

Inhoud

1.	INLEIDING	1
2.	API REQUESTS	2
3.	HET DATAMODEL.....	2
3.1	Datamodel relaties met het Snagstream front-end.....	2
3.2	ItemTypes en Categorieën	4
3.2.1	Category.....	5
3.2.2	ItemType.....	5
3.3	Onderwerpen, Vragen en Opdrachten.....	6
3.4	Gebruikers	11
3.5	Bedrijven.....	11
3.6	Projecten	12
3.6.1	Project tekeningen.....	12
3.6.2	Project rondes	13
3.6.3	RondeTypes.....	13
3.6.4	Project antwoorden.....	14
3.7	Antwoorden	21
3.7.1	Soorten antwoorden.....	21
3.8	Voorbeeld vragen.....	22



1. Inleiding

Voor Business Intelligence (BI) dashboards is er een technisch ontwerp in de maak. Een paar klanten willen echter eerder met Power BI aan de slag en zelfstandig dashboards ontwikkelen.

Binnen het technisch ontwerp is een datamodel gemaakt voor een database waarin kopie gegevens uit Snagstream staan. Dit datamodel is ontworpen met het oog op:

- De naamgeving van stamgegevens in Snagstream;
De naamgeving van de entiteiten in de Api sluit niet aan bij de naamgeving van deze in het Snagstream front-end en de Snagstream App. In het datamodel is dit wel het geval.
- Het opvragen van gegevens over beantwoorde vragen en geplaatste snags.
De Snagstream database zelf is gericht op flexibiliteit wat betreft inrichting en het werken de gegevens van een snag of een formulier binnen een project.
Het BI datamodel is gericht op het opvragen van gegevens over beantwoorde vragen al of niet binnen een bepaald project.

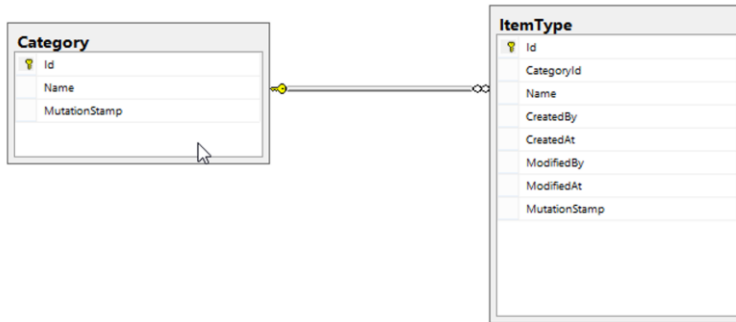
In dit document worden Api calls (end-points) die de huidige versie van Snagstream ondersteund gerelateerd aan het datamodel. Doel van dit document is klanten ondersteunen zodat zij zelfstandig met Api calls een BI database op kunnen zetten en bijvoorbeeld met Power BI dashboards kunnen ontwikkelen.

Category	<p>Deze tabel bevat “vaste” gegevens. Vanuit het front-end kunnen categorieën niet gemuteerd worden. Er zijn 2 categorieën:</p> <ul style="list-style-type: none"> • Snag • Formulier
Item Type	<p>Dit is een Snag of een Formulier. Een verwijzing naar de Category tabel bepaald of het ItemType een Snag of een formulier is.</p> <p>ItemType worden aangemaakt door via de volgende front-end route een nieuw snag of formulier aan te maken:</p> <ul style="list-style-type: none"> • Beheer Sjablonen Snags • Beheer Sjablonen Formulieren
Onderwerp	Een Snag of Formulier is onderverdeeld in Onderwerpen. Onderwerpen worden vanuit het Snag of Formulier scherm aangemaakt.
Vraag	Een onderwerp binnen een Snag of formulier is onderverdeeld in vragen. Vragen worden vanuit het Snag of Formulier scherm aangemaakt.
Opdracht	Een vraag binnen een Snag of formulier is onderverdeeld in opdrachten. Opdrachten worden vanuit het Snag of Formulier scherm aangemaakt.
Opdracht Optie	<p>Aan opdrachten van het volgende type kan een lijst van opties gekoppeld worden:</p> <ul style="list-style-type: none"> • Selectie • Knoppen • Actie • Meerkeuze • Prioriteit <p>Opdrachtopties worden vanuit het Snag of Formulier scherm aangemaakt.</p>
Opdracht Type	<p>Het Opdracht type bepaald de “soort van het antwoord” (medewerker keuze, Status keuze, etc.).</p> <p>Opdrachttypes kunnen niet vanuit het front-end gemuteerd worden. Er is een vaste lijst van Opdrachttypes.</p>
Opdracht Type Optie	<p>Aan de volgende opdracht-typen is een vast lijst van opties gekoppeld</p> <ul style="list-style-type: none"> • Akkoord • Snag prioriteit (prioriteit standaard) • Status <p>Deze opties kunnen niet vanuit het front-end gemuteerd worden. Er is per type een vaste lijst van opties.</p> <p>Aan het volgende OpdrachtType kan vanuit Beheer Type/Labels een lijst van Types en gerelateerde Labels ingevoerd worden.</p> <ul style="list-style-type: none"> • Type/Tag <p>To do BEPALEN WAAR DEZE IN HET MODEL KOMEN</p>

