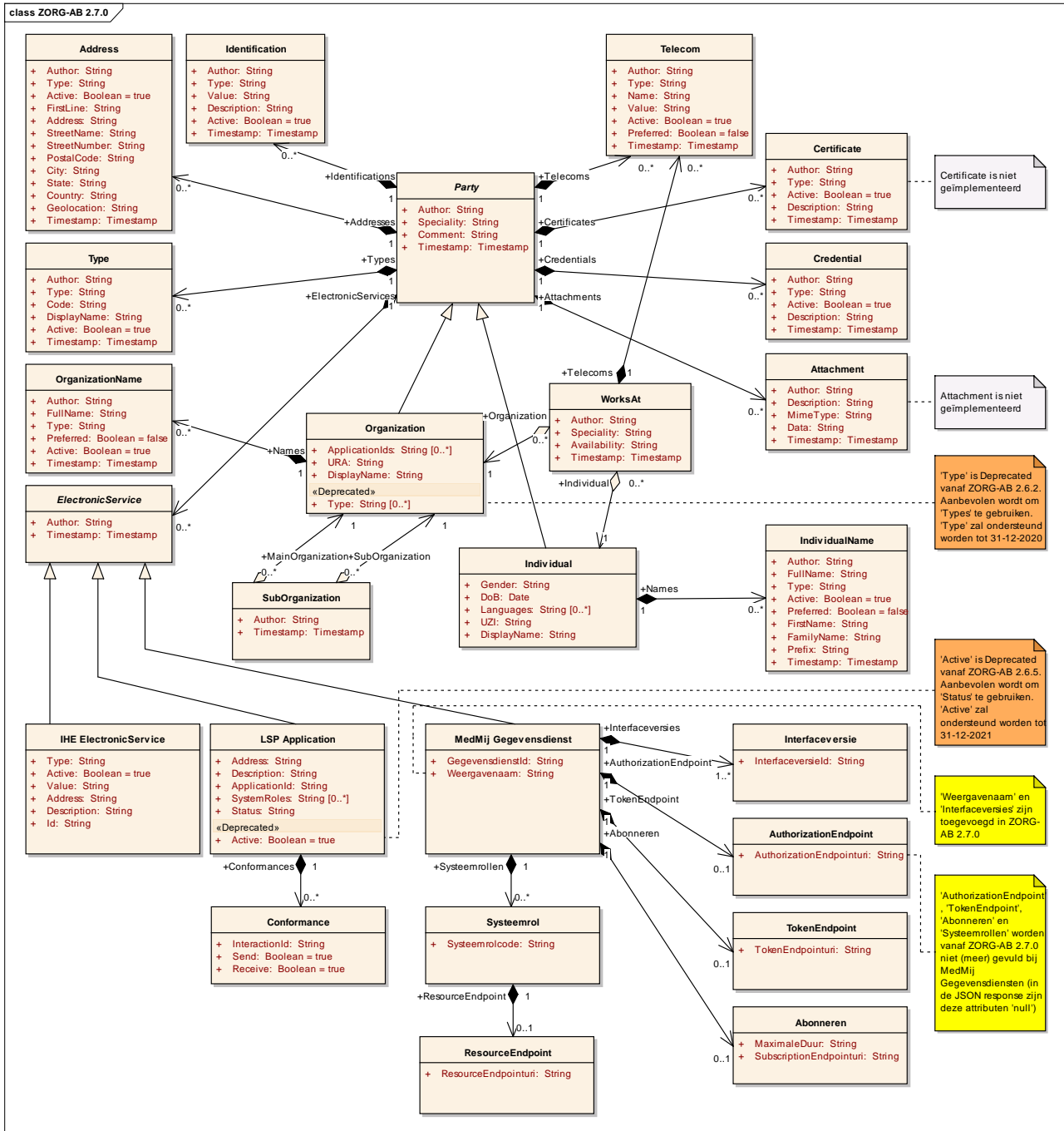
Voor de HL7 FHIR interface kunnen voorbeeldquery's en -berichten worden gevonden in de volgende paragrafen:

- Zoeken via voorgedefinieerde query's: par. 4.5
- Zoeken via vrije tekst query's: par. 4.9
- Muteren van gegevens: par. 4.12

2 Datamodel

2.1 Entiteiten en Attributen

Het datamodel van ZORG-AB wordt weergegeven in de onderstaande figuur:



Figuur 1. Datamodel ZORG-AB 2.7.0

Aanpassingen van het Datamodel

ZORG-AB 2.6.2:

- Het attribuut Types (meervoud) is toegevoegd aan de klasse Party en daarmee aan de subklassen Organization en Individual
- Het attribuut Type (enkelvoud) van de klasse Organization is gemarkeerd als Deprecated. Aanbevolen wordt om het attribuut Types (meervoud) te gebruiken. Het attribuut Type zal ondersteund worden tot 31-12-2020

ZORG-AB 2.6.5:

- Het attribuut Author is toegevoegd aan alle klassen (uitgezonderd de klassen die zijn gekoppeld aan LSP Application en MedMij Gegevensdienst: Conformance, AuthorizationEndpoint, TokenEndpoint, Systeemrol en ResourceEndpoint).
NB Organization en Individual erven het attribuut Author van hun superklasse Party. IHE ElectronicService, LSP Application en MedMij Gegevensdienst erven het attribuut Author van hun superklasse ElectronicService
- Het attribuut Status is toegevoegd aan de klasse LSP Application
- Het attribuut Active van de klasse LSP Application is gemarkeerd als Deprecated. Aanbevolen wordt om het attribuut Status te gebruiken. Het attribuut Active zal ondersteund worden tot 31-12-2021

ZORG-AB 2.7.0:

De klasse MedMij Gegevensdienst is als volgt gewijzigd:

- De attributen Weergavenaam en Interfaceversies zijn toegevoegd aan de klasse MedMij Gegevensdienst.
- Het datamodel is gebaseerd op enerzijds de ZKL XSD van MedMij Afsprakenstelsel 1.3.0 en anderzijds de ZAL XSD van MedMij Afsprakenstelsel 1.3.0. In de ZAL XSD is ook het attribuut 'Abonneren' gedefinieerd. Daarom is ook het attribuut 'Abonneren' toegevoegd aan de klasse MedMij Gegevensdienst. Dit attribuut wordt echter (vooralsnog) niet gevuld.
- De attributen AuthorizationEndpoint, TokenEndpoint, Abonneren en Systeemrollen van MedMij Gegevensdienst worden niet (meer) gevuld. In het JSON-antwoordbericht zijn deze attributen 'null'.

Toelichting op het Datamodel

In de onderstaande tabel wordt een toelichting gegeven per entiteit en attribuut:

Entiteit	Toelichting	Attribuut	Toelichting
Party	Superklasse van zorgaanbieder (Organization) en zorgverlener (Individual)	Author	'Eigenaar' van het zorgaanbieder (Organization) of zorgverlener (Individual) object exclusief gekoppelde gegevensobjecten (Types, Names, Addresses etc.). Voor toelichting zie par. 2.4
		Addresses	Lijst (array) van 1 of meer Addresses, zie Address
		Telecoms	Lijst (array) van 1 of meer Telecoms, zie Telecom

		Identifications	Lijst (array) van 1 of meer Identifications, zie Identification
		Credentials	Lijst (array) van 1 of meer Credentials, zie Credential
		ElectronicServices	Lijst (array) van 1 of meer ElectronicServices, zie Electronic Service
		Attachments	Lijst (array) van 1 of meer Attachments, zie Attachment
		Speciality	<p>Specialisme van de zorgverlener (of zorgaanbieder). Het specialisme wordt niet opgeleverd in dit attribuut. Het specialisme van de zorgverlener wordt opgeleverd in het attribuut Speciality van de WorksAt relatie. De WorksAt relatie kan worden gevonden via bijv. GET <code>[Base URL]/individuals/[ID]/relations</code> Zie ook WorksAt.</p>
		Comment	Commentaarveld t.b.v. een toelichting op de gegevens van de zorgaanbieder/-verlener
		Types	Zorgaanbiedertype(s). Dit is een lijst die 1 of meer codes bevat conform mogelijk verschillende waardelijsten/codesystemen. De lijst bevat instanties van de klasse Type, zie Type
		Timestamp	Datum en tijdstip van laatste wijziging
Organizati on	Zorgaanbi eder	Names	Lijst (array) van 1 of meer OrganizationNames, zie OrganizationName
		Type	<p>Lijst (array) van 1 of meer zorgaanbiedertypcodes (strings), bijvoorbeeld 'Z001' in het geval van een apotheek</p> <p>Deprecated vanaf ZORG-AB 2.6.2. Aanbevolen wordt om 'Types' te gebruiken. 'Type' wordt ondersteund tot 31-12-2020</p>
		ApplicationI ds	Alleen-lezen. Bevat de LSP applicatie ID's van de zorgaanbieder (Om precies te zijn bevat dit attribuut de ApplicationId's van de LSP Applications in het attribuut ElectronicServices)
		URA	Alleen-lezen. Bevat het URA-nummer van de zorgaanbieder (Om precies te zijn bevat dit attribuut de Value van de Identification met Type= 'URA' in het attribuut Identifications)
		DisplayNa me	Alleen-lezen. Bevat de naam met Type='Display' van de zorgaanbieder

			(Om precies te zijn bevat dit attribuut de FullName van de OrganizationName met Active=true en Type='Display' in het attribuut Names. In het geval dit meerdere namen zijn wordt de naam met Preferred=1 geselecteerd)						
Individual	Zorgverlener	Names	Lijst (array) van 1 of meer IndividualNames, zie IndividualName						
		Gender	<p>Geslacht, geldige waardes zijn: ¹</p> <ul style="list-style-type: none"> • male • female • other • unknown <p>Validatie: bij het gebruiken van een ongeldige waarde in de request body wordt een HTTP 400 Bad Request teruggegeven met de foutmelding:</p> <pre>{"error": "Invalid gender <waarde>"}</pre>						
		DoB	<p>Geboortedatum</p> <p>Formaat: in JSON requests moet de geboortedatum meegegeven worden als een 'string' in het formaat YYYYMMDD</p>						
		Languages	<p>Lijst (array) van BCP-47 taalcodes, bijvoorbeeld 'nl' of 'nl-BE'.</p> <p>Validatie: bij het gebruiken van een ongeldige waarde in de request body wordt een HTTP 400 Bad Request teruggegeven met de foutmelding:</p> <pre>{"error": "Invalid language <waarde>"}</pre>						
		UZI	<p>Alleen-lezen. Bevat het UZI-nummer van de zorgverlener</p> <p>(Om precies te zijn bevat dit attribuut de Value van de Identification met Type= 'UZI' in het attribuut Identifications)</p>						
		DisplayName	<p>Alleen-lezen. Bevat de naam met Type='Display' van de zorgverlener</p> <p>(Om precies te zijn bevat dit attribuut de FullName van de IndividualName met Active=true en Type='Display' in het attribuut Names. In het geval dit meerdere namen zijn wordt de naam met Preferred=1 geselecteerd)</p>						
OrganizationName		Author	'Eigenaar' van het gegevensobject (OrganizationName). Voor toelichting zie par. 2.4						
		FullName	De naam van de zorgaanbieder (van het type aangegeven bij Type)						
		Type	<p>Type van de naam, bijvoorbeeld 'Legal', 'Display', 'Tradename' of 'Location'</p> <p>Toelichting:</p> <table border="1" data-bbox="635 1821 1046 1861"> <thead> <tr> <th>Bron</th> <th>Type Naam</th> <th>Type</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> </tr> </tbody> </table>	Bron	Type Naam	Type			
Bron	Type Naam	Type							

¹ Gebaseerd op HL7 FHIR AdministrativeGender: <http://hl7.org/fhir/STU3/valueset-administrative-gender.html>

			<table border="1"> <tr> <td>Vektis</td> <td>Handelsnaam</td> <td>Tradename</td> </tr> <tr> <td></td> <td>Naam</td> <td>Display</td> </tr> <tr> <td></td> <td>Statutaire Naam</td> <td>Legal</td> </tr> <tr> <td>VZVZ</td> <td>Gevelnaam</td> <td>Display</td> </tr> <tr> <td></td> <td>Locatiennaam</td> <td>Location</td> </tr> <tr> <td>UZI</td> <td>UZI-naam</td> <td>Legal</td> </tr> </table> <p><i>I.v.m. backward compatibility met de OAS van ZORG-AB 2.6.2 worden OrganizationNames met type 'Tradename' en 'Location' vooralsnog niet opgeleverd</i></p>	Vektis	Handelsnaam	Tradename		Naam	Display		Statutaire Naam	Legal	VZVZ	Gevelnaam	Display		Locatiennaam	Location	UZI	UZI-naam	Legal
Vektis	Handelsnaam	Tradename																			
	Naam	Display																			
	Statutaire Naam	Legal																			
VZVZ	Gevelnaam	Display																			
	Locatiennaam	Location																			
UZI	UZI-naam	Legal																			
		Preferred	Boolean dat gebruikt kan worden om aan te geven dat de voorkeur bestaat dat deze naam wordt gebruikt of weergegeven i.p.v. de andere namen																		
		Active	Geeft aan of deze naam actief is. Zowel actieve als inactieve namen worden geïndexeerd, d.w.z. er kan zowel op actieve als inactieve namen worden gezocht via de verschillende query's. Een technisch verschil is dat inactieve namen niet getoond worden in het DisplayName veld (zie Organization)																		
		Timestamp	Datum en tijdstip van laatste wijziging																		
Individual Name		Author	'Eigenaar' van dit gegevensobject (IndividualName)																		
		FullName	De naam van de zorgverlener (van het type aangegeven bij Type)																		
		Type	Type van de naam, bijvoorbeeld 'Legal' of 'Display'																		
		Active	Geeft aan of deze naam actief is. Zowel actieve als inactieve namen worden geïndexeerd, d.w.z. er kan zowel op actieve als inactieve namen worden gezocht via de verschillende query's. Een technisch verschil is dat inactieve namen niet getoond worden in het DisplayName veld (zie Individual)																		
		Preferred	Boolean dat gebruikt kan worden om aan te geven dat de voorkeur bestaat dat deze naam wordt gebruikt of weergegeven i.p.v. de andere namen																		
		FirstName	Voornaam, bijvoorbeeld 'Jan'																		
		FamilyName	Achternaam, bijvoorbeeld 'Zand'																		
		Prefix	Voorvoegsel, bijvoorbeeld 'van het'																		
		Timestamp	Datum en tijdstip van laatste wijziging																		
Address		Author	'Eigenaar' van het gegevensobject (Address). Voor toelichting zie par. 2.4																		
		Type	Type van het adres, bijvoorbeeld 'Postal', 'Billing', 'Practice' of 'Delivery'																		
		Active	Geeft aan of dit adres actief is. Inactieve adressen worden getoond bij de zorgaanbieder of -verlener maar worden niet geïndexeerd, d.w.z. als een adres op inactief wordt gezet wordt de zorgaanbieder of -verlener bijvoorbeeld niet opgeleverd wanneer op deze plaatsnaam wordt gezocht via de verschillende query's.																		

			(Zodra het adres weer op actief wordt gezet is de zorgaanbieder of -verlener weer vindbaar onder de betreffende plaatsnaam)						
		FirstLine	Eerste adresregel (de regel vóór straat/huisnummer)						
		Address	Volledig adres (indien leeggelaten, wordt automatisch gevuld met concatenatie van de 'losse' adresvelden)						
		StreetName	Straatnaam						
		StreetNumber	Huisnummer						
		PostalCode	Postcode						
		City	Woonplaats						
		State	Provincie						
		Country	ISO 3166-1 alpha-2 code van het land, bijvoorbeeld 'NL'						
		Geolocation	Geografische coördinaten, bijvoorbeeld '40.714172,-74.006393'						
		Timestamp	Datum en tijdstip van laatste wijziging						
Telecom	Contactgegeven	Author	'Eigenaar' van het gegevensobject (Telecom). Voor toelichting zie par. 2.4						
		Type	Type van het contactgegeven, bijvoorbeeld 'Phone', 'Email', 'Fax' of 'URL' <i>I.v.m. backward compatibility met de OAS van ZORG-AB 2.6.2 worden Telecoms met type 'URL' vooralsnog niet opgeleverd</i>						
		Name	Beschrijving van of toelichting op het contactgegeven, bijvoorbeeld 'Algemeen'						
		Value	Het gegeven, bijvoorbeeld '012-3456789'						
		Active	Geeft aan of dit contactgegeven actief is						
		Preferred	Boolean dat gebruikt kan worden om aan te geven dat de voorkeur bestaat dat dit contactgegeven wordt gebruikt of weergegeven i.p.v. de andere contactgegevens						
		Timestamp	Datum en tijdstip van laatste wijziging						
Identification	Identificerend gegeven	Author	'Eigenaar' van het gegevensobject (Identification). Voor toelichting zie par. 2.4						
		Type	Type van het identificerende gegeven Voorbeelden: <table border="1" data-bbox="635 1691 1428 1870"> <tr> <td>Zorgaanbieder</td> <td>'URA', 'MedMij', 'AGB_Onderneming', 'AGB_Vestiging'</td> </tr> <tr> <td>Zorgaanbieder (Toestemmingen)</td> <td>'AGB_Zorgaanbieder'</td> </tr> <tr> <td>Zorgverlener</td> <td>'UZI', 'AGB_Zorgverlener', 'BIG'</td> </tr> </table>	Zorgaanbieder	'URA', 'MedMij', 'AGB_Onderneming', 'AGB_Vestiging'	Zorgaanbieder (Toestemmingen)	'AGB_Zorgaanbieder'	Zorgverlener	'UZI', 'AGB_Zorgverlener', 'BIG'
Zorgaanbieder	'URA', 'MedMij', 'AGB_Onderneming', 'AGB_Vestiging'								
Zorgaanbieder (Toestemmingen)	'AGB_Zorgaanbieder'								
Zorgverlener	'UZI', 'AGB_Zorgverlener', 'BIG'								
		Value	Het identificerende gegeven, bijvoorbeeld '12345678'						

		Description	Beschrijving van het identificerende gegeven						
		Active	Geeft aan of dit ID actief is. Inactieve ID's worden getoond bij de zorgaanbieder of -verlener maar worden niet geïndexeerd, d.w.z. als een ID op inactief wordt gezet wordt de zorgaanbieder of -verlener bijvoorbeeld niet opgeleverd wanneer op dit ID wordt gezocht. (Zodra het ID weer op actief wordt gezet is de zorgaanbieder of -verlener weer vindbaar onder het betreffende ID)						
		Timestamp	Datum en tijdstip van laatste wijziging						
Electronic Service	Superklasse van IHE Electronic Service, LSP Application en MedMij Gegevensdienst	Author	'Eigenaar' van het ElectronicService object. Voor toelichting zie par. 2.4						
		Timestamp	Datum en tijdstip van laatste wijziging						
IHE Electronic Service	Elektronische dienst	Type	Type elektronische dienst						
		Active	Geeft aan of deze elektronische dienst actief is						
		Value	ID van de elektronische dienst, bijvoorbeeld '12345678'						
		Address	Adres (endpoint) van de elektronische dienst						
		Description	Beschrijving van de elektronische dienst						
		Id	Berekende waarde, bevat de waarde van Type en de waarde van Value gescheiden door een dubbele punt (':')						
		Timestamp	Datum en tijdstip van laatste wijziging						
LSP Application		Active	Geeft aan of de LSP-applicatie actief is. Wordt via live koppeling uit LSP APR gehaald.						
		Deprecated vanaf ZORG-AB 2.6.5. Aanbevolen wordt om 'Status' te gebruiken. 'Active' wordt ondersteund tot 31-12-2021	De volgende waarden zijn mogelijk:						
			<table border="1"> <thead> <tr> <th>Waarde</th> <th>Toelichting</th> </tr> </thead> <tbody> <tr> <td>true</td> <td>Actief</td> </tr> <tr> <td>false</td> <td>Een status anders dan Actief (Gereed, Onderhoud, Storing of Afgesloten)</td> </tr> </tbody> </table>	Waarde	Toelichting	true	Actief	false	Een status anders dan Actief (Gereed, Onderhoud, Storing of Afgesloten)
Waarde	Toelichting								
true	Actief								
false	Een status anders dan Actief (Gereed, Onderhoud, Storing of Afgesloten)								
			Zowel actieve als inactieve LSP applicaties worden geïndexeerd/worden opgeleverd via de applicatie ID query's, bijvoorbeeld GET /zab/applicationId/[LSP Applicatie ID]						

		Address	Adres van de LSP-applicatie. Wordt via live koppeling uit LSP APR gehaald										
		Description	Beschrijving van de LSP-applicatie										
		ApplicationId	LSP applicatie ID										
		SystemRoles	Lijst (array) van 1 of meer systeemrollen. Wordt via live koppeling uit LSP APR gehaald										
		Conformances	Lijst (array) van 1 of meer Conformances, zie Conformance. Wordt via live koppeling uit LSP APR gehaald										
		Status	Status van de LSP-applicatie. Wordt via live koppeling uit LSP APR gehaald. De volgende waarden zijn mogelijk: ² <table border="1" data-bbox="635 660 1021 896"> <thead> <tr> <th>Waarde</th> <th>Toelichting</th> </tr> </thead> <tbody> <tr> <td>active</td> <td>Actief</td> </tr> <tr> <td>suspended</td> <td>Gereed of Onderhoud</td> </tr> <tr> <td>error</td> <td>Storing</td> </tr> <tr> <td>off</td> <td>Afgesloten</td> </tr> </tbody> </table>	Waarde	Toelichting	active	Actief	suspended	Gereed of Onderhoud	error	Storing	off	Afgesloten
Waarde	Toelichting												
active	Actief												
suspended	Gereed of Onderhoud												
error	Storing												
off	Afgesloten												
		Timestamp	Datum en tijdstip van laatste wijziging										
Conformance	LSP Conformance	InteractionId	Interactie ID, bijvoorbeeld 'COMT_IN113113NL'										
		Send	Boolean, bijvoorbeeld 'true'										
		Receive	Boolean, bijvoorbeeld 'true'										
MedMij Gegevensdienst	Zie par. 3.19 voor toelichting	GegevensdienstId	GegevensdienstId, bijvoorbeeld '4'										
		Weergavenaam	Weergavenaam corresponderend met GegevensdienstId, bijvoorbeeld 'Laboratoriumresultaten' <i>Toegevoegd in ZORG-AB 2.7.0</i>										
		Interfaceversies	Lijst van Interfaceversies, zie Interfaceversie <i>Toegevoegd in ZORG-AB 2.7.0</i>										
		AuthorizationEndpoint	Wordt (vooralsnog) niet gevuld										
		TokenEndpoint	Wordt (vooralsnog) niet gevuld										

² Gebaseerd op HL7 FHIR EndpointStatus: <http://hl7.org/fhir/STU3/valueset-endpoint-status.html>

		Abonneren	Wordt (vooralsnog) niet gevuld	
		Systeemrollen	Wordt (vooralsnog) niet gevuld	
		Timestamp	Datum en tijdstip van laatste wijziging	
Interfaceversie		Interfaceversie	Interfaceversie, bijvoorbeeld '1.3.0'	
Certificatie	Certificaat	Author	Niet geïmplementeerd	
		Type		
		Active		
		Description		
		Timestamp		
Credentia l	Kwalificatie (bijvoorbeeld academische titel of diploma)	Author	'Eigenaar' van het gegevensobject (Credential). Voor toelichting zie par. 2.4	
		Type		Type kwalificatie
		Active		Geeft aan of deze kwalificatie actief is
		Description		Beschrijving van of toelichting op de kwalificatie
		Timestamp		Datum en tijdstip van laatste wijziging
Attachme nt	Bijlage	Author	Niet geïmplementeerd	
		Description		
		MimeType		
		Data		
		Timestamp		
Type	Zorgaanbi edertype	Author	'Eigenaar' van het gegevensobject (Type). Voor toelichting zie par. 2.4	
		Type	De waardelijst/codesysteem waaruit de code afkomstig is, bijvoorbeeld: 'NICTIZ', 'VZVZ' of 'Vektis Zorgsoort'. (De code wordt opgeleverd in het attribuut 'Code', zie hieronder.) In het geval van waardelijst/codesysteem 'NICTIZ' wordt een code opgeleverd conform de volgende waardelijst: https://decor.nictiz.nl/decor/services/RetrieveValueSet?id=2.16.840.1.113883.2.4.3.11.60.1.11.1&effectiveDate=2020-05-18T12:30:00&prefix=nictiz2bbr-&format=html&collapsible=true&language=nl-NL&ui=nl-NL	