	Types en Tags worden in de ItemModuleTypeOption tabel weg geschreven. Types hebben geen parentId en Tags/Labels wel en deze verwijst naar het Type.
LabelType	Label types worden aangemaakt en gemuteerd vanuit het front-end path: Beheer Type / Label
Label	Label types worden aangemaakt en gemuteerd vanuit het label type scherm Een LabelType en daarbinnen een Label kan gekozen worden vanuit de een opdracht van het OpdrachtType Type/Label.
Gebruiker	Gebruikers worden aangemaakt en gemuteerd vanuit het front-end path: Beheer Gebruikers
Bedrijf	Bedrijven worden aangemaakt en gemuteerd vanuit het Gebruiker scherm.
Ronde Type	Ronde types worden aangemaakt en gemuteerd vanuit het front-end path: Beheer Rondetypes
Project	Projecten worden aangemaakt en gemuteerd vanuit het front-end path: Beheer Projecten
Ronde	Rondes worden aangemaakt vanuit en gemuteerd het front-end path: <ul style="list-style-type: none"> Project Rondes
Tekening	Tekeningen worden aangemaakt en gemuteerd vanuit het front-end path: <ul style="list-style-type: none"> Project Tekeningen
Ronde Tekening	Een tekening kan vanuit het Ronde scherm gekoppeld worden aan een ronde
Rondelitem	Een ronde item is een snag of een formulier gekoppeld aan een ronde.
Antwoord	Een Rondeitem bestaat uit een lijst van vragen. Tijdens het beantwoorden van deze vragen worden de antwoorden opgeslagen. Een antwoord kan een vrije tekst zijn, een datum, een of meerdere gekozen medewerkers, een keuze uit een lijst van mogelijke antwoorden of een foto. Vragen worden beantwoordt vanuit een Snag of Formulier gekoppeld aan een project.
LinkedRondelitem	Aan een antwoord kan vanuit het formulier scherm een Snag gekoppeld worden. Dit resulteert in een LinkedRondelitem.

3.2 ItemTypes en Categorieën

Binnen Snagstream kan gewerkt worden met Snags en Formulieren. Wat betreft datamodel zijn beide gelijk alleen verwijst een snag naar een “snag” Category en een formulier naar een “survey” Category. De volgende figuur toont het datamodel voor Category en ItemType:



3.2.1 Category

Categorieën kunnen opgevraagd worden met Het volgende end-point:

<GET /api/item/categories/all>

Deze call geeft een lijst van categorieën met per Category de volgende eigenschappen (niet alle eigenschappen zijn weergegeven):

```
{
  "Id": "458ac9fe-f77c-4cc1-b47f-d5f857413ac1",
  "Name": "snagCategory"
}
```

3.2.2 ItemType

Een lijst van alle ItemTypes kan opgevraagd worden met Het volgende end-point:

<POST api/item/categoriesandtypes/all>

Deze call geeft een lijst terug met daarin per category de volgende gegevens:

```
{
  "Category": { zie de api/item/categories/all call },
  "Types": [ { type } ]
}
```

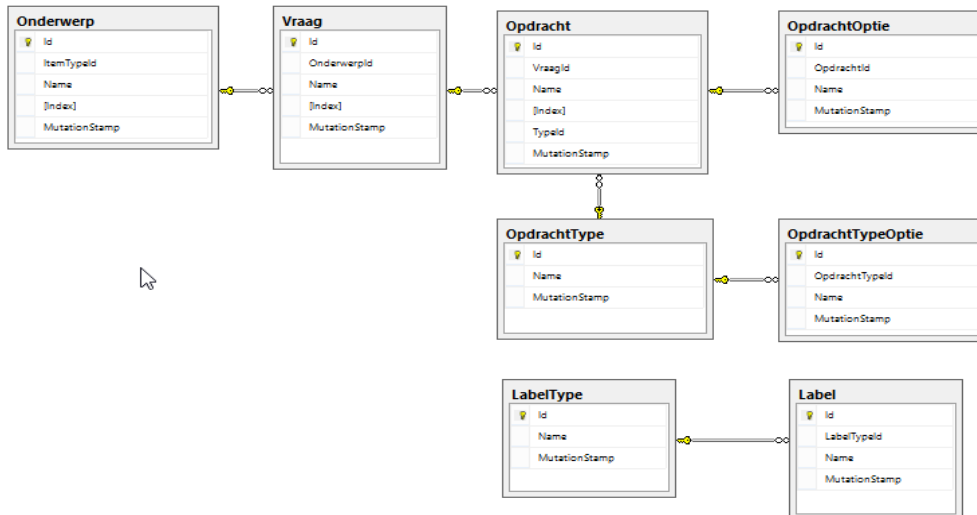
De types gekoppeld aan een category worden als een array van types terug gegeven. De gegevens van een type zien er als volgt uit (niet alle eigenschappen zijn weergegeven):

```
{
  "Type": {
    "Id": "b02fde01-d9e2-46c3-a08f-c031b3843d0f",
    "ItemCategoryId": "458ac9fe-f77c-4cc1-b47f-d5f857413ac1",
    "Name": "Keuringsnag",
    "CreatedAt": "2018-10-17T11:15:42.461649",
    "CreatedBy": "5ce18cc0-d26c-470d-b489-a9ea70df5529",
    "ModifiedAt": "2020-04-22T09:06:52.281987",
    "ModifiedBy": "5ce18cc0-d26c-470d-b489-a9ea70df5529",
    "Versions": [ {
      "Id": "1019a8ea-d459-443e-9441-67f52cc45a0a",
      "CreatedAt": "2020-04-22T09:06:52.382271",
      "CreatedBy": "5ce18cc0-d26c-470d-b489-a9ea70df5529"
    } ]
  },
  "Versions": [ { zelfde lijst als hierboven } ]
}
```

}

3.3 Onderwerpen, Vragen en Opdrachten

De volgende figuur geeft de Onderwerpen, Vragen en Opdrachten tabellen weer met daaraan gerelateerde tabellen:



Een ItemType (Snag of Formulier) is opgedeeld in Onderwerpen, Vragen en Opdrachten. De volgende call geeft de details gegevens van een ItemType:

<POST api/items/itemDictionary/latest/{itemTypeld}>

In de body van deze call wordt een filter ingesteld:

```
{
  "itemVersionIds": [null]
}
```

Door versionId's mee te geven kun je de details van een specifieke versie van het ItemType ophalen. Wordt er niet op versie Id's gefilterd dan krijg je de laatste versie van het ItemType terug.

N.B. Hoewel de naam anders doet vermoeden bevat de response informatie van alle versies van het item als er geen ItemVersionIds in de body worden meegestuurd.

Deze call geeft een vrij complexe structuur terug met daarin vele gegevens dubbel. Voor het datamodel is alleen de lijst van Versions belangrijk. De complete structuur ziet er als volgt uit:

```
{
  "Conditions": [],
  "FieldConditions": [],
  "Fields": [],
  "ModuleOptions": [],
  "Modules": [],
  "ModuleTypes": []
  "Sections": [],
  "Types": [],
  "Versions": []
}
```

```
}
```

De “Versions” eigenschap is een array/lijst “Version” informatie met per Version de volgende gegevens:

```
{
  "Id": "c95be59a-b284-44e3-a4be-4d34242726ca",
  "Sections": []
}
```

De gegevens van een Version komen niet terug in het datamodel. De lijst van Sections bevat per Section de volgende informatie:

```
{
  "Id": "150f7107-7261-4042-a816-853aa9b9b1ac",
  "Index": 1,
  "Name": "Algemeen",
  "Fields": []
}
```

Een Section is een Onderwerp in het BI datamodel. De lijst van Fields bevat per Field de volgende informatie:

```
{
  "Id": "8f904646-801b-481a-a70c-5e4c5bcf2aba",
  "Index": 1,
  "Name": "Omschrijving",
  "Modules": []
}
```

Een Field is een Vraag in het BI datamodel. De lijst van Modules bevat per Module de volgende informatie:

```
{
  "Id": "dc379f47-9cee-4e33-b64d-58e2c04cf6cd",
  "ItemModuleTypeId": "be62d7d5-d5ec-41e0-a57a-747a2b63f4d7",
  "Name": "Omschrijving",
  "Index": 1,
  "ModuleOptions": []
}
```

Een Module is een Opdracht in het BI datamodel. De ItemModuleTypeId eigenschap verwijst naar een entry in de OpdrachtType tabel.

De ModuleOptions eigenschap kan een array van ModuleOptions bevatten maar is leeg als er een specifieke versie van een ItemType wordt opgevraagd. Is het array gevuld dan bevat deze per entry de volgende gegevens:

```
{
  "Id": "21b1f51d-5623-419a-90d6-b1fcf78a24ab",
  "ParentId": null,
  "Name": "Afgerond",
  "ItemModuleTypeId": "7bf16585-70c0-449c-8e28-b6cbef01fb85",
}
```

Een ModuleOption is een OpdrachtOptie in het BI datamodel.

Het parentId van een entry is alleen gevuld als de optie een Type/Label optie is. Het parentId verwijst dan naar een LabelType en het Id in de entry zelf is het Id van het Label binnen de LabelType. Een voorbeeld hiervan is:

```
{
  "Id": "c2d2658d-8b6c-49fe-bcc1-6de1e52b423b",
  "ParentId": "3bc4a930-1795-4ea6-986b-341c28a7ddd7",
  "Name": "tester de test",
  "ItemModuleTypeId": "0a8d3464-c5dd-4d3a-a1d3-a538225edfd5",
}
```

De gegevens uit het ModleOptions array worden overgenomen in de OpdrachtOptie tabel van het datamodel.

3.3.1.1 OpdrachtTypes

De volgende tabel bevat de Id's van de OpdrachtType waarmee binnen Snagstream gewerkt kan worden.

OpdrachtTypeId	Opdracht type
ab095e22-f97c-4c88-bb4f-cf98675065ed	Meerkeuze
91c15760-001f-47f1-a75f-3e02ac003a6d	Selectie
be62d7d5-d5ec-41e0-a57a-747a2b63f4d7	Vrij veld
4f225325-1e2d-4662-8e6d-fa073bad94a2	Akkoord
837a73eb-f637-4b14-b2eb-ca082a06168f	Actie
c4df9b2c-d6bb-42fc-a1a8-9caf4ea56556	Gebruikers
bfd945df-dab8-4dd0-ae60-6344304a1666	Prioriteit standaard
ee2ba447-96b4-44ce-aae4-605083a525cf	Prioriteit uniek
ee5c7106-a6f7-46c0-8834-8f9f1f8a8536	Datum
e8b26278-642e-4f4f-aff9-b97b1cdc2fcf	Getal
2481d661-7027-4e0e-a106-98cb2de75d58	Knoppen
2132e34a-cbae-4a79-88b4-aa125a0fee98	Foto
84e0d26c-3808-488e-aab2-72651f1d2bcf	Handtekening
0a8d3464-c5dd-4d3a-a1d3-a538225edfd5	Type / Tag
7bf16585-70c0-449c-8e28-b6cbef01fb85	Status
c5d6cc34-4926-462c-9fc6-54bbf0abcef8	CC
60619ca6-054d-4e0f-a2f2-e3b78f7a1e89	Toegewezen aan
42efbebf-0397-4652-90c8-4b4de6647113	Beschrijving

Er is geen end-point waarmee de gegevens uit deze tabel opgehaald kunnen worden.