			<p>De FHIR ValueSet kan als volgt worden opgehaald:</p> <p>GET http://decor.nictiz.nl/fhir/3.0/public/ValueSet/2.16.840.1.113883.2.4.3.11.60.1.11.1--20200518123000</p> <p>GET http://decor.nictiz.nl/fhir/3.0/public/ValueSet/2.16.840.1.113883.2.4.3.11.60.1.11.1--20200518123000/\$expand</p> <p>GET http://decor.nictiz.nl/fhir/4.0/public/ValueSet/2.16.840.1.113883.2.4.3.11.60.1.11.1--20200518123000</p> <p>GET http://decor.nictiz.nl/fhir/4.0/public/ValueSet/2.16.840.1.113883.2.4.3.11.60.1.11.1--20200518123000/\$expand</p>
		Code	Code conform de waardelijst/codesysteem, bijvoorbeeld 'J8'
		DisplayName	Weergavenaam corresponderend met de code, bijvoorbeeld 'Openbare apotheek'
		Active	Geeft aan of de code actief is. Inactieve codes zijn zichtbaar maar worden niet geïndexeerd waardoor deze bijvoorbeeld niet gevonden worden via de zorgaanbiedertype-query (zie par. 3.11 en 4.8)
		Timestamp	Datum en tijdstip van laatste wijziging. Alleen-lezen waarde die wordt opgeleverd door ZORG-AB
SubOrganization	'SubOrganization'-relatie, de relatie tussen een (sub)organisatie en de (hoofd)organisatie waar deze een suborganisatie van is	Author	'Eigenaar' van het SubOrganization relatie object. Voor toelichting zie par. 2.4
		MainOrganizationId	ID van de hoofdorganisatie. Dit is het technische ID dat gevonden kan worden bij de organisatie in het JSON-veld '_id'
		SubOrganizationId	ID van de suborganisatie. Dit is het technische ID dat gevonden kan worden bij de organisatie in het JSON-veld '_id'
		Timestamp	Datum en tijdstip van laatste wijziging
WorksAt	'WorksAt'-relatie, de relatie tussen een individu	Author	'Eigenaar' van het WorksAt relatie object. Voor toelichting zie par. 2.4

	(zorgverlener) en de organisatie (zorgaanbieder) waarvoor hij/zij werkzaam is		
		OrganizationalId	ID van de organisatie (zorgaanbieder). Dit is het technische ID dat gevonden kan worden bij de organisatie in het JSON-veld '_id'
		IndividualId	ID van het individu (zorgverlener). Dit is het technische ID dat gevonden kan worden bij het individu in het JSON-veld '_id'
		Telecoms	Lijst (array) van 1 of meer Telecoms, zie Telecom. Dit attribuut is bedoeld voor het opslaan van intercollegiale telefoonnummers en e-mailadressen. Contactgegevens opgeslagen in dit attribuut zijn met de standaardinstellingen van ZORG-AB niet zichtbaar voor GBP's en PGO's. Zie ook par. 3.16
		Speciality	<p>Specialisme (rolcode of Vektis Zorgaanbodcode) van de zorgverlener bij deze zorgaanbieder, bijvoorbeeld '01.015' (rolcode) of '0100' (Vektis Zorgaanbodcode)</p> <p>Rolcodes worden opgeleverd conform de volgende waardelijst: https://decor.nictiz.nl/decor/services/RetrieveValueSet?id=2.16.840.1.113883.2.4.3.11.60.1.11.2&effectiveDate=2020-04-01T00:00:00&prefix=nictiz2bbr-&format=html&collapsable=true&language=nl-NL&ui=nl-NL</p> <p>De FHIR ValueSet kan als volgt worden opgehaald:</p> <p>GET http://decor.nictiz.nl/fhir/3.0/public/ValueSet/2.16.840.1.113883.2.4.3.11.60.1.11.2--20200401000000</p> <p>GET http://decor.nictiz.nl/fhir/3.0/public/ValueSet/2.16.840.1.113883.2.4.3.11.60.1.11.2--20200401000000/\$expand</p> <p>GET http://decor.nictiz.nl/fhir/4.0/public/ValueSet/2.16.840.1.113883.2.4.3.11.60.1.11.2--20200401000000</p> <p>GET http://decor.nictiz.nl/fhir/4.0/public/ValueSet/2.16.840.1.113883.2.4.3.11.60.1.11.2--20200401000000/\$expand</p> <p><i>I.v.m. backward compatibility met de OAS van ZORG-AB 2.6.2 worden Vektis Zorgaanbodcodes vooralsnog niet opgeleverd</i></p>
		Availability	De beschikbaarheid van de zorgverlener bij deze zorgaanbieder. Er wordt voorgesteld dat het iCalendar-formaat (RFC 5545) wordt gebruikt, maar dit wordt niet technisch afgedwongen. Het datatype is 'String'. Voor aanvullende details en een voorbeeld zie par. 3.17
		Timestamp	Datum en tijdstip van laatste wijziging

2.2 Toelichting op de Entiteit 'Party'

De entiteit Party is (zoals ook aangegeven in de tabel) een superclass van de entiteiten Organization en Individual. Dit betekent dat, als er een organisatie of individu wordt teruggegeven door ZORG-AB (zie Hoofdstuk 3 voor een voorbeeld), deze attributen van zowel de Party-entiteit als de Organisatie- of de Individu-entiteit kan bevatten. Ook bij het toevoegen van een organisatie of individu kunnen zowel de 'algemene' attributen van de Party-entiteit als de 'specifieke' attributen van de Organisatie- of Individu-entiteit meegegeven worden.

2.3 Hoofdletters en Kleine Letters

De namen van de attributen in de figuur en tabel beginnen met een hoofdletter (bijvoorbeeld 'StreetName'). In de JSON-body, die teruggegeven wordt bij het uitvoeren van een query (GET-request) op ZORG-AB, beginnen de namen van de attributen met een kleine letter (bijvoorbeeld 'streetName'). Ook bij het toevoegen of wijzigen van gegevens (POST of PATCH) moeten kleine letters gebruikt worden aan het begin van de naam van de attributen ('keys').

2.4 Author

2.4.1 Doeleinden

Het attribuut 'Author', in JSON berichten geschreven als 'author', geeft de 'eigenaar' van het gegevensobject aan.

Het attribuut is alleen-lezen.

De author wordt gebruikt voor de volgende doeleinden:

- Autorisatie. Zorgaanbieders/XIS-leveranciers kunnen 'eigen' gegevens die zijn toegevoegd via de API (native REST of FHIR) wijzigen maar een aantal gegevens dat automatisch is geïmporteerd vanuit externe gegevensbronnen/registers kan niet via de API worden gewijzigd. M.b.v. de author wordt bepaald welke gegevens muteerbaar zijn. Bijv. een adres dat is geïmporteerd vanuit het AGB-register (Vektis) heeft author 'Vektis'. Dit adres is niet muteerbaar via de API. Een adres dat door de zorgaanbieder/XIS is toegevoegd heeft author 'URA:nnnnnnn', bijv. 'URA:12345678'. Dit adres kan door de betreffende zorgaanbieder/XIS worden gemuteerd
- Bepaling welke gegevens moeten worden bijgewerkt bij het uitvoeren van de verschillende automatische imports (Vektis import, VZVZ BI import, UZI LDAP import en LSP import). De imports werken alleen 'eigen' gegevens bij, bijvoorbeeld de VZVZ BI import voegt adresgegevens toe met author 'VZVZ'. Vervolgens vervangt de VZVZ BI import, wanneer de adresgegevens in de gegevensbron wijzigen, alleen de adresgegevens met die author, 'VZVZ'. Evt. adresgegevens uit andere gegevensbronnen blijven in dit geval ongewijzigd. Bijv. adressen afkomstig uit Vektis hebben author 'Vektis' en adressen toegevoegd door zorgaanbieder/XIS hebben author 'URA:12345678'. Die adressen worden niet gewijzigd door de VZVZ BI import
- Use cases zoals: er zijn adresgegevens uit meerdere gegevensbronnen gekoppeld aan een zorgaanbieder en een bevragende partij wenst de adresgegevens op te halen die afkomstig zijn uit een bepaalde gegevensbron, bijvoorbeeld VZVZ BI. Het onderscheid kan in dit geval worden gemaakt o.b.v. de author, bijvoorbeeld 'VZVZ'

2.4.2 Author van Hoofdobject / Author van Gegevensobject

ZORG-AB bevat 'basisdata', d.w.z. een initiële set van zorgaanbieders en -verleners (Organizations en Individuals). Ook worden er zorgaanbieders en -verleners toegevoegd door automatische imports vanuit externe gegevensbronnen/registers. Bij deze zorgaanbieders en -verleners is de author van het hoofdobject 'Vektis', 'VZVZ' of 'UZI'.

Voorbeeld van een zorgaanbieder (Organization):

```
{
  ...
  "author": "Vektis",
  ...
}
```

In het bovenstaande voorbeeld wordt de author van het 'hoofdobject' getoond. In dit geval is de author 'Vektis'. Daarnaast hebben alle gekoppelde gegevensobjecten, bijv. Names, Addresses en ElectronicServices een author.

Gegevensobjecten die in ZORG-AB zijn toegevoegd door de automatisch imports vanuit externe gegevensbronnen/registers hebben een author die overeenkomt met die externe gegevensbron/register. Bijvoorbeeld een adres dat is geïmporteerd vanuit het AGB-register (Vektis) heeft author 'Vektis' zoals in het onderstaande voorbeeld:

(Zie de vetgedrukte regel.)

```
{
  ...
  "author": "Vektis",
  ...
  "addresses": [
    {
      "active": true,
      "address": "Orteliuslaan 1004\r\n3528BD UTRECHT",
      "author": "Vektis",
      "city": "UTRECHT",
      "country": "NEDERLAND",
      "firstLine": null,
      "geolocation": null,
      "postalCode": "3528BD",
      "state": "UT",
      "streetName": "Orteliuslaan",
      "streetNumber": "1004",
      "timestamp": "2020-08-24 13:39:19.118",
      "type": "Practice"
    }
  ],
  ...
}
```

Gegevensobjecten die zijn toegevoegd door de zorgaanbieder/XIS via de API (native REST of FHIR) hebben author 'URA:nnnnnnn' waarbij 'nnnnnnn' het URA-nummer is van de zorgaanbieder, bijvoorbeeld:

(Zie de vetgedrukte regel.)

```
{
  ...
  "author": "Vektis",
  ...
  "addresses": [
    {
      "active": true,
      "address": "Orteliuslaan 1004\r\n3528BD UTRECHT",
      "author": "Vektis",
      "city": "UTRECHT",
```

```

    "country": "NEDERLAND",
    "firstLine": null,
    "geolocation": null,
    "postalCode": "3528BD",
    "state": "UT",
    "streetName": "Orteliuslaan",
    "streetNumber": "1004",
    "timestamp": "2020-08-24 13:39:19.118",
    "type": "Practice"
  },
  {
    "active": true,
    "address": "Mendelweg 32\r\n2333CS LEIDEN",
    "author": "URA:12345678",
    "city": "LEIDEN",
    "country": "NEDERLAND",
    "firstLine": null,
    "geolocation": null,
    "postalCode": "2333CS",
    "state": "ZH",
    "streetName": "Mendelweg",
    "streetNumber": "32",
    "timestamp": "2020-08-24 13:39:19.118",
    "type": "Practice"
  }
],
...
}

```

Om precies te zijn bestaat de author in dit geval uit de prefix 'URA:' gevolgd door het URA-nummer dat is opgenomen in het Subject Alternative Name attribuut van het UZI-servercertificaat dat door de zorgaanbieder/XIS is gebruikt bij het versturen van het request waarmee het gegevensobject is toegevoegd/gewijzigd, bijvoorbeeld `POST [Base URL]/organizations/[ID]/addresses`.

2.4.3 Mogelijke Authors

Voor alle initieel geladen en automatisch geïmporteerde zorgaanbieders en -verleners in ZORG-AB geldt het volgende:

1. De author van het hoofdobject is 'Vektis', 'VZVZ' of 'UZI'
2. De gegevensobjecten die hieraan zijn gekoppeld (Names, Addresses etc.) hebben een author zoals hieronder aangegeven:

Type	Author
Gegevensobjecten afkomstig uit externe gegevensbron/register	'Vektis', 'VZVZ' of 'UZI'
LSP-applicaties (afkomstig uit LSP APR)	'LSP'
MedMij Gegevensdiensten	'MedMij'
Gegevensobjecten die door zorgaanbieder/XIS zijn toegevoegd via API (native REST of FHIR)	'URA:nnnnnnnn' bijv. 'URA:12345678'
Gegevensobjecten die door zorgaanbieder/XIS zijn toegevoegd via API (native REST of FHIR) met PKI-o-certificaat anders dan UZI-servercertificaat	'FQDN:[FQDN]'

2.4.4 Wijzigen van Gegevens

Niet mogelijk is:

- Het is niet mogelijk om de initieel geladen en automatisch geïmporteerde zorgaanbieders en -verleners, waarbij de author van het hoofdobject 'Vektis', 'VZVZ' of 'UZI' is, te wijzigen/verwijderen via de API.
- Het is ook niet mogelijk om bij deze zorgaanbieders en -verleners de gekoppelde gegevensobjecten met author 'Vektis', 'VZVZ', 'UZI', 'LSP' of 'MedMij' te wijzigen/verwijderen via de API.

Wel mogelijk is:

- Gegevensobjecten (Names, Addresses etc.) kunnen worden **toegevoegd** aan zorgaanbieders en -verleners door een zorgaanbieder/XIS via de API (native REST of FHIR) ondanks dat de author van het hoofdobject 'Vektis', 'VZVZ' of 'UZI' is. Dit wordt beschreven in de volgende paragraaf.
- Een zorgaanbieder/XIS kan deze toegevoegde, 'eigen' gegevensobjecten wijzigen/verwijderen. Dit zijn de gegevensobjecten met author 'URA:nnnnnnn', bijvoorbeeld 'URA:12345678'.
- Hetzelfde geldt voor zorgaanbieders en -verleners (Organizations en Individuals) die door de zorgaanbieder/XIS zijn toegevoegd via de API en author 'URA:nnnnnnn' hebben. Bijvoorbeeld afdelingen (een afdeling is een Organization), zoals beschreven in de volgende paragraaf. Deze kunnen door de zorgaanbieder/XIS worden gewijzigd/verwijderd via de API.

2.4.5 Toevoegen van Gegevens

Zoals hiervoor beschreven bevat ZORG-AB een initiële vulling en worden daarnaast via automatische imports zorgaanbieders en -verleners toegevoegd. Voor deze initieel geladen en automatisch geïmporteerde zorgaanbieders en -verleners is de author van het hoofdobject 'Vektis', 'VZVZ' of 'UZI'.

Zoals ook hierboven genoemd kunnen deze zorgaanbieders en -verleners niet worden gewijzigd/verwijderd.

Het is echter wel mogelijk om gegevensobjecten (Names, Addresses etc.) toe te voegen aan deze zorgaanbieders en -verleners via de API. Het is bijvoorbeeld mogelijk om een adres toe te voegen via een `POST [Base URL]/organizations/[ID]/addresses` request via de native REST API.

In dit geval moet bij het sturen van het request het URA-nummer van het gebruikte UZI-servercertificaat overeenkomen met het URA-nummer dat is gekoppeld aan de zorgaanbieder. (Om precies te zijn moet het URA-nummer dat is opgenomen in het Subject Alternative Name attribuut van het UZI-servercertificaat overeenkomen met de 'Identification' met type 'URA' in het 'Identifications' attribuut van de zorgaanbieder (Organization).)

Op een vergelijkbare manier kunnen Names, Telecoms, ElectronicServices etc. worden toegevoegd. De paden die daarvoor moeten worden gebruikt kunnen worden gevonden in het OAS (Swagger) bestand, zie par. 1.3.

Verder is het mogelijk om via de API zorgaanbieders en -verleners (Organizations en Individuals) toe te voegen. Het is bijvoorbeeld mogelijk om een afdeling toe te voegen en te koppelen aan een bestaande zorgaanbieder. Een afdeling is ook een Organization. De volgende stappen moet hiervoor worden gevolgd:

- Afdeling toevoegen via `POST [Base URL]/organizations` met in request body de gegevens van de afdeling
- 'SubOrganization' relatie toevoegen via `POST [Base URL]/subOrganization` met de volgende request body:

```
{
  "mainOrganizationId": "[ID van Zorgaanbieder]",
  "subOrganizationId": "[ID van Afdeling]"
}
```

Bij Organizations en Individuals die zijn toegevoegd door een zorgaanbieder/XIS via de API, inclusief afdelingen (een afdeling is een Organization), is de author van het hoofdobject 'URA:nnnnnnn', bijv. 'URA:12345678'.

Dezelfde mogelijkheden worden geboden via de FHIR API.

2.5 Suborganisaties

In het geval een zorgaanbieder 'locaties' heeft is er sprake van een structuur zoals hieronder aangegeven:

- Hoofdorganisatie:

ID	123
URA-nummer	12345678
Naam	Stichting Gezondheidscentrum Janssen

- Suborganisatie:

ID	456
AGB-code	11111111
Zorgaanbiedertype	J8 (=Openbare apotheek)
Naam	Stichting Gezondheidscentrum Janssen (Apotheek)
LSP-applicatie	88888888

- Suborganisatie:

ID	789
AGB-code	22222222
Zorgaanbiedertype	Z3 (=Huisartspraktijk)
Naam	Stichting Gezondheidscentrum Janssen (Huisartsenpraktijk)
LSP-applicatie	99999999

In dit geval bestaat de zorgaanbieder dus uit een 'hoofdorganisatie' en 'suborganisaties'. In termen van het ZORG-AB datamodel zijn zowel de hoofdorganisatie als suborganisaties instanties van Organization. M.a.w. er zijn 3 Organization objecten: 123, 456 en 789.

Deze instanties zijn gekoppeld d.m.v. de SubOrganization relation (vergelijkbaar met de WorksAt relation) die wordt getoond in **Figuur 1**.

Er zijn specifieke gegevens gekoppeld aan de suborganisaties: een specifieke AGB-code, zorgaanbiedertype, naam en LSP-applicatie.

NB

- Suborganisaties hebben altijd een 'eigen' naam en/of zorgaanbiedertype. In dit voorbeeld hebben de suborganisaties ook een 'eigen' AGB-code en LSP-applicatie. Echter niet in alle gevallen hebben suborganisaties een 'eigen' AGB-code of LSP-applicatie. Er zijn gevallen waarin dezelfde AGB-code of LSP-applicatie is gekoppeld aan meerdere suborganisaties.
- Een suborganisatie kan meerdere zorgaanbiedertypes hebben en meerdere LSP-applicaties.

Om gegeven een hoofdorganisatie de suborganisaties op te vragen kan de volgende call worden uitgevoerd:

```
GET https://zab.test.lsp.aorta-zorg.nl/zab/organizations/[Hoofdorganisatie ID]/subOrganizations
```


Bijvoorbeeld:

```
GET https://zab.test.lsp.aorta-
zorg.nl/zab/organizations/123/subOrganizations
```

Er wordt dan een JSON array teruggegeven zoals hieronder getoond:

```
[
  {
    "_self": "/organizations/456",
    "_className": "CSC.HPD.Organization",
    "id": "456",
    "addresses": [
      {
        "active": true,
        "address": "Orteliuslaan 1004\r\n3528 BD UTRECHT",
        "author": "VZVZ",
        "city": "UTRECHT",
        "country": null,
        "firstLine": null,
        "geolocation": null,
        "postalCode": "3528 BD",
        "state": null,
        "streetName": "Orteliuslaan",
        "streetNumber": "1004",
        "timestamp": "2021-06-24 09:42:09.1",
        "type": "Practice"
      }
    ],
    "applicationIds": [
      "88888888"
    ],
    "attachments": null,
    "author": "VZVZ",
    "comment": null,
    "credentials": null,
    "displayName": "Stichting Gezondheidscentrum Janssen (Apotheek)",
    "electronicServices": [
      {
        "_className": "CSC.HPD.LSP.Application",
        "active": true,
        "address": "janssen.voorbeeld.nl",
        "applicationId": "88888888",
        "author": "LSP",
        "conformances": [
          {
            "interactionId": "COMT_IN113113NL",
            "receive": true,
            "send": true
          },
          ...
        ],
        "description": null,
        "status": "active",
        "systemRoles": [
          "AllPurpose"
        ],
        "timestamp": "2021-06-24 09:42:09.1"
      }
    ],
    "identifications": [
      {
        "active": true,
        "author": "VZVZ",
        "description": null,
        "timestamp": "2021-06-24 09:42:09.1",
        "type": "AGB_Zorgaanbieder",
        "value": "11111111"
      }
    ]
  }
]
```

```

"names": [
  {
    "active": true,
    "author": "VZVZ",
    "fullName": "Stichting Gezondheidscentrum Janssen (Apotheek)",
    "preferred": false,
    "timestamp": "2021-06-24 09:42:09.1",
    "type": "Display"
  }
],
"speciality": null,
"telecoms": null,
"timestamp": "2021-06-24 09:42:09.1",
"type": null,
"types": [
  {
    "active": true,
    "author": "VZVZ",
    "code": "Z001",
    "displayName": "Apotheek",
    "timestamp": "2021-06-24 09:42:09.1",
    "type": "VZVZ"
  },
  {
    "active": true,
    "author": "VZVZ",
    "code": "J8",
    "displayName": "Openbare apotheek",
    "timestamp": "2021-06-24 09:42:09.1",
    "type": "NICTIZ"
  }
],
"ura": null
},
{
  "_self": "/organizations/789",
  "_className": "CSC.HPD.Organization",
  "_id": "789",
  "addresses": [
    {
      "active": true,
      "address": "Orteliuslaan 1004\r\n3528 BD UTRECHT",
      "author": "VZVZ",
      "city": "UTRECHT",
      "country": null,
      "firstLine": null,
      "geolocation": null,
      "postalCode": "3528 BD",
      "state": null,
      "streetName": "Orteliuslaan",
      "streetNumber": "1004",
      "timestamp": "2021-06-24 09:42:09.108",
      "type": "Practice"
    }
  ],
  "applicationIds": [
    "99999999"
  ],
  "attachments": null,
  "author": "VZVZ",
  "comment": null,
  "credentials": null,
  "displayName": "Stichting Gezondheidscentrum Janssen (Huisartsenpraktijk)",
  "electronicServices": [
    {
      "_className": "CSC.HPD.LSP.Application",
      "active": true,
      "address": "janssen.voorbeeld.nl",
      "applicationId": "99999999",
      "author": "LSP",
      "conformances": [

```

```

        "interactionId": "COMT_IN113113NL",
        "receive": true,
        "send": true
    },
    ...
    ],
    "description": null,
    "status": "active",
    "systemRoles": [
        "AllPurpose"
    ],
    "timestamp": "2021-06-24 09:42:09.108"
}
],
"identifications": [
    {
        "active": true,
        "author": "VZVZ",
        "description": null,
        "timestamp": "2021-06-24 09:42:09.108",
        "type": "AGB_Zorgaanbieder",
        "value": "22222222"
    }
],
"names": [
    {
        "active": true,
        "author": "VZVZ",
        "fullName": "Stichting Gezondheidscentrum Janssen
(Huisartsenpraktijk)",
        "preferred": false,
        "timestamp": "2021-06-24 09:42:09.108",
        "type": "Display"
    }
],
"speciality": null,
"telecoms": null,
"timestamp": "2021-06-24 09:42:09.108",
"type": null,
"types": [
    {
        "active": true,
        "author": "VZVZ",
        "code": "Z004",
        "displayName": "Huisartsenpraktijk",
        "timestamp": "2021-06-24 09:42:09.108",
        "type": "VZVZ"
    },
    {
        "active": true,
        "author": "VZVZ",
        "code": "Z3",
        "displayName": "Huisartspraktijk (zelfstandig of groepspraktijk)",
        "timestamp": "2021-06-24 09:42:09.108",
        "type": "NICTIZ"
    }
],
"ura": null
}
]

```

Om gegeven een suborganisatie de hoofdorganisatie op te vragen kan de volgende call worden uitgevoerd:

```
GET https://zab.test.lsp.aorta-
zorg.nl/zab/organizations/[Suborganisatie ID]/mainOrganization
```

Bijvoorbeeld:

```
GET https://zab.test.lsp.aorta-  
zorg.nl/zab/organizations/456/mainOrganization
```

Ook in dat geval wordt een JSON array teruggegeven.