Afhankelijk van het OpdrachtType kan er een lijst van options aan een Opdracht gekoppeld zijn.

3.3.1.2 Standaard Opdracht types met opties

Voor de standaard Snagstream OpdrachtTypes zijn er specifieke end-points om de opties op te halen:

- Akkoord
<GET api/items/itemModuleTypeOptions/akkoord>
- Priority
<GET api/items/itemModuleTypeOptions/priority>

- Status
<GET api/items/itemModuleTypeOptions/status>

Voor de standaard Snagstream OpdrachtTypes kunnen de options ook opgehaald worden met dit endpoint

<GET api/items/itemModuleTypeOptions/{itemModuleTypeId}>

De response van deze end-points is een array van opties met per optie de volgende gegevens:

```
{
  "Id": "8eeef73e-f3e3-4ccc-a79c-e41a526cbaa4",
  "ItemModuleTypeId": "4f225325-1e2d-4662-8e6d-fa073bad94a2",
  "Name": "Ja",
}
```

Deze gegevens kunnen verwerkt worden in de OpdrachtTypeOptie tabel van het data model.

3.3.1.3 Type / Labels Opdracht type

Alle Types en labels kunnen opgehaald worden met het volgende end-point

[GET api/tag/all](#)

De response is een array van LabelTypes met per entry de volgende gegevens:

```
{
  "TagId": "a86d7218-0cf0-4bb8-be8a-40650a2e5fae",
  "Name": "Afbouwtimmerwerk",
  "Children": []
}
```

Deze gegevens kunnen verwerkt worden in de LabelType tabel van het datamodel. Het Children array bevat per entry de volgende gegevens:

```
{
  "TagId": "b20ddca5-f8b7-4b3d-bff6-27953ae8a088",
  "ParentTagId": "a86d7218-0cf0-4bb8-be8a-40650a2e5fae",
  "Name": "Aftimmering ontbreekt"
}
```

Deze gegevens kunnen verwerkt worden in de Label tabel van het datamodel.

3.3.1.4 Opdracht types waarbij een lijst van opties ingevoerd kan worden

Bij de Opdrachten gekoppeld aan een van de volgende OpdrachtTypes kan een lijst van opties vastgelegd worden:

- Selectie
- Knoppen
- Actie
- Meerkeuze
- Prioriteit

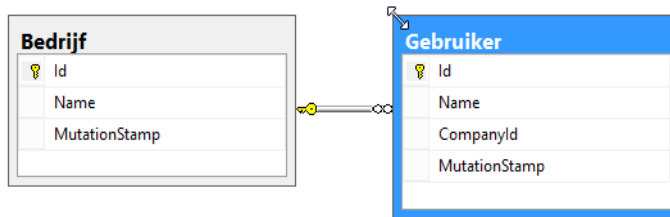
De opties gekoppeld aan een Opdracht opties kunnen vanuit de response op een

[POST api/items/itemDictionary/latest/{itemTypeld}](#)

aanroep aangemaakt worden. Zie de hoofdparagraaf van deze sub-paragraaf.

3.4 Gebruikers

De volgende figuur toont de gebruikers tabel en de direct daaraan gerelateerde tabellen:



Het volgende end-point geeft een lijst van gebruikers:

[POST /api/users/allwithdeleted](#)

Met in de body:

```
{
  "staticCriteria": [],
  "specificStaticCriteria": []
}
```

Het resultaat is een array van users met per user de volgende gegevens

```
{
  "UserId": "",
  "CompanyId": "",
  "FirstName": "",
  "SurName": "",
  "FullName": ""
}
```

3.5 Bedrijven

Het volgende end-point geeft een lijst van bedrijven:

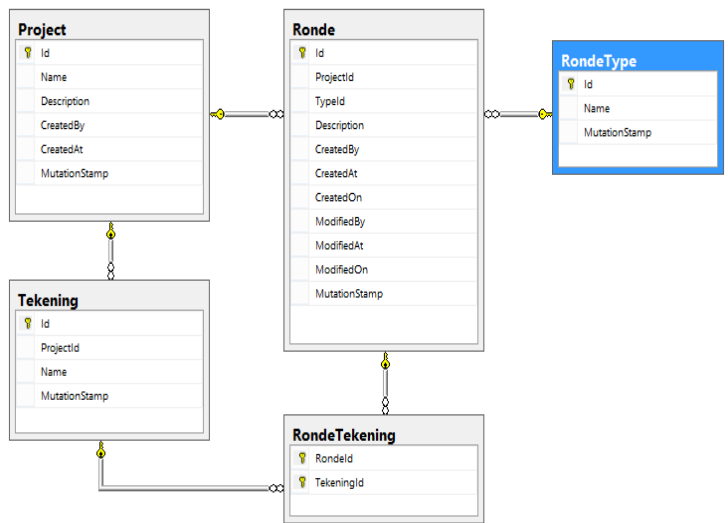
[GET /api/company/all](#)

Het resultaat is een array van companies met per company de volgende gegevens

```
{
  "CompanyId": "",
  "Name": ""
}
```

3.6 Projecten

De volgende figuur toont de projecten tabel en de directe relaties met deze tabel:



Het volgende end-point geeft een lijst van projecten:

[GET api/projects/all](#)

Het resultaat is een array van projecten met per project de volgende gegevens

```

{
  "ProjectId": "",
  "Name": "",
  "Description": ""
  "CreatedAt": "date time"
  "CreatedBy": "id van een gebruiker"
}
    
```