2.5.1 Suborganisatie Zoeken o.b.v. LSP Applicatie ID

Om een suborganisatie te zoeken op LSP applicatie ID kan de volgende call worden uitgevoerd:

```
GET https://zab.test.lsp.aorta-zorg.nl/zab/applicationId-v2/ [LSP  
Applicatie ID]
```

Bijvoorbeeld:

```
GET https://zab.test.lsp.aorta-zorg.nl/zab/applicationId-v2/88888888
```

Er wordt dan een JSON array teruggegeven met daarin zowel de hoofdorganisatie als suborganisaties.

Zie ook Bijlage 1.

Deze aanroep kan ook worden gebruikt om zorgaanbieders zonder 'locaties' te vinden. De array bevat in dat geval 1 organisatie (i.p.v. een hoofdorganisatie en suborganisaties).

3 Gebruik van de Native REST Interface

NB:

- Voor details m.b.t. het verbinden met ZORG-AB (zoals het endpoint van de test- en productieomgeving) zie par. 1.2
- Er zijn OAS (Swagger) web service definities beschikbaar, zie par. 1.3
- Dit hoofdstuk is bedoeld als toelichting op de native REST interface en met name op het gebruik van de queryparameter \$filter aangezien dit niet in de OAS (Swagger) web service definities is vastgelegd
- Voor details m.b.t. de FHIR-interface zie hoofdstuk 4

3.1 Resources

Hieronder wordt een overzicht gegeven van de resources:

Een volledige specificatie kan worden gevonden in het OAS (Swagger) bestand, zie par. 1.3.

Resource	Toelichting
/organizations	Voor het uitvoeren van een query (GET) op de zorgaanbieders (Organizations) in ZORG-AB. Zie voor verdere toelichting par. 3.4 en 3.5. Ook voor het toevoegen/wijzigen/verwijderen van zorgaanbieders (POST/PATCH/DELETE).
/organizations/_summary	Idem als /organizations met de volgende uitzonderingen: <ul style="list-style-type: none"> • Er wordt een summary object (subset van de beschikbare velden) teruggegeven • Op dit pad kan de zorgaanbiedertypequery worden uitgevoerd, bijvoorbeeld: <code>GET [Base URL]/organizations/_summary?\$filter=type eq 'JGZ'</code> Zie voor details par. 3.11
/individuals	Voor het uitvoeren van een query (GET) op de zorgverleners (Individuals) in ZORG-AB. Zie voor verdere toelichting par. 3.4 en 3.5. Ook voor het toevoegen/wijzigen/verwijderen van zorgverleners (POST/PATCH/DELETE).
/identifications/[Type]:[Value]	Voor het zoeken van een enkele zorgaanbieder of -verlener op een unieke ID zoals URA-, UZI-, KVK- of BIG-nummer, AGB-code of MedMij zorgaanbiedernaam. Het type en de waarde, [Type] en [Value], moeten meegegeven worden in het pad, gescheiden door een dubbele punt (':'). Voorbeelden Zorgaanbieder <ul style="list-style-type: none"> • Zoek zorgaanbieder op URA-nummer: <code>GET [Base URL]/identifications/URA:12345678</code>

	<ul style="list-style-type: none"> • Zoek zorgaanbieder op MedMij zorgaanbiedernaam: <pre>GET [Base URL]/identifications/MedMij:apotheekjanssen@medmij</pre> • Zoek zorgaanbieder op AGB-code: <pre>GET [Base URL]/identifications/AGB_Onderneming:12345678</pre> <p>Of: <pre>GET [Base URL]/identifications/AGB_Vestiging:12345678</pre> </p> • Zoek zorgaanbieder op KVK-nummer: <pre>GET [Base URL]/identifications/KVK:12345678</pre> <p>Voorbeeld Zorgaanbieder (Toestemmingen)</p> <p>Zoek zorgaanbieder op AGB-code van het type AGB_Zorgaanbieder:</p> <p>Dit zijn AGB-codes die in ZORG-AB worden geïmporteerd vanuit het BI-bestand van VZVZ t.b.v. 'toestemmingen'.</p> <pre>GET [Base URL]/identifications/AGB_Zorgaanbieder:12345678</pre> <p>Voorbeelden Zorgverlener</p> <ul style="list-style-type: none"> • Zoek zorgverlener op UZI-nummer: <pre>GET [Base URL]/identifications/UZI:123456789</pre> • Zoek zorgverlener op AGB-code: <pre>GET [Base URL]/identifications/AGB_Zorgverlener:12345678</pre> • Zoek zorgverlener op BIG-nummer: <pre>GET [Base URL]/identifications/BIG:12345678910</pre> <p>Response</p> <p>Alle <code>GET [Base URL]/identifications/[Type]:[Value]</code> query's kunnen technisch gezien een zorgaanbieder (Organization) of een zorgverlener (Individual) opleveren ongeacht het type waarop wordt gezocht.</p> <p>Een zorgaanbieder of -verlener wordt om precies te zijn opgeleverd als het attribuut Identifications een Identification bevat waarvan het attribuut Type en Value overeenkomen met het in de query aangegeven Type en Value.</p> <p>Er wordt precies 1 JSON object opgeleverd of een HTTP statuscode 404 als er geen zorgaanbieder of -verlener is gevonden die voldoet aan de query.</p>
<p><code>/ura/[URA-nummer]</code></p>	<p>Leverd zorgaanbieder (Organization) op met het aangegeven URA-nummer.</p> <p>Equivalent aan <code>/identifications/URA:[URA-nummer]</code></p>

<code>/uzi/[UZI-nummer]</code>	<p>Leverd zorgverlener (Individual) op met het aangegeven UZI-nummer.</p> <p>Equivalent aan <code>/identifications/UZI:[UZI-nummer]</code></p>
<code>/applicationId/[LSP Applicatie ID]</code>	<p>Leverd zorgaanbieder (Organization) op waaraan een LSP applicatie is gekoppeld met het aangegeven LSP Applicatie ID.</p> <p>Er wordt precies 1 JSON object opgeleverd of een HTTP statuscode 404 als er geen zorgaanbieder is gevonden die voldoet aan de query.</p>
<code>/applications/[LSP Applicatie ID]</code>	<p>Leverd LSP applicatie (LSP Application) op met het aangegeven LSP Applicatie ID.</p> <p>Er wordt precies 1 JSON object opgeleverd of een HTTP statuscode 404 als er geen LSP applicatie is gevonden die voldoet aan de query.</p>
<code>/organizations/[ID]/addresses</code>	T.b.v. toevoegen (POST) van een adres aan een zorgaanbieder
<code>/organizations/[ID]/addresses/[Index]</code>	T.b.v. wijzigen/verwijderen (PATCH/DELETE) van een adres van een zorgaanbieder. [Index] is de op 1 gebaseerde positie van het adres
Er zijn vergelijkbare resources voor names, telecoms, services etc. Zie OAS (Swagger) bestand.	
<code>/individuals/[ID]/addresses</code>	T.b.v. toevoegen (POST) van een adres aan een zorgverlener
<code>/individuals/[ID]/addresses/[Index]</code>	T.b.v. wijzigen/verwijderen (PATCH/DELETE) van een adres van een zorgverlener. [Index] is de op 1 gebaseerde positie van het adres
Er zijn vergelijkbare resources voor names, telecoms etc. Zie OAS (Swagger) bestand.	
<code>/subOrganization</code>	T.b.v. toevoegen/wijzigen/verwijderen SubOrganization relatie
<code>/worksAt</code>	T.b.v. toevoegen/wijzigen/verwijderen WorksAt relatie
<code>/organizations/[ID]</code>	Geeft de zorgaanbieder (Organization) met het aangegeven ID. Dit betreft het technische ID dat door de ZORG-AB API wordt opgeleverd
<code>/organizations/[ID]/_summary</code>	Idem als <code>/organizations/[ID]</code> met als uitzondering dat een summary object wordt opgeleverd
<code>/individuals/[ID]</code>	Geeft de zorgverlener (Individual) met het aangegeven ID. Dit betreft het technische ID dat door de ZORG-AB API wordt opgeleverd
<code>/subOrganization/[ID]</code>	Geeft de SubOrganization relatie met het aangegeven ID. Dit betreft het technische ID dat door de ZORG-AB API wordt opgeleverd

/worksAt/[ID]	Geeft de WorksAt relatie met het aangegeven ID. Dit betreft het technische ID dat door de ZORG-AB API wordt opgeleverd
---------------	--

De resources /organizations, /individuals, /subOrganization en /worksAt komen overeen met resp. de entiteiten Organization, Individual, SubOrganization en WorksAt van het ZORG-AB datamodel (zie Figuur 1).

3.2 HTTP-methoden

De volgende HTTP-methoden worden ondersteund:

- GET, voor het uitvoeren van query's
- POST, voor het toevoegen van resources
- PATCH, voor het wijzigen van bestaande resources
- DELETE, voor het verwijderen van resources

De HTTP-methode PUT wordt niet ondersteund door de native REST interface.

3.3 Queryparameters

De volgende queryparameters kunnen gebruikt worden in combinatie met de resources /organizations en /individuals:

- \$filter
- \$search **Toegevoegd in ZORG-AB 2.6.0**
- \$select
- \$expand³

Deze queryparameters zijn geïmplementeerd conform een subset van de OData 4.0 URL-conventie.⁴

De queryparameters worden beschreven in de volgende paragrafen.

3.4 Voorgedefinieerde Query's (\$filter)

Om te zoeken via de voorgedefinieerde query's moet de queryparameter \$filter worden gebruikt zoals in het volgende voorbeeld:

```
https://zab.test.lsp.aorta-zorg.nl/zab/organizations?$filter=name eq 'Apotheek Janssen' and city eq 'Utrecht'
```

Dit filter zorgt ervoor dat er gezocht wordt op zorgaanbieders met de naam 'Apotheek Janssen' en plaats 'Utrecht'.

Hieronder wordt een lijst gegeven van de combinaties van zoekvelden die, per resource, gebruikt kunnen worden in de \$filter queryparameter:

(Het is zowel mogelijk om Engelse als Nederlandse veldnamen te gebruiken.)

³ Niet geïmplementeerd in de huidige versie (HPD 1.0)

⁴ Zie: <http://docs.oasis-open.org/odata/odata/v4.0/odata-v4.0-part2-url-conventions.html>

Resource	Zoekvelden	Mogelijkheid 1 – Engelse Veldnamen	Mogelijkheid 2 – Nederlandse Veldnamen
/organizations + /organizations/_summary	Naam en Plaats	name eq 'Apotheek Janssen' and city eq 'Utrecht'	naam eq 'Apotheek Janssen' and plaats eq 'Utrecht'
	Naam en Specialisme	name eq 'Apotheek Janssen' and speciality eq 'Voorbeeld'	naam eq 'Apotheek Janssen' and specialisme eq 'Voorbeeld'
	Specialisme en Plaats	speciality eq 'Voorbeeld' and city eq 'Utrecht'	specialisme eq 'Voorbeeld' and plaats eq 'Utrecht'
	Specialisme en Postcode	speciality eq 'Voorbeeld' and postalcode eq '3528 BD'	specialisme eq 'Voorbeeld' and postcode eq '3528 BD'
	LSP Applicatie-ID	application eq '12345678' of applicationid eq '12345678'	
	URA	ura eq '12345678'	
/organizations/_summary	Type	type eq 'JGZ' Zie voor details par. 3.11.	
/individuals	Naam en Plaats	name eq 'Jan van het Zand' and city eq 'Utrecht'	naam eq 'Jan van het Zand' and plaats eq 'Utrecht'
	Naam en Specialisme	name eq 'Jan van het Zand' and speciality eq 'Apotheker'	naam eq 'Jan van het Zand' and specialisme eq 'Apotheker'
	Specialisme en Plaats	speciality eq 'Apotheker' and city eq 'Utrecht'	specialisme eq 'Apotheker' and plaats eq 'Utrecht'
	Specialisme en Postcode	speciality eq 'Apotheker' and postalcode eq '3528 BD'	specialisme eq 'Apotheker' and postcode eq '3528 BD'
	UZI	uzi eq '123456789'	
	Naam en Organisatie	name eq 'Jan van het Zand' and organization eq 'Apotheek Janssen'	naam eq 'Jan van het Zand' and organisatie eq 'Apotheek Janssen'
	Specialisme en Organisatie	speciality eq 'Apotheker' and organization eq 'Apotheek Janssen'	specialisme eq 'Apotheker' and organisatie eq 'Apotheek Janssen'

/identifications/[Type]:[Value]	<p>T.b.v. zoeken op ID zoals URA-, UZI-, KVK- of BIG-nummer, AGB-code of MedMij zorgaanbiedernaam.</p> <p>Moet gebruikt worden zonder \$filter.</p> <p>Zie par. 3.1 voor details.</p>
---------------------------------	--

Wat betreft zorgaanbieders (Organizations):

Bij het uitvoeren van een voorgedefinieerde query met de 'name' parameter moet in de 'name' parameter in principe de volledige naam van de zorgaanbieder worden gebruikt. Bijvoorbeeld in het geval van zorgaanbieder 'J.J. Janssen' moet in de 'name' parameter worden gebruikt 'J.J. Janssen' om de betreffende zorgaanbieder te vinden. Hierop zijn twee uitzonderingen:

- Een aantal specifieke voor- en achtervoegsels hoeft niet te worden gebruikt in de query, zie hieronder
- Het is mogelijk om de functie 'StartsWith' te gebruiken, zie par. 3.12

Een aantal voor- en achtervoegsels wordt bij zowel het indexeren als het zoeken automatisch verwijderd. Het betreft de volgende voor- en achtervoegsels:

Voorvoegsels:

- ZIEKENHUISAPOTHEEK
- APOTHEEK
- HUISARTSPRAKTIJK
- HUISARTSENPRAKTIJK
- HUISARTSENPOST
- ZIEKENHUIS
- DOKTERSDIENST

Achtervoegsels:

- STICHTING
- BV
- B.V.
- U.A.
- VOF
- V.O.F.
- CV
- C.V.
- COÖPERATIEVE
- COÖPERATIE

Dit betekent dat bij zoeken op bijvoorbeeld 'J.J. Janssen' er ook een match is met 'Apotheek J.J. Janssen B.V.' en, vice versa, dat er bij zoeken op 'Apotheek J.J. Janssen B.V.' een match is met 'J.J. Janssen'.

Wat betreft zorgverleners (Individuals):

Bij het uitvoeren van een voorgedefinieerde query met de 'name' parameter moet in de 'name' parameter in principe de achternaam van de zorgverlener worden gebruikt zonder voorvoegsel ('de', 'van' etc.).

Initialen, voornamen en voorvoegsels worden echter bij zowel het indexeren als het zoeken automatisch verwijderd.

Dit betekent dat bij zoeken op bijvoorbeeld 'Boer' er ook een match is met 'J.J. de Boer' en, vice versa, dat er bij zoeken op 'J.J. de Boer' een match is met 'Boer'.

De hierboven bij zorgaanbieders genoemde lijst van voor- en achtervoegsels ('ZIEKENHUISAPOTHEEK', 'APOTHEEK' etc.) wordt bij zorgverleners niet toegepast.

Bij zorgverleners kan ook gebruik worden gemaakt van de functie 'StartsWith', zie par. 3.12.

Gender en Language:

Daarnaast kunnen bij het zoeken van een zorgverlener, in combinatie met één van de in de bovenstaande tabel genoemde combinaties, de zoekvelden 'gender' (of 'sex' of 'geslacht') en/of 'language' (of 'taal') gebruikt worden, bijvoorbeeld:

```
https://zab.test.lsp.aorta-zorg.nl/zab/individuals?$filter=speciality eq 'Apotheker' and postalcode eq '3528 BD' and gender eq 'male'
```

Voor de 'gender' parameter moet een van de volgende waardes worden gebruikt:⁵

- male
- female
- other
- unknown

Voor de 'languages' parameter moet een BCP-47 code, bijv. 'nl' of 'nl-BE', worden gebruikt.

Escape van Apostrof:

Een apostrof, bijvoorbeeld in een plaatsnaam zoals 's-Gravenhage, moet worden geëscaped d.m.v. een backslash zoals in het volgende voorbeeld:

```
https://zab.test.lsp.aorta-zorg.nl/zab/organizations?$filter=name eq 'Apotheek Janssen' and city eq '\s-Gravenhage'
```

3.5 Vrije Tekst Query's (\$search)

Toegevoegd in ZORG-AB 2.6.0

Via de parameter \$search kunnen vrije tekst query's worden uitgevoerd. Hiermee kan op een zoekmachine-achtige manier worden gezocht, d.w.z. vergelijkbaar met hoe een zoekopdracht via een internetzoekmachine wordt uitgevoerd. Hiermee biedt ZORG-AB een alternatief voor het zoeken via de voorgedefinieerde query's (via \$filter, zie vorige paragraaf). Er is een aantal verschillen tussen beide manieren van zoeken:

- \$filter kan gebruikt worden om gericht en efficiënt te zoeken en ondersteunt het zoeken op een aantal velden dat met \$search niet doorzocht kan worden zoals postcode. Via \$filter kan ook gefilterd worden op geslacht en taal. Bij het zoeken via \$filter wordt het Soundex algoritme toegepast.
- \$search kan gebruikt worden om op een zoekmachine-achtige manier te zoeken. \$search stelt minder eisen aan het formaat van de zoekexpressie en maakt gebruik van 'fuzzy search' om aanvullende resultaten te vinden. Het biedt ook een aantal geavanceerde mogelijkheden, zoals het kunnen gebruiken van wild cards en synoniemen (zie het vervolg van deze paragraaf voor een toelichting).

Een voorbeeld van het zoeken via \$search:

```
https://zab.test.lsp.aorta-zorg.nl/zab/organizations?$search=janssen utrecht
```

Hiermee wordt gezocht naar zorgaanbieders (Organizations) die matchen met de zoekexpressie 'janssen utrecht'.

Nog een voorbeeld:

⁵ Gebaseerd op HL7 FHIR AdministrativeGender: <http://hl7.org/fhir/STU3/valueset-administrative-gender.html>

```
https://zab.test.lsp.aorta-zorg.nl/zab/individuals?$search=apotheker utrecht
```

Hiermee wordt gezocht naar zorgverleners (Individuals) die matchen met de zoekexpressie 'apotheker utrecht'.

De opgeleverde resultaten worden aflopend gesorteerd op relevantie volgens een door het zoekalgoritme bepaalde rank. (Er wordt in de tweede plaats oplopend gesorteerd op ID, d.w.z. bij resultaten met dezelfde rank wordt het record met het laagste ID het eerste opgeleverd.)

Er worden maximaal 100 resultaten opgeleverd.

Er wordt gebruik gemaakt van 'fuzzy search' om aanvullende resultaten te vinden. De zoektechnologie die ZORG-AB gebruikt maakt daarvoor gebruik van het Levenshtein algoritme. Hierdoor levert bijvoorbeeld 'janssen' ook een match op met 'jansen'. Dit is vergelijkbaar met het Soundex algoritme dat geïmplementeerd is bij de voorgedefinieerde query's (\$filter), zie de vorige paragraaf. De 'normale' matches worden eerst opgeleverd en daarna de 'fuzzy search' matches tot het maximum van 100 resultaten is bereikt.

Er kan gebruik worden gemaakt van wild cards (*) en logische operatoren (AND, OR en NOT). Voorbeelden:

```
https://zab.test.lsp.aorta-zorg.nl/zab/organizations?$search=jan* utrecht
```

```
https://zab.test.lsp.aorta-zorg.nl/zab/organizations?$search=(janssen or pieterssen) utrecht
```

De gegevens die doorzocht worden zijn zorgaanbiedertype(s), namen, plaatsen en specialisme die zijn gekoppeld aan de Organization/Individual. Om precies te zijn betreft dit de volgende velden van het ZORG-AB datamodel (zie par. 2.1):

Organization/Individual Veld	Datatype	Sub-veld
Types <i>Toegevoegd in ZORG-AB 2.6.2</i>	Type	Code en/of DisplayName
Names	OrganizationName/IndividualName	FullName
Addresses	Address	City
Speciality	String	-

NB:

- Voor zorgaanbieders (Organizations) is op dit moment het veld Types gevuld. Organizations beschikken technisch gezien ook over het veld Speciality maar dat wordt op dit moment niet gebruikt voor zorgaanbieders.
- Voor zorgverleners (Individuals) is op dit moment het veld Speciality gevuld. Individuals beschikken technisch gezien ook over het veld Types maar dat wordt op dit moment niet gebruikt voor zorgverleners.
- Voor IndividualName geldt dat als FullName niet expliciet gevuld is deze wordt opgebouwd uit FirstName, Prefix en FamilyName.
- Het veld Speciality van zorgverleners (Individuals) is in ZORG-AB gevuld met een rolcode volgens [deze](#) definitie. Deze rolcode wordt door een automatische import die dagelijks wordt uitgevoerd opgehaald uit UZI LDAP. In de vrije tekst query moet niet de code worden gebruikt, zoals 17.000, maar de weergavenaam, zoals Apotheker. (Dit is niet case sensitive en vanwege fuzzy search mogen enkele letters afwijken.)

Het is tevens mogelijk gemaakt om synoniemen van bepaalde woorden te gebruiken in query's uitgevoerd via \$search. Hiermee is o.a. het probleem opgelost dat zoeken op 's-Gravenhage geen match oplevert met Den Haag en vice versa. Hiervoor wordt door ZORG-AB een synoniementabel gebruikt. (Dit is een interne tabel,

deze is niet toegankelijk via de web service interfaces.) Initieel zijn de volgende plaatsnamen geladen in de synoniementabel:

- 's-Gravenhage / Den Haag
- 's-Hertogenbosch / Den Bosch
- Gorinchem / Gorcum / Gorkum

Zoekopdrachten uitgevoerd via de \$search parameter zijn niet case sensitive. Er mogen leestekens gebruikt worden rondom woorden (en streepjes en punten in woorden), deze worden genegeerd.

Aanvullende aandachtspunten zijn:

- Woorden moeten worden gescheiden door spaties. Aan elkaar geschreven woorden, bijvoorbeeld 'janssenutrecht', worden beschouwd als één woord en leveren daarom mogelijk geen match op. (De zoektechnologie die door ZORG-AB wordt gebruikt ondersteunt wel het ontleden van woorden maar dit is nog niet geïmplementeerd in ZORG-AB.)
- Een uitzondering daarop is 'huisarts'. Dit woord wordt als een synoniem beschouwd van 'huisartspraktijk' en 'huisartsenpraktijk'. (Ook dit is gerealiseerd via de eerder genoemde synoniementabel.) Dit geldt andersom ook, waardoor bij zoeken op bijvoorbeeld 'huisartspraktijk' er toch een match plaatsvindt met 'huisarts'.
- Een zorgaanbieder/zorgverlener (Organization/Individual) wordt als match beschouwd als alle woorden gevonden zijn in de eerder genoemde velden: zorgaanbiedertype(s), namen, plaatsen en specialisme. Anders gezegd, spaties zijn impliciet 'AND' operatoren. Wanneer er 'extra' woorden worden gebruikt, zoals bijvoorbeeld een straatnaam ('janssen orteliuslaan utrecht'), vindt er daarom mogelijk geen match plaats.

De \$search parameter kan eventueel gebruikt worden i.c.m. \$select zoals beschreven in de volgende paragraaf. \$search kan niet gebruikt worden i.c.m. \$filter.

3.6 Gecombineerde Query's (\$search i.c.m. \$filter)

Toegevoegd in ZORG-AB 2.6.6

Het is niet mogelijk om \$search te combineren met \$filter, uitgezonderd een aantal specifieke query's dat in deze paragraaf wordt beschreven.

GET /organizations:

\$filter kan worden gecombineerd met \$search maar in dit geval wordt m.b.t. de \$filter parameter alleen het filteren op zorgaanbiedertype ondersteund. (Om precies te zijn kunnen in dit geval in \$filter alleen een of meerdere 'eq' expressies worden gebruikt die betrekking hebben op het veld 'type' en geen expressies die betrekking hebben op andere velden.) Voorbeeld:

```
GET https://zab.test.lsp.aorta-  
zorg.nl/zab/organizations?$filter=type eq  
'NICTIZ:Z3' &$search=janssen
```

(NB De zorgaanbiedertypequery (par. 3.11) moet worden uitgevoerd op de resource /organizations/_summary en levert voor elke gevonden zorgaanbieder een summary object op, dat is bij deze query niet het geval.)

Hiermee wordt gezocht naar zorgaanbieders die overeenkomen met het zoekwoord 'janssen' zoals aangegeven in de \$search parameter. Wat dit betreft is deze query een vrije-tekst query zoals beschreven in de vorige paragraaf. Er wordt echter in aanvulling daarop het zoekcriterium toegepast dat is aangegeven in de \$filter parameter, d.w.z. dat het zorgaanbiedertype gelijk moet zijn aan 'NICTIZ:Z3' (Huisartspraktijk (zelfstandig of groepspraktijk)).

Het is bij deze query mogelijk om te zoeken op meerdere zorgaanbiedertypes door de operator 'or' te gebruiken in de \$filter parameter zoals in het onderstaande voorbeeld:

```
GET https://zab.test.lsp.aorta-zorg.nl/zab/organizations?$filter=type eq 'NICTIZ:J8' or type eq 'NICTIZ:K3' &$search=janssen
```

In dit geval wordt gezocht naar zorgaanbieders die ofwel het zorgaanbiedertype 'NICTIZ:J8' (Openbare apotheek) ofwel het zorgaanbiedertype 'NICTIZ:K3' (Apotheekhoudende huisartspraktijk) ofwel beide zorgaanbiedertypes hebben en voldoen aan de zoekexpressie 'janssen'.

'and' of andere operatoren worden in het geval van deze query niet ondersteund.

Het is ook mogelijk om i.p.v. 'NICTIZ' een ander codesysteem zoals 'VZVZ' te gebruiken:

```
GET https://zab.test.lsp.aorta-zorg.nl/zab/organizations?$filter=type eq 'VZVZ:Z001' or type eq 'VZVZ:Z002' &$search=janssen
```

Als de prefix, bijvoorbeeld 'NICTIZ:', wordt weggelaten wordt gezocht op zowel 'NICTIZ:<Code>' als 'VZVZ:<Codes>'.

GET /lspOrganizations:

Hiervoor werd een query op de resource /organizations beschreven maar het is ook mogelijk om een query uit te voeren op de resource /lspOrganizations. Hierbij wordt er gezocht binnen de zorgaanbieders waaraan tenminste één LSP-applicatie is gekoppeld. In dit geval kan de \$filter parameter ook worden gecombineerd met \$search. Hierbij geldt dezelfde beperking dat m.b.t. de \$filter parameter alleen het filteren op zorgaanbiedertype wordt ondersteund. Voorbeeld:

```
GET https://zab.test.lsp.aorta-zorg.nl/zab/lspOrganizations?$filter=type eq 'NICTIZ:Z3' &$search=janssen
```

In dit geval zullen alleen zorgaanbieders waaraan tenminste één LSP-applicatie is gekoppeld worden opgeleverd. Er wordt daarnaast het zoekcriterium toegepast dat is aangegeven in de \$filter parameter, d.w.z. dat het zorgaanbiedertype gelijk moet zijn aan 'NICTIZ:Z3' (Huisartspraktijk (zelfstandig of groepspraktijk)).

Het is mogelijk om meerdere zorgaanbiedertypes aan te geven d.m.v. een 'or' operator zoals in het eerder gegeven voorbeeld. De 'and' operator wordt in dit geval niet ondersteund.

\$filter is in het geval van deze query optioneel.

Er wordt niet gefilterd op de status van de LSP-applicatie(s), d.w.z. deze query levert zorgaanbieders op waaraan één of meerdere LSP-applicatie(s) zijn gekoppeld ongeacht of de status van die LSP-applicatie(s) Actief, Onderhoud of Afgesloten is.

Read-Only Interface:

De hierboven genoemde query's kunnen ook uitgevoerd worden via de read-only interface (zie par. 1.2.1), bijvoorbeeld:

```
GET https://zab.prod.aorta-zorg.nl/zab-ro/organizations?$filter=type eq 'NICTIZ:Z3' &$search=janssen
```

3.7 \$select

De parameter `$select` kan gebruikt worden om een gedeelte van de velden terug te laten geven. De waarde die meegegeven moet worden is een komma-gescheiden lijst van veldnamen. Deze kan gecombineerd worden met `$filter` of `$search`. Bijvoorbeeld:

```
https://zab.test.lsp.aorta-zorg.nl/zab/organizations?$filter=naam eq 'Apotheek Janssen' and plaats eq 'Utrecht' &$select=addresses,telecoms
```

In dit geval worden alleen de adresgegevens ('addresses') en contactgegevens ('telecoms') teruggegeven van de organisatie die voldoet aan het filter. De queryparameter `$select` kan ook gebruikt worden i.c.m. andere URI's (resources), bijvoorbeeld:

```
https://zab.test.lsp.aorta-zorg.nl/zab/organizations/1?$select=addresses,telecoms
```

Bij het uitvoeren van een GET-request naar deze URI worden de adresgegevens en contactgegevens van de organisatie met ID 1 teruggegeven.

3.8 Content Type

Objecten worden teruggegeven in JSON-formaat. Ook bij een POST- of PATCH request moet de request-body meegegeven worden in JSON-formaat (content-type 'application/json').

3.9 Validatie en Restricties

Op de volgende velden wordt validatie uitgevoerd:

- Het veld Gender (JSON: 'gender') van een zorgverlener (Individual). Geldige waardes zijn:
 - male
 - female
 - other
 - unknown

Bij het doorgeven van een ongeldige waarde wordt een HTTP 400 Bad Request status teruggegeven met de foutmelding: `{"error": "Invalid gender <waarde>"}`.

- Het veld Languages (JSON: 'languages') van een zorgverlener (Individual). Dit veld is een lijst (array) en moet BCP-47 taalcodes bevatten, bijvoorbeeld:
 - nl
 - nl-BE

Bij het doorgeven van een ongeldige waarde wordt een HTTP 400 Bad Request status teruggegeven met de foutmelding: `{"error": "Invalid language <waarde>"}`.

Op de overige velden wordt geen validatie uitgevoerd. Er wordt dus, bijvoorbeeld, niet gecontroleerd of een postcode overeenkomt met een straatnaam/huisnummer of dat een e-mailadres correct geformatteerd is.

3.10 Voorbeeld van het Uitvoeren van een Query

In deze paragraaf wordt een voorbeeld gegeven van twee eenvoudige query's.

Een groot aantal query's kan worden uitgevoerd via één van de volgende GET requests:

```
GET https://zab.test.lsp.aorta-zorg.nl/zab/organizations
```

```
GET https://zab.test.lsp.aorta-zorg.nl/zab/individuals
```

Het eerste request dient voor het zoeken van zorgaanbieders (Organizations) en het tweede request dient voor het zoeken van zorgverleners (Individuals).

Er moet een queryparameter worden meegegeven, '\$filter' of '\$search'. Wanneer een van de GET requests wordt uitgevoerd zonder een van deze queryparameters wordt de volgende foutmelding gegeven:

```
{
  "error": "No $filter or $search supplied"
}
```

Deze foutmelding wordt gegeven omdat zoals gezegd de queryparameter '\$filter' of de queryparameter '\$search' moet worden meegestuurd. Voor details over '\$filter' zie paragraaf 3.4. Voor details over '\$search' zie paragraaf 3.5.

('\$filter' of '\$search' is verplicht bij de hierboven genoemde GET requests. Er zijn ook andere GET requests mogelijk zoals `GET https://zab.test.lsp.aorta-zorg.nl/zab/organizations/[ID]`. Hierbij zijn '\$filter' en '\$search' niet van toepassing.)

Hieronder wordt een voorbeeld gegeven van het zoeken van een zorgaanbieder (Organization) via de queryparameter '\$filter':

```
GET https://zab.test.lsp.aorta-zorg.nl/zab/organizations?$filter=name eq 'Apotheek Janssen' and city eq 'Utrecht'
```

Er zal nu bij het versturen van het request, aangenomen dat de betreffende zorgaanbieder is opgenomen in ZORG-AB, een response worden gegeven zoals hieronder getoond:

```
[{
  "_self": "/organizations/1",
  "_className": "CSC.HPD.Organization",
  "_id": "1",
  "addresses": [{
    "active": true,
    "address": "Orteliuslaan 1004\r\n3528BD UTRECHT",
    "author": "Vektis",
    "city": "UTRECHT",
    "country": "NEDERLAND",
    "firstLine": null,
    "geolocation": null,
    "postalCode": "3528BD",
    "state": "ZH",
    "streetName": "Orteliuslaan",
    "streetNumber": "1004",
    "timestamp": "2020-11-03 12:49:40.265",
    "type": "Practice"
  },
  ...
],
  "applicationIds": [
    "12345678"
  ],
  "attachments": null,
  "author": "Vektis",
  "comment": null,
```



```

"credentials": null,
"displayName": "Apotheek Janssen",
"electronicServices": [{
  "_className": "CSC.HPD.LSP.Application",
  "active": true,
  "address": "janssen.voorbeeld.nl",
  "applicationId": "12345678",
  "author": "LSP",
  "conformances": [{
    "interactionId": "COMT_IN113113NL",
    "receive": true,
    "send": true
  }],
  ...
},
"description": null,
"status": "active",
"systemRoles": [
  "AllPurpose"
],
"timestamp": "2020-08-24 13:48:56.726"
},
],
"identifications": [{
  "active": true,
  "author": "UZI",
  "description": null,
  "timestamp": "2020-11-03 12:49:40.265",
  "type": "URA",
  "value": "12345678"
},
...
],
"names": [{
  "active": true,
  "author": "Vektis",
  "fullName": "Apotheek Janssen",
  "preferred": false,
  "timestamp": "2020-11-03 12:49:40.265",
  "type": "Display"
},
...
],
"speciality": null,
"telecoms": [{
  "active": true,
  "author": "Vektis",
  "name": "Algemeen",
  "preferred": false,
  "timestamp": "2020-08-24 13:39:19.313",
  "type": "Phone",
  "value": "012-3456789"
},
...
],
"timestamp": "2020-11-03 12:49:40.265",
"type": null,

```

```

    "types": [{
      "active": true,
      "author": "Vektis",
      "code": "02",
      "displayName": "Apothekers",
      "timestamp": "2020-11-03 12:49:40.265",
      "type": "Vektis Zorgsoort"
    }, {
      "active": true,
      "author": "VZVZ",
      "code": "Z001",
      "displayName": "Apotheek",
      "timestamp": "2020-11-03 12:49:40.265",
      "type": "VZVZ"
    }, {
      "active": true,
      "author": "VZVZ",
      "code": "J8",
      "displayName": "Openbare apotheek",
      "timestamp": "2020-11-03 12:49:40.265",
      "type": "NICTIZ"
    }
  ],
  "ura": "12345678"
}
]

```

Hieronder wordt een voorbeeld gegeven van het zoeken van een zorgaanbieder (Organization) via de queryparameter '\$search':

```
GET https://zab.test.lsp.aorta-zorg.nl/zab/organizations?$search=apotheek
janssen utrecht
```

Bij het versturen van dit request zal, aangenomen dat de betreffende zorgaanbieder is opgenomen in ZORG-AB, dezelfde response worden gegeven als hierboven of een vergelijkbare response.

(Er is een aantal verschillen tussen het zoeken via '\$filter' en '\$search' waardoor er ook een verschil kan zijn qua matches die worden opgeleverd. Bijvoorbeeld wordt bij '\$filter' alleen gezocht binnen de in de query aangegeven velden, in dit geval naam en plaats. Bij '\$search' wordt altijd gezocht binnen zorgaanbiedertype, naam en plaats ongeacht welke trefwoorden zijn meegegeven (en ongeacht de volgorde waarin deze staan). Verder wordt bij '\$filter' Soundex toegepast waardoor er niet alleen een match is met 'janssen' maar ook met 'jansen'. Bij '\$search' wordt er fuzzy search toegepast waardoor er ook een match is met bijvoorbeeld 'hansen'. Zie ook par. 3.5.)

3.11 Voorbeeld van het Uitvoeren van de Zorgaanbiedertype-query ('JGZ-query')

De zorgaanbiedertype-query moet uitgevoerd worden op de resource `/organizations/_summary`, waarbij de querystring eruit moet zien zoals hieronder getoond:

```
GET https://zab.test.lsp.aorta-
zorg.nl/zab/organizations/_summary?$filter=type eq '[Type]: [Code]'

GET https://zab.test.lsp.aorta-
zorg.nl/zab/organizations/_summary?$filter=type eq '[Code]'
```

Hierin is *[Code]* de zorgaanbiedertype-code, bijvoorbeeld 'JGZ', en *[Type]* de code van de corresponderende waardelijst/codesysteem, bijvoorbeeld 'NICTIZ'. Het is mogelijk om alleen een code te gebruiken, zoals in de tweede regel hierboven, in dat geval wordt er gezocht op alle 'zorgaanbiedertype-types' die bekend (geconfigureerd) zijn.

Voorbeelden:

```
GET https://zab.test.lsp.aorta-
zorg.nl/zab/organizations/_summary?$filter=type eq 'NICTIZ:JGZ'

GET https://zab.test.lsp.aorta-
zorg.nl/zab/organizations/_summary?$filter=type eq 'JGZ'
```

In dit voorbeeld wordt er uitgegaan van de volgende waardelijst:

Naam	RoleCodeNL - zorgaanbiedertype (organisaties)
OID	2.16.840.1.113883.2.4.15.1060
URI	http://nictiz.nl/fhir/NamingSystem/organization-type
URL	https://decor.nictiz.nl/decor/services/RetrieveValueSet?id=2.16.840.1.113883.2.4.3.11.60.1.11.1&effectiveDate=2020-05-18T12:30:00&prefix=nictiz2bbr-&format=html&collapsable=true&language=nl-NL&ui=nl-NL
ZORG-AB Prefix	NICTIZ

Via het eerste voorbeeld hierboven wordt in ZORG-AB gezocht op JGZ-instellingen (JeugdGezondheidsZorg-instellingen) of om precies te zijn op Organizations (zorgaanbieders) met in het Types attribuut een zorgaanbiedertype met een Type gelijk aan 'NICTIZ' en een Code gelijk aan 'JGZ'.

Een ander voorbeeld uitgaande van dit codesysteem:

```
GET https://zab.test.lsp.aorta-
zorg.nl/zab/organizations/_summary?$filter=type eq 'NICTIZ:J8'
```

Hiermee wordt in ZORG-AB gezocht op (openbare) apotheken of om precies te zijn op Organizations (zorgaanbieders) met in het Types attribuut een zorgaanbiedertype met een Type gelijk aan 'NICTIZ' en een Code gelijk aan 'J8'.

Na het versturen van het eerste voorbeeld-request, dat met parameter 'NICTIZ:JGZ', zal er een response worden gegeven zoals hieronder getoond:

```
[{
  "_self": "/organizations/1/_summary",
  "_id": "1",
  "applicationIds": null,
  "displayName": "JGZ Voorbeeld 1",
  "type": ["JGZ"],
  "types": [{
    "active": true,
    "code": "JGZ",
    "displayName": null,
    "timestamp": "2019-10-08 16:59:22.618",
    "type": "NICTIZ"
  }
  ],
  "ura": "00000001"
```

```

    }, {
      "_self": "/organizations/2/_summary",
      "_id": "2",
      "applicationIds": null,
      "displayName": "JGZ Voorbeeld 2",
      "type": ["JGZ"],
      "types": [{
        "active": true,
        "code": "JGZ",
        "displayName": null,
        "timestamp": "2019-10-08 16:59:23.403",
        "type": "NICTIZ"
      }]
    },
    ],
    "ura": "00000002"
  }
]

```

Als volgt kan een query worden uitgevoerd waarbij gezocht wordt op dit zorgaanbiedertype ongeacht het 'zorgaanbiedertype-type':

```

GET https://zab.test.lsp.aorta-
zorg.nl/zab/organizations/_summary?$filter=type eq 'JGZ'

```

Er wordt een summary object (subset van de beschikbare velden) teruggegeven. De volgende velden zijn onderdeel van het summary object en worden daarom teruggegeven:

Naam (JSON-naam)	Toelichting
applicationIds	Lijst van LSP applicatie ID's
displayName	Gevelnaam (om precies te zijn is dit een vluchtig veld dat wordt gevuld met de naam in het Names attribuut met een Type gelijk aan 'Display')
type <i>Deprecated in ZORG-AB 2.6.2</i>	Lijst (array) van 1 of meer zorgaanbiedertypes (strings)
types <i>Toegevoegd in ZORG-AB 2.6.2</i>	Zorgaanbiedertype(s). Dit is een lijst die 1 of meer codes bevat conform mogelijk verschillende waardelijsten/codesystemen. Zie voor een beschrijving van de attributen de tabel in par. 2.1
ura	URA-nummer

3.11.1 Niet-Ondersteunde Vektis Zorgsoortcodes

Het is mogelijk om te zoeken op Vektis Zorgsoortcode, voorbeeld:

```
GET https://zab.test.lsp.aorta-
zorg.nl/zab/organizations/_summary?$filter=type eq 'Vektis Zorgsoort:02'
```

Het zoeken op de volgende Vektis Zorgsoortcodes wordt in de huidige versie van ZORG-AB echter niet ondersteund:

Code	Weergavenaam
01	Huisartsen
04	Fysiotherapeuten
05	Logopedisten
12	Tandartsen
41	ZZP-ers in wijkverpleging/ PGB aanbieders / Beheerstichtingen
76	Leveranciers Hulpmiddelen
89	Schoonheidspecialisten
90	Overige therapeuten en Complementair en Aanvullende zorg
94	Psychologische Zorgverleners
96	Pedicuren
98	Declaranten/Servicebureaus/Zorgverzekeraars

Er wordt in dit geval de volgende foutmelding opgeleverd:

```
{
  "error": "Searching for organization type Vektis Zorgsoort
01,04,05,12,41,76,89,90,94,96,98 is currently not supported"
}
```

3.12 Voorbeeld van het Gebruik van de 'StartsWith' Functie in een Query

Er kan gebruik worden gemaakt van de functie 'startswith' om aan te geven dat er ook gezocht moet worden naar waarden die 'beginnen met' de zoekterm i.p.v. dat deze volledig overeen moet komen. Voorbeeld:

```
https://zab.test.lsp.aorta-
zorg.nl/zab/organizations?$filter=startswith(naam, 'Jan') and
startswith(plaats, 'Utrech')
```

Met deze query wordt gezocht naar zorgaanbieders waarvan de naam begint met (of gelijk is aan) 'Jan' en waaraan een adres is gekoppeld in een plaats die begint met (of gelijk is aan) 'Utrech'. Dit levert een match op met bijvoorbeeld:

- Een zorgaanbieder met naam 'Huisartsenpraktijk Jan de Boer' en plaats Utrecht
- Een zorgaanbieder met naam 'Apotheek Janssen B.V.' en plaats Utrecht

NB De woorden 'Huisartsenpraktijk', 'Apotheek' en 'B.V.' zijn woorden die op een lijst staan van voor- en achtervoegsels die door ZORG-AB worden verwijderd bij het zoeken (en indexeren). Om deze reden is er toch sprake van een match met de bovengenoemde zorgaanbieders.

De 'startswith' functie wordt ondersteund bij de volgende zoekvelden: naam / name, plaats / city, postcode / postalcode en organisatie / organization.

Bij het uitvoeren van het 'startswith' zoekalgoritme wordt er géén Soundex toegepast. Wel is het zo dat in aanvulling op het 'startswith' zoekalgoritme het standaard zoekalgoritme wordt uitgevoerd, waarbij wél Soundex wordt toegepast. Er worden dus twee zoekalgoritmes uitgevoerd en de resultaten van beide worden gecombineerd. De gedachtegang daarachter is dat als er gezocht wordt op bijvoorbeeld 'Rijn' er een zorgaanbieder of -verlener met de naam 'Rijnland' kan worden gevonden, maar ook een zorgaanbieder of -verlener met de naam 'Rein'. Zo kan er toch (mogelijk) geprofiteerd worden van de Soundex-functionaliteit.

De 'startswith' functie is beschreven in de OData URL-conventie, zie: <http://docs.oasis-open.org/odata/odata/v4.0/odata-v4.0-part2-url-conventions.html>

3.13 Voorbeeld van het Toevoegen van een Organisatie (POST-Request)

In deze paragraaf wordt een voorbeeld gegeven van het toevoegen van een organisatie aan ZORG-AB. Hieronder wordt een voorbeeld-URI getoond:

```
https://zab.test.lsp.aorta-zorg.nl/zab/organizations
```

En hieronder wordt een voorbeeld organisatie-object getoond in JSON-formaat:

```
{
  "names": [
    {
      "type": "Legal",
      "fullName": "Apotheek Janssen B.V."
    }
  ],
  "addresses": [
    {
      "type": "Practice",
      "streetName": "Orteliuslaan",
      "streetNumber": "1004",
      "postalCode": "3528 BD",
      "city": "Utrecht"
    }
  ]
}
```

Wanneer er een HTTP POST-request wordt uitgevoerd naar de bovenstaande URI met het bovenstaande JSON-object in de request-body dan zal de organisatie toegevoegd worden aan ZORG-AB. Als de organisatie succesvol is toegevoegd zal een HTTP 201 (Created) status code worden teruggegeven, alsmede de volledige JSON-body van de nieuw aangemaakte organisatie.

3.14 Voorbeeld van het Toevoegen van een Suborganisatie (POST-Request)

Een suborganisatie kan op dezelfde wijze toegevoegd worden als een reguliere organisatie (zie paragraaf 3.13), uitgezonderd dat er een 'subOrganization'-relatie aangemaakt moet worden waarmee wordt aangegeven aan welke hoofdorganisatie de suborganisatie is gerelateerd. Dit moet via een POST-verzoek naar de volgende URI:

```
https://zab.test.lsp.aorta-zorg.nl/zab/subOrganization
```

Met de volgende berichtinhoud:

```
{
  "mainOrganizationId" : <ID van de hoofdorganisatie>,
  "subOrganizationId" : <ID van de suborganisatie>
}
```

Hierin moeten de cursief gedrukte gedeeltes vervangen worden door de ID's van respectievelijk de hoofd- en suborganisatie.

3.15 Voorbeeld van het Toevoegen van een Individu (POST-Request)

In deze paragraaf wordt een voorbeeld gegeven van het toevoegen van een individu (een specialist) aan ZORG-AB. Hieronder wordt een voorbeeld-URI getoond:

```
https://zab.test.lsp.aorta-zorg.nl/zab/individuals
```

En hieronder wordt een voorbeeld individu-object getoond in JSON-formaat:

```
{
  "names": [
    {
      "type": "Legal",
      "fullName": "Jan van het Zand"
    },
    {
      "type": "Display",
      "firstName": "Jan",
      "initials": "J.",
      "prefix": "van het",
      "familyName": "Zand",
      "suffix": "MD"
    }
  ],
  "gender": "male",
  "languages": [
    "nl"
  ],
  "speciality": "17.000"
}
```

Wanneer er een HTTP POST-request wordt uitgevoerd naar deze URI met dit object in de request-body dan zal het individu worden toegevoegd en als dit succesvol is zal er een response worden teruggegeven met een HTTP 201 (Created) status code en het nieuw aangemaakte object in JSON-formaat in de request-body.