3.6.1 Project tekeningen

Het volgende end-point geeft een lijst van tekeningen gekoppeld aan een project:

[POST api/projects/{projectId}/drawings/all](#)

Het resultaat is een lijst van tekeningen met per tekening:

```

{
  "DrawingId": "",
  "ProjectId": "verwijzing naar de project tabel"
  "Name": ""
}
    
```

3.6.2 Project rondes

Het volgende end-point geeft een lijst van rondes gekoppeld aan een project:

[api/visit/{projectId}/all](#)

Geeft een lijst met de rondes van het project met per ronde:

```
{
  "Visid": "",
  "ProjectId": " verwijst naar de Project tabel",
  "Description": "",
  "VisitTypeId": " verwijst naar de RondeType tabel"
  "CreatedAt": "date time",
  "CreatedBy": "verwijst naar de gebruikers tabel",
  "CreatedOn": "Verwijst naar ?? To do ",
  "ModifiedAt": "date time",
  "ModifiedBy": "verwijst naar de gebruikers tabel",
  "ModifiedOn": "Verwijst naar ?? To do ",
  "Drawings": [ lijst van drawings ],
}
```

De lijst van drawings bevat de tekeningen gekoppeld aan de ronde. Van de detailgegevens per drawing uit de lijst is voor het datamodel alleen de volgende eigenschap van belang:

```
{
  "DrawingId": "verwijst naar de tekeningen tabel"
}
```

3.6.3 RondeTypes

Het volgende end-point geeft een lijst van RondeTypes:

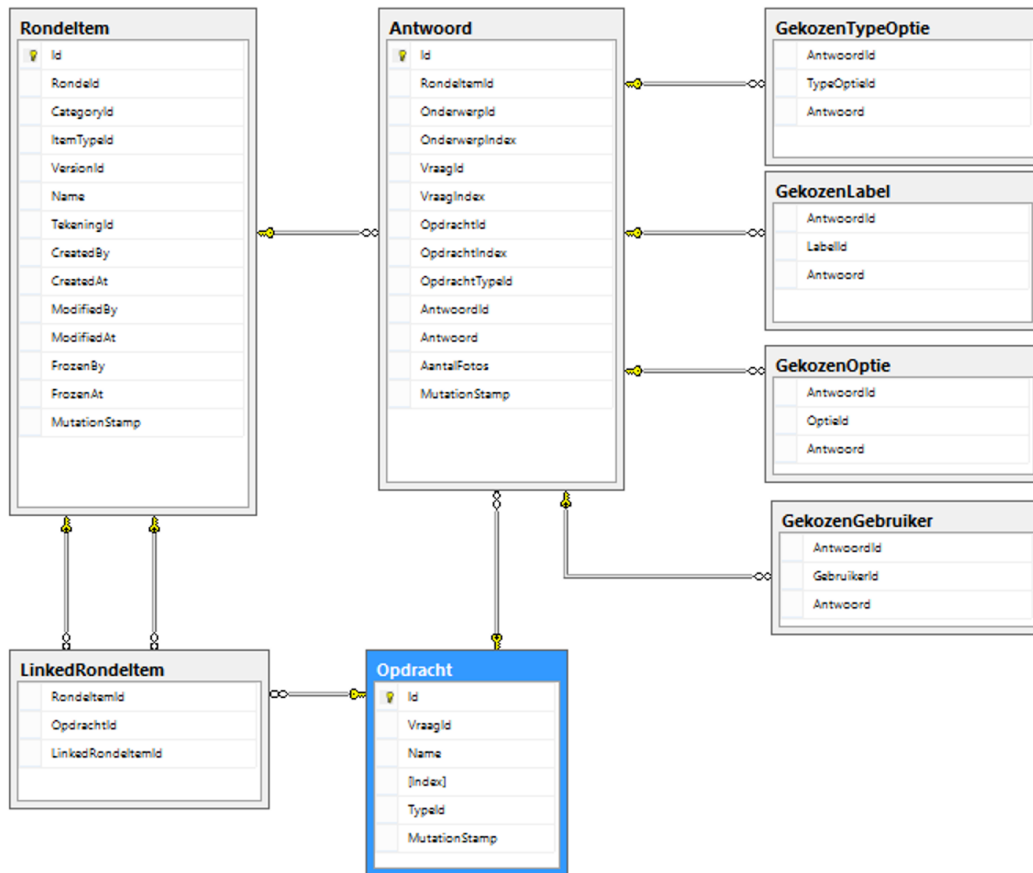
[GET api/visit/types](#)

Het resultaat is een array van visit types per type de volgende gegevens

```
{
  "Id": "",
  "Name": ""
}
```

3.6.4 Project antwoorden

De volgende figuur geeft de tabellen betrokken bij de antwoordgegevens:



Het ophalen van antwoorden moet in 2 stappen uitgevoerd worden:

1. De Id's ophalen van de snags of formulieren gekoppeld aan het project
2. De antwoorden ophalen van de snags of formulieren.

3.6.4.1 Project snags of formulieren opvragen

Het volgende end-point

<POST api/item/{projectId}/data/lazy/{categoryId}/{itemTypeld}/{itemVersionId}>

Geeft de snags of formulieren van een project.

In de body van de aanroep worden de pagina en het aantal regels per pagina voor de response opgegeven:

```

{
    "offset":0,
    "increment":100
}
    
```

In bovenstaande voorbeeld wordt de eerste pagina opgevraagd met daarin 100 regels/items.

De parameters in het path van de aanroep bepalen wat je opvraagt:

categoryId	Soort
458ac9fe-f77c-4cc1-b47f-d5f857413ac1	Snags
ccb72a59-032d-4867-a2ea-d8ab830b54a1	Formulieren

itemTypeId	Soort
00000000-0000-0000-0000-000000000000	Alles snags of formulieren gekoppeld aan het project. Bevat geen antwoordgegevens
Specifieke guid	Een specifiek ItemType (snag of formulier). Bevat wel antwoordgegevens

itemVersionId	Soort
00000000-0000-0000-0000-000000000000	Alles versies van het ItemType
Specifieke guid	Een specifieke versie van het ItemType

De snags gekoppeld aan een project worden opgevraagd met de volgende call:

<POST api/item/{projectId}/data/lazy/458ac9fe-f77c-4cc1-b47f-d5f857413ac1/00000000-0000-0000-0000-000000000000/00000000-0000-0000-0000-000000000000>

met in de body de pagineringsparameters.

Dit geeft een lijst van gekoppelde snags. Deze lijst wordt voorafgegaan door informatie over het pagineren en filters. De response ziet er globaal gezien als volgt uit

```
{
  "Offset": 0,
  "Increment": 100,
  "SortOnColumn": "00000000-0000-0000-0000-000000000000",
  ....
  "Items": []
}
```

Het Items array bevat de lijst van gekoppelde Snag / Formulieren (afhankelijk van het categoryId). Deze lijst bevat per entry naast allerlei detail informatie: het Id van de snag/ het formulier binnen het project en het ItemTypeId van het item:

```
{
  ...
  "Id": "03167957-8263-40d8-b2fb-8d14bac75db8",
  "ItemTypeId": "c8f28c00-2c73-4502-925f-fde097d85eee",
  ...
}
```

De gegevens per Item bevatten geen antwoord informatie. Deze informatie kan op twee manieren opgehaald worden.

3.6.4.2 Snags of formulieren antwoorden per snag of formulier

Per Item entry uit het resultaat van de aanroep uit de vorige paragraaf kan dit end-point aangeroepen worden

<POST api/item/{projectId}/data/lazy/458ac9fe-f77c-4cc1-b47f-d5f857413ac1/{itemTypeId}/00000000-0000-0000-0000-000000000000>

met in de body de pagineringsparameters.

Door In bovenstaand end-point een ItemTypeId in te vullen (c8f28c00-2c73-4502-925f-fde097d85eee in voorgaande voorbeeld)) worden de antwoord gegevens van de betreffende snag of formulier opgehaald.

Het resultaat van deze aanroep bevat de volgende gegevens

```
{
  "Offset": 0,
  "Increment": 100,
  "SortOnColumn": "00000000-0000-0000-0000-000000000000",
  ....
  "Items": []
}
```

met per Item:

```
{
  "Id": "03167957-8263-40d8-b2fb-8d14bac75db8",
  "VisitId": "150962d0-e6b5-4841-b9fd-0cff6e52ae51",
  "ItemCategoryId": "458ac9fe-f77c-4cc1-b47f-d5f857413ac1",
  "ItemTypeId": "c8f28c00-2c73-4502-925f-fde097d85eee",
  "ItemTypeName": "Snaggy",
  "ItemVersionId": "d3a389fc-29ef-4478-8b3a-2a19632d8260",
  "DrawingId": null,
  "CreatedAt": "2020-04-30T13:36:56.086857",
  "CreatedBy": "5ce18cc0-d26c-470d-b489-a9ea70df5529",
  "CreatedOn": "00000000-0000-0000-0000-000000000000",
  "ModifiedAt": "2020-04-30T13:37:16.117427",
  "ModifiedBy": "5ce18cc0-d26c-470d-b489-a9ea70df5529",
  "FrozenAt": null,
  "FrozenBy": null,
  "ModuleInstanceDictionary": {}
}
```

Deze gegevens worden verwerkt in de RondelItem tabel:

Reponse eigenschap	RondelItem kolom
Item.Id	Id
Item.VisitId	Rondeld
Item. ItemCategoryId	CategoryId
Item. ItemTypeId	ItemTypeId
Item. ItemTypeName	Name
Item. ItemVersionId	VersionId
Item. DrawingId	TekeningId
Item.CreatedAt	CreatedBy
Item.CreatedBy	CreatedAt
Item.ModifiedBy	ModifiedBy
Item.ModifiedAt	ModifiedAt
Item.FrozenAt	FrozenBy
Item.FrozenBy	FronzenAt

De ModuleInstanceDictionary eigenschap van een item bevat een lijst van ModuleInstance objecten. Deze objecten bevatten de antwoorden. Bij elk object hoort dan ook een regel uit de antwoord tabel. Dit geldt alleen voor objecten waarvan de Value eigenschap niet leeg is. Alleen dan is er een antwoord ingevoerd bij de opdracht in Snagstream.

Elk object bevat de volgende gegevens:

```

"ItemModuleVild" : {
  "Id": "e0c33a13-feac-4a1e-9337-1a437764733f",
  "ItemModuleTypeid": "7bf16585-70c0-449c-8e28-b6cbef01fb85",
  "ItemModuleid": "51545010-b870-46eb-826a-cf60a8a83ec3",
  "Value": "aabc7c19-366d-4944-b66e-20ac746fd33c",
  "StringValue": "Feedback gewenst",
  "Media": []
}

```

De gegevens van de response worden verwerkt in de Antwoord tabel en afhankelijk van het ItemModuleTypeid in een van "Gekozen" tabellen.