Het individu kan vervolgens gekoppeld worden aan een organisatie door een HTTP POST-request uit te voeren naar de URI:

```
https://zab.test.lsp.aorta-zorg.nl/zab/worksAt
```

Met als request-body een JSON-object met zowel het technische ID van de organisatie als het technische ID van het individu:

```
{
  "organizationId" : 110496,
  "individualId" : 110500
}
```

Hierin zijn '110496' en '110500' voorbeelden, deze moeten vervangen worden door de daadwerkelijke technische ID's. De ID's worden na het toevoegen van een resource (HTTP POST) teruggegeven in het veld '_id' in de response-body. Ook bij het uitvoeren van een query (HTTP GET) wordt het ID teruggegeven in het veld '_id'.

3.16 Voorbeeld van het Toevoegen van Contactgegevens die Alleen Zichtbaar Mogen Zijn voor Collega-Zorgaanbieders

Contactgegevens zoals telefoon- en faxnummers en e-mailadressen kunnen opgeslagen worden in het veld 'telecoms' bij een individu (specialist). Deze gegevens zijn zichtbaar voor alle organisaties die ZORG-AB kunnen raadplegen, inclusief organisaties waarvoor in het LSP-applicatieregister (APR) is geregistreerd dat deze van het type 'GBP' zijn (Goed Beheerd Patiëntenportaal).

Wanneer specifieke contactgegevens alleen gedeeld mogen worden met collega-zorgaanbieders en niet zichtbaar mogen zijn vanuit patiëntenportalen, dan kunnen de gegevens toegevoegd worden aan het veld 'telecoms' van de 'worksAt'-relatie van het individu. In dit geval worden de gegevens bij het opvragen van het individu alleen getoond aan niet-GBP's. Bij een opvraging door een GBP worden de gegevens niet getoond.

Als er nog geen 'worksAt'-relatie bestaat tussen het individu en de organisatie dan kan er een 'worksAt'-relatie aangemaakt worden door een POST-request te versturen naar de volgende URI:

```
https://zab.test.lsp.aorta-zorg.nl/zab/worksAt
```

Een voorbeeld van de request-body die dan meegegeven moet worden wordt hieronder gegeven:

```
{
  "organizationId" : 110496,
  "individualId" : 110500,
  "telecoms" : [
    {
      "type" : "Phone",
      "name": "Intercollegiaal",
      "value": "+31123456789"
    }
  ]
}
```

Hierin moeten de voorbeeld-ID's ('110496' en '110500') vervangen worden door de daadwerkelijke technische ID's van de organisatie en het individu. In het 'telecoms'-veld wordt hier één voorbeeldtelefoonnummer getoond, eventueel kunnen er meerdere contactgegevens (zoals een e-mailadres) worden meegegeven in de array.

Als de 'worksAt'-relatie al bestaat dan kan het 'telecoms'-veld toegevoegd worden door een PATCH-request uit te voeren naar de URI van de 'worksAt'-relatie (in onderstaande regel moet '98004' vervangen worden door het daadwerkelijke ID van de 'workAt'-relatie):

```
https://zab.test.lsp.aorta-zorg.nl/zab/worksAt/98004
```

In de request-body hoeft dan alleen het 'telecoms'-veld meegegeven te worden, als volgt:

```
{
  "telecoms" : [
    {
      "type" : "Phone",
      "name": "Intercollegiaal",

```



```
        "value": "+31123456789"  
      }  
    ]  
  }
```

3.17 Voorbeeld van het Toevoegen van Beschikbaarheidsgegevens van een Zorgverlener

Het veld 'availability' van de WorksAt-relatie (zie ook de figuur in par. 2.1) is bedoeld voor het opslaan van beschikbaarheidsgegevens van zorgverleners. Er wordt voorgesteld dat gebruikers / leveranciers hiervoor het iCalendar-formaat gebruiken. Dit wordt beschreven in de RFC 5545 standaard, link: <https://tools.ietf.org/html/rfc5545>

Er vindt echter geen validatie of parsing plaats (zoals er ook op een groot aantal andere velden, bijvoorbeeld postcode, niet wordt gevalideerd). M.a.w. er wordt niet technisch afgedwongen dat RFC 5545 wordt gebruikt. Het veld heeft datatype 'String'.

Om de beschikbaarheidsgegevens (in iCalendar-formaat) in het veld 'availability' te plaatsen moeten de volgende stappen worden gevolgd:

1. Bepaal het ID van de zorgverlener (Individual), bijvoorbeeld door deze op te zoeken via zijn/haar UZI-nummer: `GET [Base URL]/zab/uzi/[UZI Nummer]`
2. Bepaal het ID van de WorksAt-relatie via het volgende GET request: `GET [Base URL]/zab/individuals/[Individual ID]/relations`
3. Voer het volgende PATCH request uit: `PATCH [Base URL]/zab/worksAt/[WorksAt ID]` met in de body een JSON-object met daarin het veld 'availability' met als waarde de iCalendar-stream als JSON-string

Hiervan wordt in deze paragraaf een voorbeeld gegeven.

Er wordt uitgegaan van de volgende iCalendar-stream:

```
BEGIN:VCALENDAR  
VERSION:2.0  
PRODID:-//Huisartsenpraktijk J. Janssen//NONSGML v1.0//EN  
BEGIN:VEVENT  
UID:19970610T172345Z-AF23B2@example.com  
DTSTAMP:20180615T120000  
DTSTART:20180618T090000  
RRULE:FREQ=WEEKLY;BYDAY=MO;BYHOUR=8,9,10,11,12,13,14  
SUMMARY:Dhr. J. Janssen is aanwezig op maandag van 08:00 tot 15:00  
END:VEVENT  
BEGIN:VEVENT  
UID:19970610T172345Z-AF23B2@example.com  
DTSTAMP:20180615T120000  
DTSTART:20180618T090000  
RRULE:FREQ=WEEKLY;BYDAY=TU;BYHOUR=8,9,10  
SUMMARY:Dhr. J. Janssen is aanwezig op dinsdag van 08:00 tot 11:00  
END:VEVENT  
END:VCALENDAR
```

Om deze toe te voegen aan een zorgverlener (Individual) moet eerst het (technische) ID bepaald worden van de zorgverlener. Dit kan door, bijvoorbeeld, te zoeken op een UZI-nummer via het volgende GET-request:

```
GET https://zab.test.lsp.aorta-zorg.nl/zab/uzi/00001234
```

(Hierin moet 00001234 worden vervangen door het daadwerkelijke UZI-nummer.)

Het ID wordt vervolgens opgeleverd in het veld '_id'.

Vervolgens moet het ID van de WorksAt-relatie worden bepaald via het volgende GET-request:

```
GET https://zab.test.lsp.aorta-zorg.nl/zab/individuals/56789/relations
```

(Hierin moet 56789 worden vervangen door het daadwerkelijke ID van de Individual.)

Het ID wordt vervolgens opgeleverd in het veld '_id'. Het is mogelijk dat de zorgverlener meerdere WorksAt-relaties heeft. In dat geval moet het ID gebruikt worden van de WorksAt-relatie die correspondeert met de zorgaanbieder (Organization) waarop de beschikbaarheidsgegevens van toepassing zijn.

Vervolgens moet het volgende PATCH-request worden uitgevoerd:

```
PATCH https://zab.test.lsp.aorta-zorg.nl/zab/worksAt/101112
```

(Hierin moet 101112 worden vervangen door het daadwerkelijke ID van de WorksAt-relatie.)

Met in de body het volgende JSON-object:

```
{
  "availability": "BEGIN:VCALENDAR\r\nVERSION:2.0\r\nPRODID:-//Huisartsenpraktijk J. Janssen//NONSGML v1.0//EN\r\nBEGIN:VEVENT\r\nUID:19970610T172345Z-AF23B2@example.com\r\nDTSTAMP:20180615T120000\r\nDTSTART:20180618T090000\r\nRRULE:FREQ=WEEKLY;BYDAY=MO;BYHOUR=8,9,10,11,12,13,14\r\nSUMMARY:Dhr. J. Janssen is aanwezig op maandag van 08:00 tot 15:00\r\nEND:VEVENT\r\nBEGIN:VEVENT\r\nUID:19970610T172345Z-AF23B2@example.com\r\nDTSTAMP:20180615T120000\r\nDTSTART:20180618T090000\r\nRRULE:FREQ=WEEKLY;BYDAY=TU;BYHOUR=8,9,10\r\nSUMMARY:Dhr. J. Janssen is aanwezig op dinsdag van 08:00 tot 11:00\r\nEND:VEVENT\r\nEND:VCALENDAR\r\n"
}
```

Er wordt vervolgens een response gegeven met een HTTP-statuscode 202 Accepted om aan te geven dat het request succesvol is verwerkt.

De beschikbaarheidsgegevens kunnen vervolgens worden opgehaald via een GET-request:

```
GET https://zab.test.lsp.aorta-zorg.nl/zab/worksAt/101112
```

(Hierin moet 101112 worden vervangen door het daadwerkelijke ID van de WorksAt-relatie.)

Dit levert een response op zoals in het onderstaande voorbeeld:

```
{
  "_self": "/worksAt/101112",
  "_className": "CSC.HPD.Relation.WorksAt",
  "_id": "101112",
  "availability": "BEGIN:VCALENDAR\r\nVERSION:2.0\r\nPRODID:-//Huisartsenpraktijk J. Janssen//NONSGML v1.0//EN\r\nBEGIN:VEVENT\r\nUID:19970610T172345Z-
```

```
AF23B2@example.com\r\nDTSTAMP:20180615T120000\r\nDTSTART:20180618T090000\r\nRRULE:FREQ=WEEKLY;BYDAY=MO;BYHOUR=8,9,10,11,12,13,14\r\nSUMMARY:Dhr. J. Janssen is aanwezig op maandag van 08:00 tot 15:00\r\nEND:VEVENT\r\nBEGIN:VEVENT\r\nUID:19970610T172345Z-AF23B2@example.com\r\nDTSTAMP:20180615T120000\r\nDTSTART:20180618T090000\r\nRRULE:FREQ=WEEKLY;BYDAY=TU;BYHOUR=8,9,10\r\nSUMMARY:Dhr. J. Janssen is aanwezig op dinsdag van 08:00 tot 11:00\r\nEND:VEVENT\r\nEND:VCALENDAR\r\n",
    "individualId": "56789",
    "organizationId": "131415",
    "speciality": null,
    "telecoms": null,
    "timestamp": "2018-07-30 09:14:44.364"
  }
}
```

Een alternatieve manier om het ID van de WorksAt-relatie te bepalen is via de zorgaanbieder (Organization) via de volgende stappen:

1. Bepaal het ID van de zorgaanbieder (Organization), bijvoorbeeld door deze op te zoeken via het URANummer: `GET [Base URL]/zab/ura/[URA Nummer]`
2. Bepaal het ID van de WorksAt-relatie via het volgende GET request: `GET [Base URL]/zab/organizations/[Organization ID]/relations`

Het ID wordt vervolgens opgeleverd in het veld '_id'. Het is mogelijk dat de zorgaanbieder meerdere WorksAt-en/of SubOrganization-relaties heeft. In dat geval moet het ID gebruikt worden van de WorksAt-relatie die correspondeert met de zorgverlener (Individual) waarop de beschikbaarheidsgegevens van toepassing zijn.

3.18 Voorbeeld van het Toevoegen van EDIFACT Emailadres

EDIFACT emailadressen kunnen worden toegevoegd door middel van het volgende POST request:

```
POST https://zab.test.lsp.aorta-zorg.nl/zab/organizations/[ID]/services
```

Hierin moet **[ID]** worden vervangen door het ID van de zorgaanbieder (Organization). De volgende request body moet worden gebruikt:

```
{
  "type": "EDIFACT",
  "active": true,
  "value": null,
  "address": "[Emailadres]",
  "description": null
}
```

Hierin moet **[Emailadres]** worden vervangen door het emailadres van de zorgaanbieder (Organization).

Het emailadres kan vervolgens worden gewijzigd of verwijderd door middel van resp. een PATCH request of DELETE request:

```
PATCH https://zab.test.lsp.aorta-zorg.nl/zab/organizations/[ID]/services/[Index]
```

```
DELETE https://zab.test.lsp.aorta-zorg.nl/zab/organizations/[ID]/services/[Index]
```

Hierin moet **[Index]** worden vervangen door de positie van het emailadres dat moet worden gewijzigd/verwijderd. NB Er kunnen meerdere Electronic Services in de lijst staan, ook bijvoorbeeld LSP-applicaties of MedMij Gegevensdiensten. De positie is op 1 gebaseerd (het eerste item in de lijst heeft positie 1).

3.19 MedMij Gegevensdiensten

De volgende gegevens worden geïmporteerd in ZORG-AB vanuit de MedMij ZKL (ZorgaanbiedersKoppelLijst, zie par. 5.5):

- Zorgaanbiedernaam, bijvoorbeeld: umcharderwijk@medmij
- Gegevensdiensten

De zorgaanbiedernaam is in ZORG-AB een instantie van de klasse 'Identification' en wordt opgeleverd bij zorgaanbieders (instanties van de klasse Organization) in het attribuut 'Identifications'. Zie **Figuur 1**. (Het attribuut 'Identifications' wordt geërfd van de superklasse Party.)

De gegevensdiensten zijn in ZORG-AB instanties van de klasse 'MedMij Gegevensdienst'. Deze wordt ook weergegeven in **Figuur 1**.

Vanuit de ZKL worden de volgende attributen gevuld: GegevensdienstId, Weergavenaam, Interfaceversies

De volgende attributen worden niet gevuld: AuthorizationEndpoint, TokenEndpoint, Abonneren, Systeemrollen

3.19.1 Voorbeeld Response Native REST Interface

Een voorbeeld van een JSON response wordt hieronder gegeven:

De attributen GegevensdienstId, Interfaceversies en Weergavenaam zijn gearceerd. (In de JSON response beginnen de namen van de attributen met een kleine letter.)

Ook het attribuut Identifications, waarin de zorgaanbiedernaam wordt opgeleverd, is gearceerd.

```
{
  "_self": "/organizations/1",
  "_className": "CSC.HPD.Organization",
  "_id": "1",
  ...
  "electronicServices": [{
    "_className": "CSC.HPD.MedMij.Gegevensdienst",
    "abonneren": {
      "maximaleDuur": null,
      "subscriptionEndpointuri": null
    },
    "author": "MedMij",
    "authorizationEndpoint": {
      "authorizationEndpointuri": null
    },
    "gegevensdienstId": "35",
    "interfaceversies": [{
      "interfaceversieId": "1.3.0"
    }
  ],
  "systeemrollen": null,
  "timestamp": "2021-06-23 13:12:51.897",
  "tokenEndpoint": {
    "tokenEndpointuri": null
  },
  "weergavenaam": "Medicatiegegevens"
},
  "identifications": [{
    "active": true,
    "author": "MedMij",
    "description": null,
    "timestamp": "2021-06-23 13:12:51.897",
    "type": "MedMij",
  }
]
```

```

    "value": "umcharderwijk@medmij"
  }
],
...
}

```

3.19.2 Zoeken o.b.v. MedMij Naam via Native REST Interface (OData Interface)

Om via de native REST interface te zoeken op MedMij zorgaanbiedernaam, bijvoorbeeld 'umcharderwijk@medmij', moet het volgende request worden uitgevoerd:

```
GET https://zab.test.lsp.aorta-zorg.nl/zab/identifications/MedMij: [MedMij Naam]
```

Hierin moet **[MedMij Naam]** worden vervangen door de MedMij zorgaanbiedernaam, bijvoorbeeld:

```
GET https://zab.test.lsp.aorta-zorg.nl/zab/identifications/MedMij:umcharderwijk@medmij
```

Het antwoord dat wordt teruggegeven is een JSON object dat eruitziet zoals het hierboven gegeven JSON object (aangenomen dat de zorgaanbieder is gevonden).

In het geval de zorgaanbieder niet kon worden gevonden wordt HTTP statuscode 404 Not Found teruggegeven en een lege response body.

3.19.3 Voorbeeld Response FHIR Interface

In het geval van een query via de FHIR interface wordt een FHIR 'Organization' resource opgeleverd.

De MedMij zorgaanbiedernaam en gegevensdiensten worden opgeleverd in attributen van deze resource.

Om precies te zijn wordt de zorgaanbiedernaam opgeleverd in het attribuut 'identifier'.

De gegevensdiensten worden gemapt naar FHIR 'Endpoint' resources. Deze wordt opgeleverd in de attributen 'endpoint' / 'contained'. (Het attribuut 'endpoint' bevat verwijzingen naar Endpoint resources in het attribuut 'contained'.)

De exacte mapping die wordt gebruikt is hieronder aangegeven:

FHIR Endpoint Attribuut	MedMij Gegevensdienst Attribuut	Opmerking
extension valueString	Author	Voor vanuit ZKL geïmporteerde gegevens vaste waarde: MedMij
extension url	Vaste waarde: http://www.vzviz.nl/fhir/StructureDefinition/author	
status	Vaste waarde: active	
connectionType system	Vaste waarde: http://hl7.org/fhir/endpoint-connection-type	
connectionType code	Vaste waarde: hl7-fhir-rest	
name	Weergavenaam	
payloadType coding system	Vaste waarde: http://www.medmij.nl/code/gegevensdienstid	
payloadType coding version	Interfaceversie InterfaceversieId	In het geval van meerdere interfaceversies worden meerdere payloadTypes opgeleverd
payloadType coding code	GegevensdienstId	

address	Vaste waarde: https://not-available	
---------	-------------------------------------	--

NB

- De waarden in het 'endpoint' attribuut zijn géén logische ID's van de Endpoint resources, de velden worden gevuld met een opeend nummer dat begint bij 1.
- Er kunnen ook andere typen elektronische diensten worden opgeleverd in de 'endpoint' / 'contained' attributen, bijvoorbeeld LSP-applicaties.

Een voorbeeld van een JSON response wordt hieronder gegeven:

De attributen waarin de weergavenaam, gegevensdienst ID, interfaceversie(s) en zorgaanbiedernaam worden opgeleverd zijn gearceerd.

```
{
  "resourceType": "Organization",
  "contained": [{
    "resourceType": "Endpoint",
    "address": "https://not-available",
    "connectionType": {
      "code": "hl7-fhir-rest",
      "system": "http://hl7.org/fhir/endpoint-connection-type"
    },
    "extension": [{
      "url": "http://www.vzvv.nl/fhir/StructureDefinition/author",
      "valueString": "MedMij"
    }
  ],
  "id": "1",
  "meta": {
    "lastUpdated": "2021-06-23T13:12:51.897"
  },
  "name": "Medicatiegegevens",
  "payloadType": [{
    "coding": [{
      "code": "35",
      "system": "http://www.medmij.nl/code/gegevensdienstid",
      "version": "1.3.0"
    }
  ]
  },
  "status": "active"
  ],
  "endpoint": [{
    "reference": "#1"
  }
  ],
  ...
  "id": "1",
  "identifier": [{
    "extension": [{
      "url": "http://www.vzvv.nl/fhir/StructureDefinition/author",
      "valueString": "MedMij"
    }
  ],
  "system": "http://www.medmij.nl/id/medmijnaam",
  "value": "umcharderwijk@medmij"
  }
  ],
  ...
}
```

3.19.4 Zoeken o.b.v. MedMij Naam via FHIR Interface

Om via de FHIR interface te zoeken op een MedMij zorgaanbiedernaam, bijvoorbeeld 'umcharderwijk@medmij', moet het volgende request worden uitgevoerd:

```
GET https://zab.test.lsp.aorta-zorg.nl/zab/fhir/Organization?identificer=http://www.medmij.nl/id/medmijnaam|[MedMij Naam]
```

Hierin moet **[MedMij Naam]** worden vervangen door de MedMij zorgaanbiedernaam, bijvoorbeeld:

```
GET https://zab.test.lsp.aorta-zorg.nl/zab/fhir/Organization?identificer=http://www.medmij.nl/id/medmijnaam|umcharderwijk@medmij
```

Het URI en de vertical bar ('|') mogen ook weg worden gelaten, bijvoorbeeld:

```
GET https://zab.test.lsp.aorta-zorg.nl/zab/fhir/Organization?identificer=umcharderwijk@medmij
```

(NB Hiermee wordt er technisch gezien ook op andere soorten identificers gezocht, zoals URA nummer, maar in de praktijk zullen er nooit matches met andere identificers plaatsvinden, omdat deze niet de toevoeging '@medmij' zullen bevatten.)

Het antwoord dat wordt teruggegeven is een FHIR 'Bundle' resource in JSON of XML formaat (afhankelijk van de HTTP header 'Accept' of de queryparameter '_format').

De zorgaanbieder (FHIR 'Organization' resource) wordt opgeleverd in het 'entry' attribuut.

De zorgaanbieder ziet er (in het geval het JSON formaat) is uit zoals in het hierboven gegeven voorbeeld.

In het geval de zorgaanbieder niet kon worden gevonden wordt een Bundle teruggegeven zonder 'entry' attribuut (en HTTP statuscode 200 OK).

4 Gebruik van de FHIR Interface

NB Voor details m.b.t. het verbinden met ZORG-AB zie par. 1.2.

4.1 Algemeen

De implementatie is gebaseerd op:

- De Argonaut Provider Directory Implementation Guide versie 1.0.0, link: <http://fhir.org/guides/argonaut/pd/release1>
- HL7 FHIR versie 3.0.1, link: <http://hl7.org/fhir/STU3/summary.html>

4.2 Resources / Interacties

De volgende FHIR Resources worden ondersteund:

- Organization
- Practitioner
- PractitionerRole
- Endpoint

De volgende interacties worden ondersteund voor Organization, Practitioner en PractitionerRole:

- read (GET)
- update (PUT)
- patch (PATCH)
- delete (DELETE)
- create (POST)
- search (GET)

De volgende interacties worden ondersteund voor Endpoint:

- search (GET)

Voor een volledige lijst van interacties / HTTP-methodes / URI's zie Bijlage 2.

4.3 Content Type

De volgende content types worden ondersteund:

- application/fhir+json
- application/fhir+xml

Bij GET-requests kan het gewenste content type aangegeven worden in de HTTP Accept header of in de queryparameter '_format'.

Andere content types (zoals application/xml of text/xml) zijn ook toegestaan, in dit geval zal in de Content-Type header van de response hetzelfde content-type worden aangehouden.

Bij PUT-, PATCH- en POST requests moet het content type van de in het request meegegeven data worden aangegeven in de HTTP Content-Type header. De response wordt in dit geval standaard gegeven in het content type van het request.