De volgende tabel geeft de relatie weer tussen de kolommen uit de Antwoord tabel en de response gegevens:

Reponse eigenschap	Antwoord kolom
Item.ModuleInstance.Id	Id
	OnderwerpId
	OnderwerpIndex
	VraagId
	VraagIndex
Item. ModuleInstance. ItemModuleid	OpdrachtId
	OpdrachtIndex
ModuleInstance.ItemModuleTypeid	OpdrachtTypeid
Item.ModuleInstance.StringValue	Antwoord
Aantal entries in het Item.ModuleInstance.Media array	AantalFotos

Zoals wellicht is opgevallen zijn de kolommen OnderwerpId, OnderwerpIndex, VraagId, VraagIndex en OpdrachtIndex leeg in bovenstaande tabel. Deze kunnen ingevuld worden door aan de database de opdracht en daaraan gekoppeld de Vraag en het Onderwerp op te vragen. Hierin wordt dan gefilterd op het OpdrachtId.

Afhankelijk van het ItemModuleTypeid wordt de response eigenschap "Value" verwerkt in een van de "Gekozen" tabellen. Dit wordt alleen gedaan voor de types waaraan een lijst van opties gekoppeld kan zijn. Zie de betreffende paragraaf.

OpdrachtType	Gekozen tabel
Selectie	GekozenTypeOptie
Meerkeuze	GekozenTypeOptie
Akkoord	GekozenOptie
Actie	GekozenTypeOptie
Gebruikerslijst	GekozenGebruiker
Prioriteit	GekozenOptie
Picker	GekozenTypeOptie
Switch	GekozenTypeOptie
Type / Tag	GekozenLabel
Snag prioriteit	GekozenTypeOptie
Status	GekozenOptie
CC	GekozenGebruiker
Toegewezen aan	GekozenGebruiker

De Value eigenschap kan een lijst van Id's bevatten. Voor elk van deze Id's wordt dan een entry in de betreffende Gekozen tabel gemaakt. De antwoord kolom van deze tabel kan dan bepaald worden door de "Name" eigenschap uit de Optie tabel waaraan de Gekozen tabel gekoppeld is te lezen. Een alternatief is om het antwoord uit de ValueString te lezen maar ik weet niet of de inhoud van de Value eigenschap en de ValueString eigenschap in dezelfde volgorde staat.

3.6.4.3 Snags of formulieren antwoorden voor een reeks van items

De response van het end-point uit de voorgaande paragraaf bevat geen gegevens over de structuur van de snag of het formulier. Het end-point uit deze paragraaf bevat deze gegevens wel.

Met het volgende end-point kunnen de antwoord gegevens van een reeks van snags of formulieren opgevraagd worden:

[POST /api/items/{projectId}/full](#)

Met in de body

```
{
  "itemIds": [ ],
  "loadDefinition": true
}
```

In de body kan een array van Item Id's meegestuurd worden. Door hierin de Item Id's uit de response beschreven in de paragraaf "Project snags of formulieren opvragen" te verwerken worden alle antwoorden van snags of formulieren gekoppeld aan een project opgevraagd.

Met loadDefinition wordt bepaald of er in de response van het end-point informatie zit over de structuur van de snags of formulieren achter de vragen. Met "loadDefinition": true bevat de response deze structuur. De response ziet er als volgt uit:

```
[ { item gegevens } ]
```

Met per item:

```
{
  "Id": "03167957-8263-40d8-b2fb-8d14bac75db8",
  "ItemType": "c8f28c00-2c73-4502-925f-fde097d85eee",
  "ItemTypeName": "Snaggy",
  "ItemCategoryId": "458ac9fe-f77c-4cc1-b47f-d5f857413ac1",
  "ItemVersionId": "d3a389fc-29ef-4478-8b3a-2a19632d8260",
  "VisitId": "150962d0-e6b5-4841-b9fd-0cff6e52ae51",
  "DrawingId": null,
  "CreatedAt": "2020-04-30T13:36:56.086857",
  "CreatedBy": "5ce18cc0-d26c-470d-b489-a9ea70df5529",
  "CreatedOn": "00000000-0000-0000-0000-000000000000",
  "ModifiedAt": "2020-04-30T13:37:16.117427",
  "ModifiedBy": "5ce18cc0-d26c-470d-b489-a9ea70df5529",
  "FrozenAt": null,
  "FrozenBy": null,
  "ModuleInstanceDictionary": {},
  "ItemSectionInstanceDefinitions": []
}
```

Op de "ItemSectionInstanceDefinitions" eigenschap na is de inhoud van dit response gelijk aan die uit de voorgaande paragraaf. Zie voor het verwerken van de response in de Rondeltemen en Antwoorden de voorgaande paragraaf.

N.B. In de response van dit end-point verwijst het Id van een ModuleInstance objecten naar het Id van een ModuleInstance. In de response van het end-point uit de vorige paragraaf verwijst dit Id naar een itemModuleViid.