4.4 Validatie en Restricties

Bij de native REST interface vindt er geen validatie plaats, behalve voor 'gender' en 'languages'. Voor de FHIR interface geldt (vooralsnog) hetzelfde. Met 'gender' en 'languages' (in FHIR heet het veld waarin de taal wordt aangegeven 'communication') wordt als volgt omgegaan:

- Veld 'gender': geldige waarden zijn: 'male', 'female', 'other', 'unknown'. Wanneer een andere waarde wordt gebruikt wordt deze beschouwd als 'unknown'.
- Veld 'communication':
 - De volgende URI **mag** meegegeven worden in het veld 'system': <http://hl7.org/fhir/ValueSet/languages>
 - Er moet in dit geval een BCP-47 code worden gebruikt, bijvoorbeeld 'nl' of 'nl-BE'
 - Andere codes zijn ook toegestaan (Argonaut: "other codes may be used where these codes are not suitable", link: <http://www.fhir.org/guides/argonaut/pd/release1/StructureDefinition-argo-practitioner-definitions.html>)

4.5 Voorgedefinieerde Query's

Om te zoeken via de voorgedefinieerde query's moet een GET-request worden uitgevoerd met één of meerdere zoekparameters op één van de Resources. Dit wordt in de FHIR specificatie de 'search' interaction genoemd. De volgende voorgedefinieerde query's worden ondersteund:

Resource	Parameter 1	Parameter 2	Toelichting
Organization	identificer		Zoeken op URA, MedMij naam of AGB code. Zie p. 4.7
	name	address-city	Zoeken op naam + plaats
	type		Zoeken op zorgaanbiedertype Om deze query uit te voeren moet <code>_summary=true</code> worden meegegeven
Practitioner	identificer		Zoeken op UZI of AGB code. Zie p. 4.7
	family	address-city	Zoeken op naam + plaats
	gender		Filteren op geslacht, kan gecombineerd worden
	communication		Filteren op taal, kan gecombineerd worden

PractitionerRole	practitioner.identifier		Zoeken op UZI of AGB code. Zie p. 4.7
	practitioner.family	practitioner.address-city	Zoeken op naam + plaats
	practitioner.family	specialty	Zoeken op naam + specialisme
	specialty	practitioner.address-city	Zoeken op specialisme + plaats
	specialty	practitioner.address-postalcode	Zoeken op specialisme + postcode
	organization.identifier		Zoeken op URA
	practitioner.family	organization.name	Zoeken op naam + organisatienaam
	specialty	organization.name	Zoeken op specialisme + organisatienaam
	practitioner.gender		Filteren op geslacht, kan gecombineerd worden
	practitioner.communication		Filteren op taal, kan gecombineerd worden
Endpoint	identificer		Zoeken op LSP Applicatie ID
	organization		Zoeken op organisatie ID (ZORG-AB technisch ID)
Location			Wordt (nog) niet ondersteund

4.6 Voorbeeld van een Query

```
GET https://zab.test.lsp.aorta-zorg.nl/zab/fhir/Organization
?name=Apotheek Janssen&address-city=Utrecht
```

Response:

```
{
  "resourceType": "Bundle",
  "entry": [
    [Resultaten]
  ],
  "link": [
    {
      "relation": "self",
      "url": "https://zab.test.lsp.aorta-zorg.nl/zab/fhir/Organization
?name=Apotheek Janssen&address-city=Utrecht"
    }
  ]
}
```

```

    }],
    "total": [Aantal Resultaten],
    "type": "searchset"
  }

```

4.7 Voorbeeld van Zoeken op URA- of UZI- of KVK-nummer of AGB-code

Zoeken op identificerende gegevens, zoals URA-, UZI-, BIG-, of KVK-nummer, AGB-code of MedMij zorgaanbiedernaam kan via de FHIR-interface als volgt:

```
GET https://zab.test.lsp.aorta-zorg.nl/zab/fhir/Organization?identificer=[System URI] | [Value]
```

Of:

```
GET https://zab.test.lsp.aorta-zorg.nl/zab/fhir/Practitioner?identificer=[System URI] | [Value]
```

Of:

```
GET https://zab.test.lsp.aorta-zorg.nl/zab/fhir/PractitionerRole?organization.identificer=[System URI] | [Value]
```

Of:

```
GET https://zab.test.lsp.aorta-zorg.nl/zab/fhir/PractitionerRole?practitioner.identificer=[System URI] | [Value]
```

Zie hieronder voor voorbeelden.

Hierin moet een waarde (**[Value]**) worden ingevuld. Deze wordt gescheiden van de system URI (**[System URI]**) met een vertical bar ('|'). De system URI is optioneel. Wanneer de system URI wordt weggelaten moet ook de vertical bar worden weggelaten.

De system URI kan gevonden worden in de volgende tabel:

Resource	Parameter	Native REST-interface Type	System URI	Opmerkingen
Organization	identificer	URA	http://fhir.nl/fhir/NamingSystem/ura	
		MedMij	http://www.medmij.nl/id/medmijnaam	
		AGB_Onderneming en AGB_Vestiging	http://fhir.nl/fhir/NamingSystem/agb-z	
		AGB_Zorgaanbieder (Dit zijn AGB-codes die in ZORG-AB worden geïmporteerd t.b.v. 'toestemmingen')	http://fhir.nl/fhir/NamingSystem/agb-z	consent=true moet toegevoegd worden aan querystring, zie ook voorbeeld hieronder
		KVK	http://www.vzvz.nl/fhir/NamingSystem/kvk	Toegevoegd in ZORG-AB 2.7.0
Practitioner	identificer	UZI	http://fhir.nl/fhir/NamingSystem/uzi-nr-pers	
		AGB_Zorgverlener	http://fhir.nl/fhir/NamingSystem/agb-z	

		BIG	http://fhir.nl/fhir/NamingSystem/big	
Practitioner-Role	organization.identificer	URA	http://fhir.nl/fhir/NamingSystem/ura	
		MedMij	http://www.medmij.nl/id/medmijnaam	
		AGB_Onderneming en AGB_Vestiging	http://fhir.nl/fhir/NamingSystem/agb-z	
		AGB_Zorgaanbieder (Dit zijn AGB-codes die in ZORG-AB worden geïmporteerd t.b.v. 'toestemmingen')	http://fhir.nl/fhir/NamingSystem/agb-z	consent=true moet toegevoegd worden aan querystring, zie ook voorbeeld hieronder
		KVK	http://www.vzvz.nl/fhir/NamingSystem/kvk	Toegevoegd in ZORG-AB 2.7.0
	practitioner.identificer	UZI	http://fhir.nl/fhir/NamingSystem/uzi-nr-pers	
		AGB_Zorgverlener	http://fhir.nl/fhir/NamingSystem/agb-z	
		BIG	http://fhir.nl/fhir/NamingSystem/big	

In het geval dat er geen system URI (en vertical bar) worden meegegeven wordt er gezocht op **alle** types identificerende gegevens die voor de betreffende Resource genoemd zijn in bovenstaande tabel. Zie ook voorbeeld hieronder.

Voorbeelden Zorgaanbieder

Zoek zorgaanbieder met URA-nummer:

```
GET https://zab.test.lsp.aorta-zorg.nl/zab/fhir/Organization?identificer=http://fhir.nl/fhir/NamingSystem/ura|12345678
```

(De system URI is in dit geval `http://fhir.nl/fhir/NamingSystem/ura` en wordt door de vertical bar | gescheiden van het URA-nummer 12345678.)

Zoek zorgaanbieder met MedMij zorgaanbiedernaam:

```
GET https://zab.test.lsp.aorta-zorg.nl/zab/fhir/Organization?identificer=http://www.medmij.nl/id/medmijnaam|apotheekjanssen@medmij
```

Zoek zorgaanbieder met AGB-code:

```
GET https://zab.test.lsp.aorta-zorg.nl/zab/fhir/Organization?identificer=http://fhir.nl/fhir/NamingSystem/agb-z|12345678
```

Zoek zorgaanbieder op ID zonder system URI (d.w.z. zonder het ID te 'scopen'), hiermee wordt er effectief gezocht naar zorgaanbieders met een URA-nummer **of** een AGB-code gelijk aan 12345678:

```
GET https://zab.test.lsp.aorta-zorg.nl/zab/fhir/Organization?identificer=12345678
```

(Er wordt gezocht op het type URA, AGB_Onderneming, AGB_Vestiging en AGB_Zorgaanbieder. Er wordt in principe ook gezocht op het type MedMij maar omdat dit geen numerieke ID is zal dit geen match opleveren.)

Voorbeeld Zorgaanbieder (Toestemmingen)

Zoek zorgaanbieder met AGB-code van het type AGB_Zorgaanbieder:

```
GET https://zab.test.lsp.aorta-zorg.nl/zab/fhir/Organization?identificer=http://fhir.nl/fhir/NamingSystem/agb-z|12345678&consent=true
```

NB **consent=true** moet worden toegevoegd aan de querystring. Hiermee wordt aangegeven dat moet worden gezocht op AGB-codes van het type AGB_Zorgaanbieder. Dit zijn AGB-codes die in ZORG-AB worden geïmporteerd t.b.v. 'toestemmingen'.

Voorbeelden Zorgverlener

Zoek zorgverlener met UZI-nummer:

```
GET https://zab.test.lsp.aorta-zorg.nl/zab/fhir/Practitioner?identificer=http://fhir.nl/fhir/NamingSystem/uzi-nr-pers|123456789
```

(De system URI is in dit geval `http://fhir.nl/fhir/NamingSystem/uzi-nr-pers` en wordt door de vertical bar | gescheiden van het UZI-nummer 123456789.)

Zoek zorgverlener met AGB-code:

```
GET https://zab.test.lsp.aorta-zorg.nl/zab/fhir/Practitioner?identificer=http://fhir.nl/fhir/NamingSystem/agb-z|12345678
```

Zoek zorgverlener met BIG-nummer:

```
GET https://zab.test.lsp.aorta-zorg.nl/zab/fhir/Practitioner?identificer=http://fhir.nl/fhir/NamingSystem/big|12345678910
```

Zoek zorgverlener op ID zonder system URI (d.w.z. zonder het ID te 'scopen'), hiermee wordt er effectief gezocht naar zorgverleners met een UZI-nummer **of** een AGB-code **of** een BIG-nummer gelijk aan 123456789:

```
GET https://zab.test.lsp.aorta-zorg.nl/zab/fhir/Practitioner?identificer=123456789
```

(Er wordt gezocht op het type UZI, AGB_Zorgverlener en BIG.)

Voorbeelden PractitionerRole (=WorksAt)

Zoek 'rollen' met URA-nummer (dit geeft de zorgverleners die werkzaam zijn bij de zorgaanbieder en hun rol):

```
GET https://zab.test.lsp.aorta-zorg.nl/zab/fhir/PractitionerRole?organization.identificer=http://fhir.nl/fhir/NamingSystem/ura|12345678
```

(Het is mogelijk om andere system URI's te gebruiken, zie tabel hierboven.)

Zoek 'rollen' met UZI-nummer (dit geeft de zorgaanbieders waarbij de zorgverlener werkzaam is en zijn/haar rollen):

```
GET https://zab.test.lsp.aorta-zorg.nl/zab/fhir/PractitionerRole?practitioner.identificer=http://fhir.nl/fhir/NamingSystem/uzi-nr-pers|123456789
```

(Het is mogelijk om andere system URI's te gebruiken, zie tabel hierboven.)

4.8 Voorbeeld van de Zorgaanbiedertype-query ('JGZ-query')

De zorgaanbiedertype-query moet bij de FHIR-interface uitgevoerd worden via een GET request op de Organization resource met een querystring zoals in de onderstaande twee voorbeelden:

```
GET https://zab.test.lsp.aorta-zorg.nl/zab/fhir/Organization?type=[Code]&_summary=true
```

```
GET https://zab.test.lsp.aorta-zorg.nl/zab/fhir/Organization?type=[URI] | [Code] &_summary=true
```

Hierbij moet de queryparameter `_summary` gelijk zijn aan `true`.

Hierin moet `[Code]` worden vervangen door een zorgaanbiedertype-code, bijvoorbeeld 'JGZ'. Eventueel mag de code worden voorafgegaan door een URI om het codesysteem aan te geven. De URI en de Code moeten conform de FHIR specificatie worden gescheiden door een vertical bar ('|').

Voorbeeld:

```
GET https://zab.test.lsp.aorta-zorg.nl/zab/fhir/Organization?type=http://nictiz.nl/fhir/NamingSystem/organization-type|JGZ&_summary=true
```

In dit voorbeeld wordt er uitgegaan van de volgende waardelijst:

Naam	RoleCodeNL - zorgaanbiedertype (organisaties)
OID	2.16.840.1.113883.2.4.15.1060
URI	http://nictiz.nl/fhir/NamingSystem/organization-type
URL	https://decor.nictiz.nl/decor/services/RetrieveValueSet?id=2.16.840.1.113883.2.4.3.11.60.1.11.1&effectiveDate=2020-05-18T12:30:00&prefix=nictiz2bbr-&format=html&collapsible=true&language=nl-NL&ui=nl-NL

Met de query in het voorbeeld hierboven wordt in ZORG-AB gezocht op JGZ-instellingen (JeugdGezondheidsZorg-instellingen) of om precies te zijn op Organizations (zorgaanbieders) met in het `type` element een `coding` element met een `system` gelijk aan 'http://nictiz.nl/fhir/NamingSystem/organization-type' en een `code` gelijk aan 'JGZ'.

(Dit komt overeen met de attributen `type` en `code` in het `types` (meervoud) attribuut van de native REST interface, hoewel bij de REST interface `types.type` wordt gevuld met een codesysteem-code zoals 'NICTIZ' i.p.v. de corresponderende URI zoals 'http://nictiz.nl/fhir/NamingSystem/organization-type'.)

Na het versturen van dit request zal er een response worden gegeven zoals hieronder getoond:

```
{
  "resourceType": "Bundle",
  "entry": [
    {
      "resource": {
        "resourceType": "Organization",
        "active": true,
        "id": 1,
        "identifier": [
```

```

        {
          "system": "http://fhir.nl/fhir/NamingSystem/ura",
          "type": {
            "coding": [
              {
                "code": "PRN",
                "system": "http://hl7.org/fhir/ValueSet/identifier-type"
              }
            ]
          },
          "value": "00000001"
        }
      ],
      "meta": {
        "lastUpdated": "2018-03-14 19:49:38.938"
      },
      "name": "JGZ Voorbeeld 1",
      "_name": {
        "extension": [
          {
            "url": "http://hl7.org/fhir/StructureDefinition/TBD",
            "valueString": "Display"
          },
          {
            "url": "http://hl7.org/fhir/StructureDefinition/TBD",
            "valueBoolean": true
          }
        ]
      },
      "type": [
        {
          "coding": [
            {
              "code": "JGZ",
              "system": "http://nictiz.nl/fhir/NamingSystem/organization-type"
            }
          ]
        }
      ]
    }
  },
  {
    "resource": {
      "resourceType": "Organization",
      "active": true,
      "id": 2,
      "identifier": [
        {
          "system": "http://fhir.nl/fhir/NamingSystem/ura",
          "type": {
            "coding": [
              {
                "code": "PRN",
                "system": "http://hl7.org/fhir/ValueSet/identifier-type"
              }
            ]
          },
          "value": "00000002"
        }
      ],
      "meta": {
        "lastUpdated": "2018-03-14 19:49:38.938"
      },
      "name": "JGZ Voorbeeld 2",
      "_name": {
        "extension": [
          {
            "url": "http://hl7.org/fhir/StructureDefinition/TBD",
            "valueString": "Display"
          },
          {
            "url": "http://hl7.org/fhir/StructureDefinition/TBD",

```

```

        "valueBoolean": true
      }
    ],
    "type": [
      {
        "coding": [
          {
            "code": "JGZ",
            "system": "http://nictiz.nl/fhir/NamingSystem/organization-type"
          }
        ]
      }
    ]
  }
],
"link": [
  {
    "relation": "self",
    "url": "https://zab.test.lsp.aorta-zorg.nl/zab/fhir/Organization?type=http://nictiz.nl/fhir/NamingSystem/organization-type|JGZ&_summary=true"
  }
],
"total": 2,
"type": "searchset"
}

```

In het bovenstaande voorbeeld is niet getoond het element `type.coding.display`, echter kan ook deze worden opgeleverd. Hierin kan de weergavenaam conform het codesysteem worden opgeleverd, voorbeeld 'Jeugdgezondheidszorg'. (Dit is afhankelijk van de vulling van ZORG-AB.) Dit element correspondeert met het attribuut `DisplayName` van de klasse `Type` van het ZORG-AB datamodel. (Deze waarde wordt bij de REST interface opgeleverd in de array `types` (meervoud) in het attribuut `displayName`.)

~~De response kan ook als XML worden opgeleverd afhankelijk van de HTTP Accept header en/of de `_format` queryparameter (door bijvoorbeeld `&_format=xml` toe te voegen aan de querystring).~~

Een voorbeeld waarbij er geen URI wordt gebruikt in de querystring om het codesysteem aan te geven:

```
GET https://zab.test.lsp.aorta-zorg.nl/zab/fhir/Organization?type=JGZ&_summary=true
```

In dit geval wordt er gezocht op alle 'zorgaanbiedertype-types' die bekend (geconfigureerd) zijn, d.w.z. dat zorgaanbieders worden opgeleverd met een zorgaanbiedertype gelijk aan 'JGZ' ongeacht de waardelijst/codesysteem.

Er wordt per Organization een summary opgeleverd. Onderdeel van deze summary zijn de elementen waarbij in de FHIR-specificatie is aangegeven: `Summary=true`.⁶ Dit zijn de volgende elementen:

Naam	Toelichting
id	Technisch ID van organisatie
meta.lastUpdated	Laatst bijgewerkt datum

⁶ Zie: <http://hl7.org/fhir/STU3/resource-definitions.html>, <http://hl7.org/fhir/STU3/domainresource-definitions.html> en <http://hl7.org/fhir/STU3/organization-definitions.html>

identificer	URA-nummer
active	Geeft aan of organisatie actief is (in huidige versie altijd 'true')
type	Lijst van zorgaanbiedertypes
name	Gevelnaam (Dit veld bevat de naam die bij de native REST interface wordt opgeleverd met het type 'Display'.)
partOf	Hoofdorganisatie (in het geval van een sub-organisatie)

4.8.1 Toelichting LSP-applicatie ID's

Bij de JGZ-query via de **REST-interface** worden de LSP-applicatie ID's opgeleverd in het attribuut 'applicationIds', zie par. 3.11.

Bij de JGZ-query via de FHIR-interface, wordt er, zoals ook te zien aan de voorgaande tabel, geen vergelijkbaar attribuut opgeleverd. (De reden hiervoor is dat bij de Organization resource van de FHIR-interface de gegevens van de LSP-applicaties zich bevinden in de attributen 'contained' en 'endpoint'. Deze attributen zijn echter, volgens de FHIR-specificatie, geen onderdeel van de summary. Bij deze attributen is in de FHIR-specificatie niet aangegeven Summary=true. Zie ook de URL's in de eerder gegeven voetnoot.)

De gegevens van de LSP-applicaties, inclusief LSP-applicatie ID's, zijn **wel** beschikbaar via de FHIR-interface, maar moeten (in de huidige versie van ZORG-AB) worden opgehaald door (na het uitvoeren van de JGZ-query), voor elke zorgaanbieder (Organization) waarvoor de gegevens benodigd zijn, een separaat GET request uit te voeren met het ID van de Organization. (Dit geldt overigens ook voor andere gegevens die geen onderdeel zijn van de summary, zoals adres- en contactgegevens.)

Voorbeeld: aangenomen dat er een Organization is gevonden met ID 1, zoals in het eerder gegeven voorbeeld, kunnen de gegevens van de LSP-applicaties, inclusief de LSP-applicatie ID's, van de betreffende Organization worden opgehaald door het volgende request uit te voeren:

```
GET https://zab.test.lsp.aorta-zorg.nl/zab/fhir/Organization/1
```

In dit geval worden de LSP-applicaties opgeleverd in het attribuut 'contained'. De LSP-applicatie ID's worden opgeleverd in het attribuut 'identificer'. Voorbeeld:

```
{
  "resourceType": "Organization",
  ...
  "contained": [{
    "resourceType": "Endpoint",
    ...
    "id": "1",
    "identificer": [{
      "system": "urn:oid:2.16.840.1.113883.2.4.6.6",
      "value": "10006824"
    }
  ]
},
  ...
}
```

```

    ],
    "endpoint": [{
        "reference": "#1"
    }
    ],
    "id": "1",
    ...
}

```

NB 'identificer' is een array met, in het geval van LSP-applicaties, altijd precies één element. Het LSP-applicatie ID bevindt zich, om precies te zijn, in het attribuut 'value' van dit element. Het attribuut 'system' van LSP-applicatie ID's is altijd gelijk aan 'urn:oid:2.16.840.1.113883.2.4.6.6'.

4.8.2 Niet-Ondersteunde Nictiz Zorgaanbiedertypecode 'Z3' en Niet-Ondersteunde Vektis Zorgsoortcodes

Het via de zorgaanbiedertypequery ('JGZ-query') zoeken op Nictiz zorgaanbiedertypecode 'Z3' (Huisartspraktijk) en VZVZ zorgaanbiedertypecode 'Z004' (Huisartsenpraktijk) wordt in de huidige versie van ZORG-AB niet ondersteund. Er wordt in dit geval de volgende foutmelding opgeleverd:

```

{
  "resourceType": "OperationOutcome",
  "issue": [{
    "code": "invalid",
    "details": {
      "text": "Searching for organization type Nictiz Z3
VZVZ Z004 or Vektis Zorgsoort 01,04,05,12,41,76,89,90,94,96,98 via the
FHIR interface is currently not supported"
    },
    "severity": "error"
  }
]
}

```

Het is mogelijk om te zoeken op Vektis Zorgsoortcode, voorbeeld:

```

GET https://zab.test.lsp.aorta-
zorg.nl/zab/fhir/Organization?type=http://www.vzvv.nl/fhir/NamingSystem/ve
ktis-zorgsoort|02&_summary=true

```

Het zoeken op de volgende Vektis Zorgsoortcodes wordt in de huidige versie van ZORG-AB echter niet ondersteund:

Code	Weergavenaam
01	Huisartsen
04	Fysiotherapeuten
05	Logopedisten
12	Tandartsen
41	ZZP-ers in wijkverpleging/ PGB aanbieders / Beheerstichtingen
76	Leveranciers Hulpmiddelen
89	Schoonheidsspecialisten

90	Overige therapeuten en Complementair en Aanvullende zorg
94	Psychologische Zorgverleners
96	Pedicuren
98	Declaranten/Servicebureaus/Zorgverzekeraars

Er wordt in dit geval de hierboven genoemde foutmelding opgeleverd.

4.8.3 XML niet Ondersteund bij Zorgaanbiedertypequery

Wat betreft de zorgaanbiedertypequery ('JGZ-query') wordt het opleveren van de resultaten in XML formaat in de huidige versie van ZORG-AB eveneens niet ondersteund. Er wordt in dit geval de volgende foutmelding opgeleverd:

```
<?xml version="1.0" encoding="UTF-8"?>
<OperationOutcome xmlns="http://hl7.org/fhir">
  <issue>
    <severity value="error">
    </severity>
    <code value="invalid">
    </code>
    <details>
      <text value="For the organization type query returning results
in XML format is currently not supported">
      </text>
    </details>
  </issue>
</OperationOutcome>
```

4.9 Vrije Tekst Query's

Naast zoeken via de vaste zoekparameters (ook genoemd voorgedefinieerde query's, zie par. 4.5), ondersteunt ZORG-AB als alternatief het zoeken via vrije tekst query's. Hiermee kan op een zoekmachine-achtige manier worden gezocht, d.w.z. vergelijkbaar met hoe een zoekopdracht via een internetzoekmachine wordt uitgevoerd.

Voor een introductie op deze manier van zoeken en een aantal belangrijke aandachtspunten wordt verwezen naar par. 3.5, waar tevens wordt uitgelegd hoe de vrije tekst query's uitgevoerd kunnen worden via de native REST interface (ook genoemd OData interface).

Deze functionaliteit is eveneens beschikbaar via de FHIR interface van ZORG-AB. In deze paragraaf wordt daarvan een aantal voorbeelden gegeven.

Een voorbeeld van het zoeken via `_text`:

```
https://zab.test.lsp.aorta-
zorg.nl/zab/fhir/Organization?_text=janssen rijswijk
```

Hiermee wordt gezocht naar zorgaanbieders (Organizations) die matchen met de zoekexpressie 'janssen rijswijk'.

Voorbeeld met wild card (*):

```
https://zab.test.lsp.aorta-zorg.nl/zab/fhir/Organization?_text=jan*
rijswijk
```

Voorbeeld van zoeken naar zorgverlener (Practitioner):

```
https://zab.test.lsp.aorta-  
zorg.nl/zab/fhir/Practitioner?_text=huisarts janssen
```

Hiermee wordt gezocht naar zorgverleners (Practitioners) die matchen met de zoekexpressie 'huisarts janssen'.

Voorbeeld met wild card (*):

```
https://zab.test.lsp.aorta-  
zorg.nl/zab/fhir/Practitioner?_text=huisarts jan*
```

Voor aanvullende details over deze manier van zoeken zie, nogmaals, par. 3.5.

De `_text` parameter kan eventueel gebruikt worden i.c.m. de algemene queryparameters die in de volgende paragraaf worden beschreven (en met `_format`).

`_text` kan niet gebruikt worden i.c.m. identifier, name of andere vaste zoekparameters. Er verschijnt dan de volgende foutmelding:

```
Insufficient or unsupported search parameters. The following search  
parameters are supported: 'identifier', 'name' combined with 'address-city',  
'type', '_query'.
```

Zoeken via 'custom query' (Deprecated):

Er kan ook worden gezocht via een 'custom query'. Hierbij moet de parameter `'_query'` de waarde 'search' bevatten en moet daarnaast de parameter `'expr'` worden gebruikt met daarin de zoekexpressie.

Voorbeeld:

```
https://zab.test.lsp.aorta-  
zorg.nl/zab/fhir/Organization?_query=search&expr=janssen rijswijk
```

Deprecated vanaf ZORG-AB 2.6.6

Vanaf ZORG-AB 2.6.6 is de 'custom query' (zie het voorgaande voorbeeld) gemarkeerd als Deprecated, d.w.z. dat wordt aanbevolen om bij de ontwikkeling van nieuwe implementaties de 'custom query' niet te gebruiken.

Aanbevolen wordt om de parameter `'_text'` te gebruiken, zie hierboven.

Extra toelichting betreffende `'_query'` en `'expr'` (Deprecated):

- Als naam van de custom query is gekozen 'search', analoog aan '\$search' van de native REST interface.
- Voor deze 'custom query' is voor zowel de Organization als Practitioner resource een Operation Definition opgenomen in het Capability Statement, dat opgevraagd kan worden via `/zab/fhir/metadata`.

4.10 Gecombineerde Query's (LSP-query)

Toegevoegd in ZORG-AB 2.6.6

Het is mogelijk om een query uit te voeren op alleen de zorgaanbieders waaraan tenminste één LSP-applicatie is gekoppeld.

Deze query is vergelijkbaar met de GET `/lspOrganizations` query van de native REST interface (zie par. 3.6).

Deze query is een combinatie van een vrije-tekst query (par. 4.9) en zorgaanbiedertypequery (par. 4.8).

Er worden bij deze query de volgende zoekcriteria toegepast:

- De zoekwoord(en) die zijn aangegeven in de '_text' parameter (of 'expr' parameter (Deprecated))
- Aan de zorgaanbieder moet tenminste één LSP-applicatie zijn gekoppeld
 - Er wordt niet gefilterd op de status van de LSP-applicatie
- De zorgaanbiedertype(s) die zijn aangegeven in de 'type' parameter
 - Als er meerdere zorgaanbiedertypes zijn aangegeven dan wordt er gezocht naar zorgaanbieders die tenminste één van de aangegeven types hebben
 - Het codesysteem (URI) kan worden aangegeven bij het zorgaanbiedertype (gescheiden door een pipe '|') maar dit is niet verplicht. Wanneer het codesysteem wordt weggelaten wordt uitgegaan van de volgende codesystemen:
 - <http://nictiz.nl/fhir/NamingSystem/organization-type>
 - <http://www.vzvv.nl/fhir/NamingSystem/organization-type>

Voorbeeld:

```
https://zab.test.lsp.aorta-zorg.nl/zab/fhir/Organization?_text=janssen&endpoint.identifier.system=urn:oid:2.16.840.1.113883.2.4.6.6&type=http://nictiz.nl/fhir/NamingSystem/organization-type|J8,http://nictiz.nl/fhir/NamingSystem/organization-type|K3
```

Voorbeeld zonder codesysteem (URI) in 'type' parameter:

```
https://zab.test.lsp.aorta-zorg.nl/zab/fhir/Organization?_text=janssen&endpoint.identifier.system=urn:oid:2.16.840.1.113883.2.4.6.6&type=J8,K3
```

Voorbeeld met VZVZ-codesysteem:

```
https://zab.test.lsp.aorta-zorg.nl/zab/fhir/Organization?_text=janssen&endpoint.identifier.system=urn:oid:2.16.840.1.113883.2.4.6.6&type=http://www.vzvv.nl/fhir/NamingSystem/organization-type|Z001,http://www.vzvv.nl/fhir/NamingSystem/organization-type|Z002
```

Alle parameters ('_text', 'endpoint.identifier.system' en 'type') zijn in het geval van deze query verplicht.

Voorbeeld waarbij '_query' en 'expr' parameter worden gebruikt (Deprecated):

```
https://zab.test.lsp.aorta-zorg.nl/zab/fhir/Organization?_query=search&expr=janssen&endpoint.identifier.system=urn:oid:2.16.840.1.113883.2.4.6.6&type=http://nictiz.nl/fhir/NamingSystem/organization-type|J8,http://nictiz.nl/fhir/NamingSystem/organization-type|K3
```

Alle parameters ('_query', 'expr', 'endpoint.identifier.system' en 'type') zijn in het geval van deze query verplicht.

Read-Only Interface:

De hierboven genoemde query's kunnen ook uitgevoerd worden via de read-only interface (zie par. 1.2.1), bijvoorbeeld:

```
https://zab.prod.aorta-zorg.nl/zab-ro/fhir/Organization?_text=janssen&endpoint.identifier.system=urn:oid:2.16.840.1.113883.2.4.6.6&type=http://nictiz.nl/fhir/NamingSystem/organization-type|J8,http://nictiz.nl/fhir/NamingSystem/organization-type|K3
```

4.11 Algemene Queryparameters

De volgende algemene queryparameters worden ondersteund (naast de zoekparameters beschreven in par. 4.5):

Parameter	Toelichting
_id	Als alternatief voor GET /[type]/[id] kan GET /[type]?_id=[id] worden gebruikt.
_summary	<ul style="list-style-type: none"> Als 'true' wordt een 'samenvatting' teruggegeven i.p.v. de volledige set attributen. Welke attributen onderdeel zijn van de 'samenvatting' is gedefinieerd door FHIR. Voor het uitvoeren van een query op resource Organization met zoekparameter 'type' moet <code>_summary=true</code> worden meegegeven.
_elements	I.p.v. de <code>_summary</code> parameter kan de parameter <code>_elements</code> worden meegegeven met een komma-gescheiden lijst van attribuutnamen. De teruggegeven attributen worden dan beperkt tot deze lijst. (Als zowel <code>_summary</code> als <code>_elements</code> worden meegegeven wordt alleen <code>_summary</code> verwerkt.)
_filter	De voorgedefinieerde query's kunnen ook meegegeven worden in de parameter <code>_filter</code> , hierbij wordt een formaat gehanteerd dat vergelijkbaar is met het formaat van de native REST interface. Voor details zie https://www.hl7.org/fhir/search_filter.html . (Vooralsnog) worden alleen de operators 'eq' and 'and' ondersteund.

4.12 PATCH

JSON Patch, XML Patch of FHIRPath Patch (link: <https://www.hl7.org/fhir/http.html#patch>) worden (nog) niet ondersteund. Wel worden er, net als bij de native REST interface, 'simpele' PATCH requests ondersteund waarbij er een Resource meegegeven wordt (zoals bij een PUT request) maar met een subset van de attributen. In dit geval worden alléén de meegegeven attributen bijgewerkt.

Om een voorbeeld te geven, wanneer het volgende PATCH request wordt uitgevoerd op een bestaande Organization dan wordt alleen het 'address' attribuut bijgewerkt en blijven eventuele overige attributen (zoals

'name', 'alias', 'telecom') ongewijzigd. PATCH is (net als POST en PUT) zowel mogelijk met een JSON request body als met een XML request body.

```
{
  "address": [{
    "city": "DEN HAAG",
    "line": [
      "Oude Middenweg 55"
    ],
    "postalCode": "2491 AC",
    "text": "Oude Middenweg 55\n2491 AC DEN HAAG",
    "type": "physical",
    "use": "work"
  ]
}
```

4.13 Voorbeeld van het Toevoegen van Beschikbaarheidsgegevens van een Zorgverlener

Via de native REST interface kunnen beschikbaarheidsgegevens van een zorgverlener worden toegevoegd aan het veld 'availability' van de WorksAt-relatie. Voorgesteld wordt dat het iCalendar-formaat (RFC 5545) wordt gebruikt, maar dit wordt niet technisch afgedwongen. Zie ook par. 3.17.

Analoog kunnen de beschikbaarheidsgegevens via de FHIR-interface worden toegevoegd aan het veld 'extension' van de PractitionerRole resource. Hiervan wordt een voorbeeld gegeven in deze paragraaf.

Eerst moet het ID worden bepaald van de PractitionerRole resource. Hiervoor kan bijvoorbeeld gezocht worden op het UZI-nummer van de zorgverlener via het volgende GET-request:

```
https://zab.test.lsp.aorta-zorg.nl/zab/fhir/PractitionerRole?practitioner.identificer=http://fhir.nl/fhir/NamingSystem/uzi-nr-pers|00001234
```

(Hierin moet 00001234 worden vervangen door het daadwerkelijke UZI-nummer.)

Het ID wordt vervolgens opgeleverd in het veld 'id'. Het is mogelijk dat de zorgverlener werkzaam is voor meerdere organisaties en er daardoor meerdere PractitionerRole resources worden opgeleverd. In dat geval moet het ID gebruikt worden van de PractitionerRole resource die correspondeert met de zorgaanbieder (Organization) waarop de beschikbaarheidsgegevens van toepassing zijn.

Vervolgens moet het volgende PATCH-request worden uitgevoerd:

```
PATCH https://zab.test.lsp.aorta-zorg.nl/zab/fhir/PractitionerRole/101112
```

(Hierin moet 101112 worden vervangen door het daadwerkelijke ID van de PractitionerRole resource.)

Met in de body het volgende JSON-object:

```
{
  "extension": [
    {
      "url": "http://hl7.org/fhir/StructureDefinition/TBD",
      "valueString": "BEGIN:VCALENDAR\r\nVERSION:2.0\r\nPRODID:-//Huisartsenpraktijk J. Janssen//NONSGML v1.0//EN\r\nBEGIN:VEVENT\r\nUID:19970610T172345Z-
```

```
AF23B2@example.com\r\nDTSTAMP:20180615T120000\r\nDTSTART:20180618T090000\r\nRRULE:FREQ=WEEKLY;BYDAY=MO;BYHOUR=8,9,10,11,12,13,14\r\nSUMMARY:Dhr. J. Janssen is aanwezig op maandag van 08:00 tot 15:00\r\nEND:VEVENT\r\nBEGIN:VEVENT\r\nUID:19970610T172345Z-AF23B2@example.com\r\nDTSTAMP:20180615T120000\r\nDTSTART:20180618T090000\r\nRRULE:FREQ=WEEKLY;BYDAY=TU;BYHOUR=8,9,10\r\nSUMMARY:Dhr. J. Janssen is aanwezig op dinsdag van 08:00 tot 11:00\r\nEND:VEVENT\r\nEND:VCALENDAR\r\n"
    }
  ]
}
```

Er wordt vervolgens een response gegeven met een HTTP-statuscode 200 OK om aan te geven dat het request succesvol is verwerkt.

4.14 Voorbeeld van het Toevoegen van EDIFACT Emailadres

EDIFACT emailadressen kunnen worden toegevoegd door middel van het volgende PATCH request:

```
PATCH https://zab.test.lsp.aorta-zorg.nl/zab/fhir/Organization/ [ID]
```

Hierin moet **[ID]** worden vervangen door het ID van de zorgaanbieder (Organization). De volgende request body moet worden gebruikt:

```
{
  "resourceType": "Organization",
  "contained": [{
    "resourceType": "Endpoint",
    "status": "active",
    "connectionType": {
      "system": "http://hl7.org/fhir/ValueSet/endpoint-connection-type",
      "code": "EDIFACT"
    },
    "payloadType": [{
      "coding": [{
        "system": "http://hl7.org/fhir/ValueSet/endpoint-payload-type",
        "code": "EDIFACT"
      }]
    }]
  }],
  "address": "[Emailadres]"
},
"endpoint": [{
  "reference": "#1"
}]
}
```

Hierin moet **[Emailadres]** worden vervangen door het emailadres van de zorgaanbieder (Organization).

Het emailadres kan worden gewijzigd via hetzelfde PATCH request maar dan met in de request body het nieuwe emailadres.

Het emailadres kan worden verwijderd via hetzelfde PATCH request maar dan met de volgende request body:

```
{
  "resourceType": "Organization",
  "contained": [],
  "endpoint": []
}
```

Let op dat als er een PATCH request wordt gestuurd waarin het 'contained' attribuut aanwezig is alle Endpoints (=Electronic Services) die aan de zorgaanbieder zijn gekoppeld worden vervangen door de Endpoints in het 'contained' attribuut. Wanneer er één Endpoint in het 'contained' attribuut is geplaatst zullen alle gekoppelde Endpoints worden vervangen door dat Endpoint. Wanneer er een leeg 'contained' attribuut in het request is geplaatst zullen alle gekoppelde Endpoints worden verwijderd.

Dit geldt alleen voor 'muteerbare' Endpoints. Wanneer er bijvoorbeeld LSP applicaties zijn gekoppeld aan de zorgaanbieder dan worden deze niet gewijzigd/verwijderd. Het is in dat geval niet nodig om bij het sturen van de PATCH de LSP applicaties in het 'contained' attribuut te plaatsen. Om, ervan uitgaande dat er alleen LSP applicaties zijn gekoppeld en nog geen andere Endpoints, een 'EDIFACT Endpoint' toe te voegen hoeft alleen het 'EDIFACT Endpoint' in het 'contained' attribuut te worden geplaatst.

Als het 'contained' attribuut geheel is weggelaten in het PATCH request worden er geen Endpoints toegevoegd/gewijzigd/verwijderd. (Alleen de attributen die aanwezig zijn in het PATCH request worden bijgewerkt.)

5 Koppelingen met Externe Gegevensbronnen

5.1 Vektis AGB Services

ZORG-AB is gekoppeld aan de AGB Services van Vektis. Via deze koppeling worden mutaties van ondernemingen, vestigingen en zorgverleners dagelijks opgehaald en verwerkt in ZORG-AB. Het betreft de volgende gegevens:

- Zorgaanbieder (Organization)
 - Identifications
 - Type 'AGB_Onderneming' of 'AGB_Vestiging' in het geval van een vestiging (SubOrganization)
 - Type 'KVK'
 - Names
 - Type 'Display'
 - Type 'Legal'
 - Type 'Tradename'
 - NB Wordt (voorlopig) niet opgeleverd i.v.m. backward compatibility met de waardelijst (enum) in de OAS van ZORG-AB 2.6.2
 - Addresses
 - Type 'Practice'
 - Type 'Postal'
 - Type 'Billing'
 - Type 'Delivery'
 - Telecoms
 - Type 'Phone'
 - Type 'Email'
 - Type 'Fax'
 - Type 'URL'
 - NB Wordt (voorlopig) niet opgeleverd i.v.m. backward compatibility met de waardelijst (enum) in de OAS van ZORG-AB 2.6.2
 - Types
 - Type 'Vektis Zorgsoort'
 - Comment
- Zorgverlener (Individual)
 - Identifications
 - Type 'AGB_Zorgverlener'
 - Names
 - Type 'Display'
 - Credentials

- Kwalificatie, bijvoorbeeld:

Type	Description
0200	Apothekers

- Gender
- WorksAt
 - Speciality
 - Vektis Zorgaanbodcode
 - NB Wordt (voorlopig) niet opgeleverd i.v.m. backward compatibility met de waardelijst (enum) in de OAS van ZORG-AB 2.6.2

5.2 VZVZ Clientèlebestand (BI-bestand)

Het clientèlebestand (BI-bestand) van VZVZ wordt dagelijks geïmporteerd in ZORG-AB om nieuwe zorgaanbieders toe te voegen en (adres)gegevens van bestaande zorgaanbieders bij te werken. Het gaat per zorgaanbieder (Organization) om de volgende gegevens:

Indien er geen sprake is van 'locaties':

- URA-nummer
- LSP Applicatie ID('s)
- URA naam
- URA adres
- Zorgaanbiedertype(s)
- AGB-code(s)

Indien er sprake is van 'locaties' worden er gegevens toegevoegd/bijgewerkt bij de hoofdorganisatie en bij de suborganisaties.

Bij de hoofdorganisatie worden de volgende gegevens toegevoegd/bijgewerkt:

- URA-nummer
- URA naam
- URA adres

Bij de suborganisaties worden de volgende gegevens toegevoegd/bijgewerkt:

- LSP Applicatie ID('s)
- Locatiennaam
- Locatieadres
- Zorgaanbiedertype(s)
- AGB-code(s)

Alle gegevens die vanuit het BI-bestand worden geïmporteerd hebben author 'VZVZ'.

Alle namen die vanuit het BI-bestand worden geïmporteerd hebben type 'Display'.

Alle adressen die vanuit het BI-bestand worden geïmporteerd hebben type 'Practice' of in het geval van een postbus type 'Postal'.

Alle AGB-codes die vanuit het BI-bestand worden geïmporteerd hebben type 'AGB_Zorgaanbieder'.

5.3 UZI LDAP

Daarnaast worden er dagelijks gegevens geïmporteerd vanuit UZI LDAP. Dit betreft voor zorgaanbieders (Organizations) de UZI-naam. Deze wordt in ZORG-AB getoond als een OrganizationName met Type 'Legal'.

Voor zorgverleners (Individuals) betreft dit de volgende gegevens:

- UZI-nummer
- Rolcode conform [deze](#) definitie

Nieuwe zorgverleners (Individuals) worden aangemaakt in ZORG-AB. Er wordt dan ook een WorksAt relatie aangemaakt met de Organization waaraan de zorgverlener is gekoppeld. Wanneer een zorgverlener in UZI LDAP wordt gekoppeld aan een andere zorgaanbieder wordt er een nieuwe WorksAt relatie aan ZORG-AB toegevoegd.

NB Alleen de zorgverleners die gekoppeld zijn aan een URA nummer dat is opgenomen in ZORG-AB worden geladen.

Persoonsnamen worden niet geladen vanuit UZI LDAP.

5.4 Live Koppeling met het LSP

Per zorgaanbieder worden bovendien dagelijks de LSP-applicatie ID's opgehaald uit het applicatieregister (APR) van het LSP en zo nodig toegevoegd aan de zorgaanbieder (Organization).

Daarnaast worden, op het moment dat via één van de web service interfaces van ZORG-AB een zorgaanbieder wordt opgevraagd, de actuele LSP-applicatiegegevens live opgehaald uit het applicatieregister (APR) van het LSP. Dit betreft de volgende gegevens:

- Status
- Adres
- Systemrollen
- Conformances

Via de dagelijkse import worden dus alleen de ID's opgehaald. Oude ID's (ID's die niet meer aan de betreffende URA zijn gekoppeld in het APR of fysiek zijn verwijderd) worden automatisch opgeschoond. Via de live koppeling worden alle overige gegevens (hierboven genoemd) opgehaald.

ZORG-AB bevat ook een mogelijkheid om contactgegevens (Telecoms) alleen zichtbaar te maken voor collega-zorgaanbieders en niet voor patiëntenportalen (GBP's). Zie voor details paragraaf 3.16. De bepaling of een organisatie een GBP is of niet vindt ook plaats via de live koppeling met het applicatieregister (APR) van het LSP.

5.5 Koppeling met MedMij ZKL (ZorgaanbiedersKoppelLijst)

ZORG-AB is gekoppeld aan de MedMij ZKL (ZorgaanbiedersKoppelLijst). De ZKL wordt dagelijks opgehaald en geïmporteerd in ZORG-AB. Het betreft de volgende gegevens:

- Zorgaanbieder (Organization)
 - Identifications
 - Type 'MedMij'
 - Electronic Services
 - Instanties van klasse 'MedMij Gegevensdienst' waarbij de volgende attributen worden gevuld:
 - Gegevensdienstld
 - Weergavenaam
 - Interfaceversies

AuthorizationEndpoint, TokenEndpoint, Abonneren en Systeemrollen worden niet gevuld.

Zie ook par. 3.19.

NB De zorgaanbieders in de ZKL worden gematcht met de zorgaanbieders in ZORG-AB o.b.v. URA-nummer en/of AGB-code.

Indien er geen match heeft kunnen plaatsvinden gaat het mogelijk om een nieuwe zorgaanbieder die nog niet is geïmporteerd vanuit Vektis en/of VZVZ Clientèle (zie ook par. 5.1 en 5.2). In dat geval zullen zodra de zorgaanbieder is geïmporteerd vanuit Vektis en/of VZVZ Clientèle de gegevens van de betreffende zorgaanbieder alsnog worden geïmporteerd vanuit de ZKL.

Er wordt gematcht met de AGB-codes met type 'AGB_Onderneming' en 'AGB_Vestiging'. Er wordt niet gematcht met de AGB-codes met type 'AGB_Zorgaanbieder'.

Bijlage 1 Lijst van Geldige URI's (Native REST Interface)

Onderstaande tabel geeft een opsomming van de geldige HTTP-methodes, URI's en queryparameters die beschikbaar zijn in de huidige versie:

HTTP-Methode	Geldige URI's	Query Parameters	Toelichting
GET	/organizations	\$filter, \$search, \$select	Voor het uitvoeren van een query op de zorgaanbieders (Organizations) in ZORG-AB. Zie paragraaf 3.4 en paragraaf 3.5 voor een toelichting op het gebruik van \$filter en \$search
	/organizations/{id}	\$select	Geeft zorgaanbieder met aangegeven ID
	/organizations/_summary	\$filter, \$select	Idem als /organizations met de volgende uitzonderingen: <ul style="list-style-type: none"> • Er wordt een summary object (subset van de beschikbare velden) teruggegeven • Op deze URI (resource) kan de zorgaanbiedertype-query worden uitgevoerd, bijvoorbeeld: <code>/organizations/_summary?\$filter=type eq 'JGZ'</code> Zie voor details par. 3.11.
	/organizations/{id}/_summary	\$select	Geeft summary object van zorgaanbieder met aangegeven ID
	/individuals	\$filter, \$search, \$select	Voor het uitvoeren van een query op de zorgverleners (Individuals) in ZORG-AB. Zie paragraaf 3.4 en paragraaf 3.5 voor een toelichting op het gebruik van \$filter en \$search
	/individuals/{id}	\$select	Geeft zorgverlener met aangegeven ID
	/organizations/{id}/subOrganizations	\$select	Suborganisaties van een hoofdorganisatie
	/organizations/{id}/mainOrganization	\$select	Hoofdorganisatie van een suborganisatie
	/organizations/{id}/individuals	\$select	Zorgverleners die werkzaam zijn bij een zorgaanbieder
	/individuals/{id}/organizations	\$select	Zorgaanbieders waarvoor een zorgverlener werkzaam is
	/individuals/{id}/relations	\$select	Geeft de 'WorksAt'-relaties van een zorgverlener
	/organizations/{id}/relations	\$select	Geeft de 'SubOrganization'- en 'WorksAt'-relaties van een zorgaanbieder

	/subOrganization/{id}	\$select	Geeft 'SubOrganization'-relatie met aangegeven ID
	/worksAt/{id}	\$select	Geeft 'WorksAt'-relatie met aangegeven ID
	/ura/{ura}	\$select	Geeft zorgaanbieder met het aangegeven URA-nummer
	/uzi/{uzi}	\$select	Geeft zorgverlener met het aangegeven UZI-nummer
	/applicationId/{appld}	\$select	<p>Geeft zorgaanbieder met het aangegeven LSP applicatie-ID.</p> <p>In het geval van een zorgaanbieder zonder 'locaties' wordt simpelweg die zorgaanbieder opgeleverd. Bijvoorbeeld in het geval applicatie-ID 111 is gekoppeld aan zorgaanbieder 123 en er wordt gezocht op applicatie-ID 111 wordt zorgaanbieder 123 opgeleverd.</p> <p>In het geval een zorgaanbieder 'locaties' heeft is er sprake van een hoofdorganisatie en suborganisaties waarbij het applicatie-ID is gekoppeld aan één van de suborganisaties. Er is dan bijvoorbeeld sprake van de volgende structuur:</p> <ul style="list-style-type: none"> • Hoofdorganisatie 123 <ul style="list-style-type: none"> ○ Suborganisatie 456 <ul style="list-style-type: none"> ▪ Applicatie 111 ○ Suborganisatie 789 <ul style="list-style-type: none"> ▪ Applicatie 222 <p>Als in dit geval wordt gezocht op applicatie 111 dan wordt hoofdorganisatie 123 opgeleverd.</p> <p>Ook als wordt gezocht op applicatie 222 wordt hoofdorganisatie 123 opgeleverd.</p> <p>Om de specifieke suborganisatie te vinden waaraan het applicatie-ID is gekoppeld zijn er twee mogelijkheden:</p> <ul style="list-style-type: none"> • De gegevens ophalen d.m.v. een 2e request: GET /organizations/123/subOrganizations (zie boven). In dit voorbeeld zou daarmee een array worden opgeleverd met daarin 456 en 789. • I.p.v. GET /applicationId/111 gebruiken GET /applicationId-v2/111 (zie onder). In dit voorbeeld zou daarmee een array

			worden opgeleverd met daarin 123, 456 en 789.
	/applicationId-v2/{appld} <i>Toegevoegd in ZORG-AB 2.7.0</i>	\$select	<p>Geeft zorgaanbieder met het aangegeven LSP applicatie-ID.</p> <p>Geeft als er sprake is van 'locaties' niet alleen de hoofdorganisatie maar ook de suborganisaties.</p> <p>Geeft in tegenstelling tot GET /applicationId (zie boven) een array.</p> <p>In het geval van een zorgaanbieder zonder 'locaties' bevat de array alleen die zorgaanbieder. Bijvoorbeeld in het geval applicatie-ID 111 is gekoppeld aan zorgaanbieder 123 en er wordt gezocht op applicatie-ID 111 wordt een array opgeleverd met daarin alleen zorgaanbieder 123.</p> <p>In het geval een zorgaanbieder 'locaties' heeft is er sprake van een hoofdorganisatie en suborganisaties waarbij het applicatie-ID is gekoppeld aan één van de suborganisaties. Er is dan bijvoorbeeld sprake van de volgende structuur:</p> <ul style="list-style-type: none"> • Hoofdorganisatie 123 <ul style="list-style-type: none"> ○ Suborganisatie 456 <ul style="list-style-type: none"> ▪ Applicatie 111 ○ Suborganisatie 789 <ul style="list-style-type: none"> ▪ Applicatie 222 <p>Als in dit geval wordt gezocht op applicatie 111 dan wordt een array opgeleverd met daarin hoofdorganisatie 123, suborganisatie 456 en suborganisatie 789.</p> <p>Ook als wordt gezocht op applicatie 222 wordt een array opgeleverd met daarin hoofdorganisatie 123 en suborganisaties 456 en 789.</p> <p><u>De hoofdorganisatie wordt altijd opgeleverd op de 1e positie in de array.</u></p> <p>De suborganisaties bevinden zich dus op de 2e positie in de array en de posities daarna.</p> <p>De specifieke suborganisatie waaraan het applicatie-ID is gekoppeld kan dus ook worden gevonden op de 2e positie of een positie daarna.</p> <p>Bij de hoofdorganisatie worden in het attribuut 'applicationIds' de applicatie-ID's opgeleverd van alle suborganisaties. Bij de suborganisaties wordt in het attribuut 'applicationIds' alleen het</p>

			specifieke applicatie-ID opgeleverd dat bij die suborganisatie hoort.
	/applications/{appld}	\$select	Geeft applicatie met het aangegeven LSP applicatie-ID
	/lspOrganizations <i>Toegevoegd in ZORG-AB 2.6.6</i>	\$filter, \$search, \$select	<p>Voor het uitvoeren van een vrije-tekst query op een subset van de zorgaanbieders (Organizations) in ZORG-AB, namelijk de zorgaanbieders waaraan tenminste één LSP-applicatie is gekoppeld.</p> <p>Om dit te verduidelijken, bij GET /organizations worden alle zorgaanbieders doorzocht maar bij GET /lspOrganizations worden alleen de zorgaanbieders doorzocht waaraan tenminste één LSP-applicatie is gekoppeld. Afhankelijk van de zoekparameters worden nul of meer van deze "zorgaanbieders waaraan tenminste één LSP-applicatie is gekoppeld" opgeleverd.</p> <p>Voor /lspOrganizations geldt:</p> <ul style="list-style-type: none"> • Via \$filter kan alleen worden gefilterd op zorgaanbiedertype (of evt. meerdere zorgaanbiedertypes) • \$search is verplicht <p>Zie paragraaf 3.6 voor een toelichting op het gebruik van \$filter en \$search in dit geval.</p>
	/identifications/{id}	\$select	<p>Geeft zorgaanbieders/zorgverleners (Organizations/Individuals) met het aangegeven ID. Het ID moet beginnen met een prefix waarmee het type van de identificatie wordt aangegeven, voorbeelden:</p> <ul style="list-style-type: none"> • /identifications/URA:12345678 • /identifications/UZI:123456789 <p>Zie voor voorbeelden van het zoeken op AGB-code, KVK-nummer, MedMij zorgaanbiedernaam en BIG-nummer par. 3.1 onder "/identifications".</p>
POST	/organizations		Voor het toevoegen van een zorgaanbieder
	/organizations/{id}/addresses		Voor het toevoegen van een adres aan een zorgaanbieder
	/organizations/{id}/names		Voor het toevoegen van een naam aan een zorgaanbieder
	/organizations/{id}/telecoms		Voor het toevoegen van een contactgegeven aan een zorgaanbieder
	/organizations/{id}/identifications		Voor het toevoegen van een identificatie aan een zorgaanbieder

	/organizations/{id}/services		Voor het toevoegen van een elektronische dienst aan een zorgaanbieder
	/organizations/{id}/credentials		Voor het toevoegen van een credential aan een zorgaanbieder
	/individuals		Voor het toevoegen van een zorgverlener
	/individuals/{id}/addresses		Voor het toevoegen van een adres aan een zorgverlener
	/individuals/{id}/names		Voor het toevoegen van een naam aan een zorgverlener
	/individuals/{id}/telecoms		Voor het toevoegen van een contactgegeven aan een zorgverlener
	/individuals/{id}/identifications		Voor het toevoegen van een identificatie aan een zorgverlener
	/individuals/{id}/services		Voor het toevoegen van een elektronische dienst aan een zorgverlener
	/individuals/{id}/credentials		Voor het toevoegen van een adres aan een zorgverlener
	/subOrganization		Voor het toevoegen van een 'SubOrganization'-relatie, hiermee wordt aangegeven dat een organisatie een suborganisatie is van een andere (hoofd)organisatie, zie paragraaf 3.14 voor een voorbeeld
	/worksAt		Voor het toevoegen van een 'WorksAt'-relatie, hiermee wordt aangegeven dat een zorgverlener werkzaam is bij een zorgaanbieder, zie paragraaf 3.15 voor een voorbeeld
PATCH	/organizations/{id}		Wijzigt de zorgaanbieder met het aangegeven ID
	/organizations/{id}/addresses/{index}		Wijzigt het adres op de aangegeven positie (1-gebaseerd)
	/organizations/{id}/names/{index}		Wijzigt de naam op de aangegeven positie (1-gebaseerd)
	/organizations/{id}/telecoms/{index}		Wijzigt het contactgegeven op de aangegeven positie (1-gebaseerd)
	/organizations/{id}/identifications/{index}		Wijzigt de identificatie op de aangegeven positie (1-gebaseerd)
	/organizations/{id}/services/{index}		Wijzigt de elektronische dienst op de aangegeven positie (1-gebaseerd)
	/organizations/{id}/credentials/{index}		Wijzigt de credential op de aangegeven positie (1-gebaseerd)

	/individuals/{id}		Wijzigt het zorgverlener met het aangegeven ID
	/individuals/{id}/addresses/{index}		Wijzigt het adres op de aangegeven positie (1-gebaseerd)
	/individuals/{id}/names/{index}		Wijzigt de naam op de aangegeven positie (1-gebaseerd)
	/individuals/{id}/telecoms/{index}		Wijzigt de telecom op de aangegeven positie (1-gebaseerd)
	/individuals/{id}/identifications/{index}		Wijzigt de identificatie op de aangegeven positie (1-gebaseerd)
	/individuals/{id}/services/{index}		Wijzigt de elektronische dienst op de aangegeven positie (1-gebaseerd)
	/individuals/{id}/credentials/{index}		Wijzigt de credential op de aangegeven positie (1-gebaseerd)
	/subOrganization/{id}		Wijzigt de 'SubOrganization'-relatie met het aangegeven ID
	/subOrganization		Idem als hierboven behalve dat er geen ID wordt meegegeven in het pad. Wel moeten een 'mainOrganizationId' en 'subOrganizationId' meegegeven worden in de body. Hiermee wordt de 'SubOrganization'-relatie geïdentificeerd
	/worksAt/{id}		Wijzigt de 'WorksAt'-relatie met het aangegeven ID
	/worksAt		Idem als hierboven behalve dat er geen ID wordt meegegeven in het pad. Wel moeten een 'organizationId' en 'individualId' meegegeven worden in de body. Hiermee wordt de 'WorksAt'-relatie geïdentificeerd
PUT	-		Wordt niet ondersteund
DELETE	/organizations/{id}		Verwijdert de zorgaanbieder met het aangegeven ID
	/organizations/{id}/addresses/{index}		Verwijdert het adres op de aangegeven positie (1-gebaseerd)
	/organizations/{id}/names/{index}		Verwijdert de naam op de aangegeven positie (1-gebaseerd)
	/organizations/{id}/telecoms/{index}		Verwijdert het contactgegeven op de aangegeven positie (1-gebaseerd)

	/organizations/{id}/identifications/{index}		Verwijdert de identificatie op de aangegeven positie (1-gebaseerd)
	/organizations/{id}/services/{index}		Verwijdert de elektronische dienst op de aangegeven positie (1-gebaseerd)
	/organizations/{id}/credentials/{index}		Verwijdert de credential op de aangegeven positie (1-gebaseerd)
	/individuals/{id}		Verwijdert het zorgverlener met het aangegeven ID
	/individuals/{id}/addresses/{index}		Verwijdert het adres op de aangegeven positie (1-gebaseerd)
	/individuals/{id}/names/{index}		Verwijdert de naam op de aangegeven positie (1-gebaseerd)
	/individuals/{id}/telecoms/{index}		Verwijdert het contactgegeven op de aangegeven positie (1-gebaseerd)
	/individuals/{id}/identifications/{index}		Verwijdert de identificatie op de aangegeven positie (1-gebaseerd)
	/individuals/{id}/services/{index}		Verwijdert de elektronische dienst op de aangegeven positie (1-gebaseerd)
	/individuals/{id}/credentials/{index}		Verwijdert de credential op de aangegeven positie (1-gebaseerd)
	/subOrganization/{id}		Verwijdert de 'SubOrganization'-relatie met het aangegeven ID
	/worksAt/{id}		Verwijdert de 'WorksAt'-relatie met het aangegeven ID
	/subOrganization		In dit geval moeten de 'mainOrganizationId' en 'subOrganizationId' meegegeven worden in de body om de betreffende 'SubOrganization'-relatie te verwijderen (m.a.w. de 'SubOrganization'-relatie tussen de organisaties met de in de body aangegeven ID's wordt verwijderd)
	/worksAt		In dit geval moeten de 'organizationId' en 'individualId' meegegeven worden in de body om de betreffende 'WorksAt'-relatie te verwijderen (m.a.w. de 'WorksAt'-relatie tussen de zorgaanbieder en zorgverlener met de in de body aangegeven ID's wordt verwijderd)

Bijlage 2 Lijst van Geldige URI's (FHIR Interface)

FHIR beschrijft een set van interacties (<http://hl7.org/fhir/STU3/http.html>), deze worden in onderstaande tabel opgesomd met de bijbehorende HTTP-methode(s), URI's en toelichting. De gemarkeerde interacties worden ondersteund door ZORG-AB. Voor meer details (ondersteunde content types, queryparameters, voorbeeldrequests etc.) zie hoofdstuk 4.

Turquoise = Ondersteund

Niveau	Interactie	HTTP-methode	URI	Toelichting
Instance Level	read	GET	/Organization/{id}	Geeft zorgaanbieder met ID {id}
			/Practitioner/{id}	Geeft zorgverlener met ID {id}
			/PractitionerRole/{id}	Geeft rol (WorksAt-relatie) met ID {id}
			/Endpoint/{id}	(Nog) niet ondersteund
			/Location/{id}	(Nog) niet ondersteund
	vread	GET	/Organization/{id}/_history/{vid}	(Nog) niet ondersteund
			/Practitioner/{id}/_history/{vid}	(Nog) niet ondersteund
			/PractitionerRole/{id}/_history/{vid}	(Nog) niet ondersteund
			/Endpoint/{id}/_history/{vid}	(Nog) niet ondersteund
			/Location/{id}/_history/{vid}	(Nog) niet ondersteund
	update	PUT	/Organization/{id}	Vervangt zorgaanbieder met ID {id} door in request body meegegeven zorgaanbieder
			/Practitioner/{id}	Vervangt zorgverlener met ID {id} door in request body meegegeven zorgverlener
			/PractitionerRole/{id}	Vervangt rol (WorksAt-relatie) met ID {id} door in request body meegegeven rol (WorksAt-relatie)
			/Endpoint/{id}	(Nog) niet ondersteund
			/Location/{id}	(Nog) niet ondersteund
	patch	PATCH	/Organization/{id}	Werkt zorgaanbieder met ID {id} bij
			/Practitioner/{id}	Werkt zorgverlener met ID {id} bij
			/PractitionerRole/{id}	Werkt rol (WorksAt-relatie) met ID {id} bij

			/Endpoint/{id}	(Nog) niet ondersteund
			/Location/{id}	(Nog) niet ondersteund
	delete	DELETE	/Organization/{id}	Verwijdert zorgaanbieder met ID {id}
			/Practitioner/{id}	Verwijdert zorgverlener met ID {id}
			/PractitionerRole/{id}	Verwijdert rol (WorksAt-relatie) met ID {id}
			/Endpoint/{id}	(Nog) niet ondersteund
			/Location/{id}	(Nog) niet ondersteund
	history	GET	/Organization/{id}/_history	(Nog) niet ondersteund
			/Practitioner/{id}/_history	(Nog) niet ondersteund
			/PractitionerRole/{id}/_history	(Nog) niet ondersteund
			/Endpoint/{id}/_history	(Nog) niet ondersteund
			/Location/{id}/_history	(Nog) niet ondersteund
Type Level	create	POST	/Organization	Voegt de in request body meegegeven zorgaanbieder toe
			/Practitioner	Voegt de in request body meegegeven zorgverlener toe
			/PractitionerRole	Voegt de in request body meegegeven rol (WorksAt-relatie) toe
			/Endpoint	(Nog) niet ondersteund
			/Location	(Nog) niet ondersteund
	search	GET	/Organization	Zoekt zorgaanbieder die voldoet aan zoekparameters. Zie paragraaf 4.5 en paragraaf 4.9 voor een beschrijving van de twee manieren van zoeken
			/Practitioner	Zoekt zorgverlener die voldoet aan zoekparameters. Zie paragraaf 4.5 en paragraaf 4.9 voor een beschrijving van de twee manieren van zoeken
			/PractitionerRole	Zoekt rol (WorksAt-relatie) die voldoet aan zoekparameters

			/Endpoint	Zoekt LSP-applicatie die voldoet aan zoekparameters
			/Location	(Nog) niet ondersteund
		POST	/Organization/_search	Zoekt zorgaanbieder die voldoet aan zoekparameters (alternatieve manier van zoeken met HTTP-commando POST)
			/Practitioner/_search	Zoekt zorgverlener die voldoet aan zoekparameters (alternatieve manier van zoeken met HTTP-commando POST)
			/PractitionerRole/_search	Zoekt rol (WorksAt-relatie) die voldoet aan zoekparameters (alternatieve manier van zoeken met HTTP-commando POST)
			/Endpoint/_search	Zoekt LSP-applicatie die voldoet aan zoekparameters (alternatieve manier van zoeken met HTTP-commando POST)
			/Location/_search	(Nog) niet ondersteund
	history	GET	/Organization/_history	(Nog) niet ondersteund
			/Practitioner/_history	(Nog) niet ondersteund
			/PractitionerRole/_history	(Nog) niet ondersteund
			/Endpoint/_history	(Nog) niet ondersteund
			/Location/_history	(Nog) niet ondersteund
System Level	capabilities	GET	/metadata	Geeft Capability Statement
	batch/transaction	POST	/	(Nog) niet ondersteund
	history	GET	/_history	(Nog) niet ondersteund
	search	GET	/	(Nog) niet ondersteund
		POST	/_search	(Nog) niet ondersteund

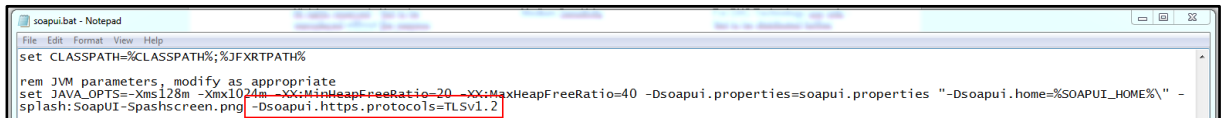
Bijlage 3 Configuratie SoapUI (Native REST Interface)

NB Voor details m.b.t. het verbinden met ZORG-AB zie par. 1.2.

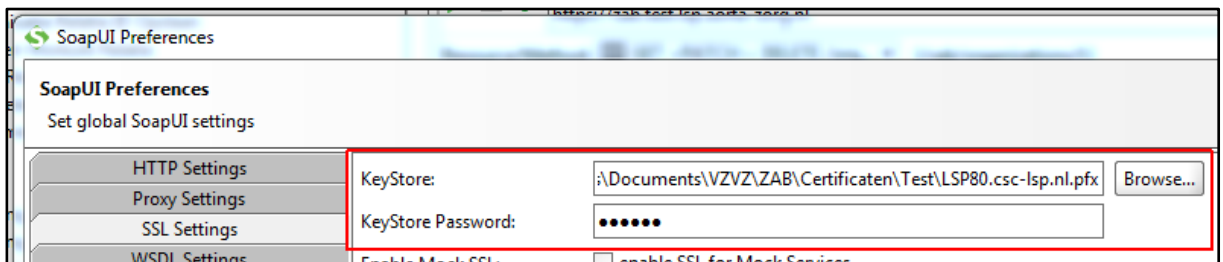
Ter voorbeeld volgt hier een uitleg van hoe de API testtool SoapUI gebruikt kan worden om een request te versturen naar de native REST interface van ZORG-AB:

- 1) Start SoapUI met ondersteuning voor TLS v1.2 door de onderstaande parameter toe te voegen aan de regel beginnende met `set JAVA_OPTS` in `soapui.bat` (deze kan gevonden worden in de bin-directory, bijvoorbeeld `C:\Program Files\SmartBear\SoapUI-5.4.0\bin`) en daarna `soapui.bat` uit te voeren:

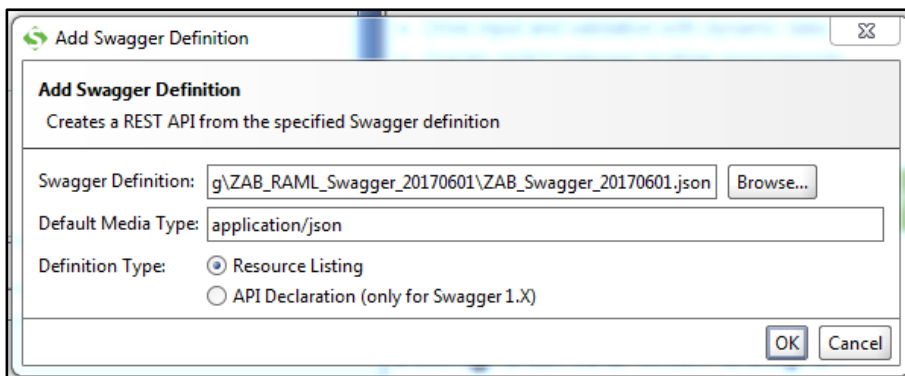
```
-Dsoapui.https.protocols=TLSv1.2
```



- 2) Selecteer het certificaat (PFX-bestand) en vul het wachtwoord in onder `File` → `Preferences` → `SSL Settings`:

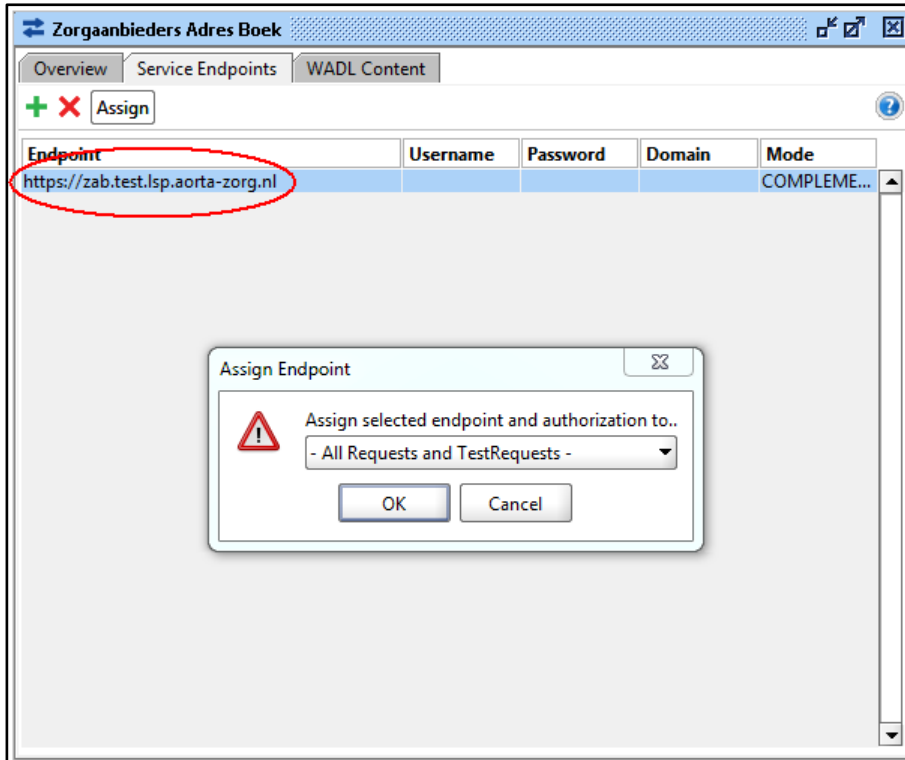


- 3) Laad het Swagger-bestand via `File` → `Create Empty Project` → rechts klikken op project → `Import Swagger`:



(NB Dit vereist dat de Swagger plug-in, `soapui-swagger-plugin-X.X-dist.jar`, is geladen, deze wordt echter standaard meegeleverd met SoapUI en na installatie van SoapUI geplaatst in de directory `C:\Users\[Gebruikersnaam]\.soapuios\plugins` en automatisch geladen.)

- 4) Stel het endpoint in via dubbelklik op `Zorgaanbieders Adres Boek` → `Service Endpoints` → vul het endpoint in van de native REST interface zoals aangegeven in de bovenstaande tabel, zonder de toevoeging `/zab` → `Assign` → `All Requests and TestRequests` → `OK`



- 5) Klik bijvoorbeeld op de resource `/ura/{uraNumber}` → GET → dubbelklik op Request 1 → vul bij de parameter `uraNumber` een URA-nummer in → klik op Submit request (linksboven):