Het ItemSectionInstanceDefinitions array bevat per entry de volgende gegevens (deze structuur is soort gelijk aan de structuur beschreven in de "3.33.3" paragraaf) :

```
{
  "Id": "938b424d-0a85-4254-a3e6-f5cf1d41a81e",
  "Index": 1,
  "ItemSectionId": "6a82f9f8-de1b-4abd-8ae3-09e83e5abf41",
  "Name": "Test onderwerp 1",
  "Fields": []
}
```

Een Field is een Vraag in het datamodel. Het Fields array bevat per entry de volgende gegevens:

```
{
  "Id": "bf8bb3ae-22df-4ce4-bd48-e41e396fad31",
  "Index": 1,
  "Name": "Test vraag",
  "Modules": []
}
```

Een Module is een Opdracht in het datamodel. Het Modules array bevat per entry de volgende gegevens:

```
{
  "Id": "499b0e50-accd-4547-974d-1ed890a05a68",
  "ItemModuleTypeId": "7bf16585-70c0-449c-8e28-b6cbef01fb85",
  "ItemModuleId": "51545010-b870-46eb-826a-cf60a8a83ec3",
  "Index": 1,
  "Name": "Test antwoord",
  "ModuleOptions": []
}
```

Een ModuleOption komt afhankelijk van het ItemModuleTypeId overeen met een van de Optie tabellen of de Label tabel uit het datamodel.

Het ModuleOptions array bevat per entry de volgende gegevens:

```
{
  "Id": "89f5fc6e-a4fe-46fd-9a6e-cf816ee2b757",
  "ParentId": null,
  "Index": 6,
  "Name": "Ter goedkeuring"
}
```

3.6.4.4 Antwoord kolommen

OpdrachtIndex

De kolom OpdrachtIndex kan bepaald worden door in het itemSectionInstanceDefinition array te zoeken naar de ModuleOptions met als Id de waarde van de ItemModuleId eigenschap van de Item.ItemModuleInstance.

VraagId en VraagIndex

De kolommen VraagId en VraagIndex staan dan in het Field object met het Module in de modules array.

Onderwerp en OnderwerpIndex

De kolommen OnderwerpId en OnderwerpIndex staan vervolgens weer in de entry van het itemSectionInstanceDefinition array waarvan het Field object in het Fields array staat.

3.7 Antwoorden

Centraal in het datamodel staan de Rondeltem en “Antwoord” tabellen. Deze tabellen bevat alle gegevens over de antwoorden die binnen project-rondes zijn gegeven op de vragen uit Snags en Formulieren. Alle andere tabellen kunnen beschouwt worden als stamgegevens tabellen.

Net zoals alle Id's is de Rondeltem.Id kolom uniek binnen de database. Dit Id wordt gegenereerd als er binnen een project een Snag of Formulier wordt gekoppeld aan een Ronde of aan een tekening. Dit is dus niet het Id van de snag of het formulier zelf. Het Id van de Snag of het Formulier staat in de kolom ItemTypeId. De naam van het ItemType is overgenomen uit de response van het gerbuikte end-point en deze naam staat ook in de Name kolom van de ItemType tabel.

De Rondeld kolom spreekt voor zich en verwijst naar de Ronde tabel waarin de naam van de Ronde staat waarbinnen de Snag is geplaatst of het antwoord is ingevuld.

3.7.1 Soorten antwoorden

Een antwoord kan bestaan uit een vrije tekst, een datum, een keuze uit een aantal opties, het kiezen van medewerker(s) of het maken van foto's. De kolommen van de Antwoord tabel die horen bij een bepaald type antwoord zijn:

Antwoord type	Betrokken kolommen in de antwoord tabel
Vrijetekst	Antwoord
Datum	Antwoord
Optie keuze	De antwoord kolom bevat de tekst van de optie en de AntwoordId kolom verwijst naar een regel uit de OpdrachtTypeOption tabel
Medewerker keuze	De antwoord kolom bevat de naam van de medewerker en de AntwoordId kolom verwijst naar een regel uit de User tabel
Foto	Aantal foto's

3.7.1.1 Positie van een antwoord binnen een snag of formulier

Elke formulier of snag binnen Snagstream is ingedeeld in een hiërarchie van Onderwerpen, Vragen en Opdrachten. De kolommen OnderwerpIndex, VraagIndex en OpdrachtIndex verwijzen naar deze hiërarchie. De waarden van deze indices horend bij de volgende figuur zijn dus : OnderwerpIndex = 2,

VraagIndex = 2 en OpdrachtIndex = 1.

The screenshot shows a web application interface with a breadcrumb trail: 1 0 Casco > 2 1 Binnenpakket woningen > 2.1 10 Asbest saneren in woningen > 2.2 Asbestsanering, woningen conform raming BK Ingenieurs B.V. mail 22-1... > Uitvoeren. The 'Uitvoeren' dialog box has three radio buttons: 'Ja', 'Nee', and 'N.v.t.', each with a close button (x). Below them is a text input field 'Optie toevoegen...'. To the right is a 'Knoppen' button with a dropdown arrow. At the bottom of the dialog are three checkboxes: 'Verplicht', 'Opent eigen browser', and 'Indicatief'. A link 'Ontkoppel versiebeheer' is located at the bottom right of the dialog.

3.8 Voorbeeld vragen

Deze paragraaf bevat voorbeeld SQL statements voor het beantwoorden van bepaalde vragen vanuit het datamodel.

Aantal Snags gekoppeld aan een project

```
SELECT count(ri.id)
FROM Ronde v
JOIN Rondeltem ri ON ri.RondelId = v.Id
WHERE v.ProjectId = '2501A64A-2B6F-4B65-B130-DE0244E5DBA7'
AND ri.CategoryId = '458ac9fe-f77c-4cc1-b47f-d5f857413ac1'
```

Aantal Snags binnen een project waarin bepaald type vragen zijn beantwoord

```
SELECT count(ri.id)
FROM Ronde v
JOIN Rondeltem ri ON ri.RondelId = v.Id
JOIN Antwoord a on a.RondeltemId = ri.Id
WHERE v.ProjectId = '2501A64A-2B6F-4B65-B130-DE0244E5DBA7'
AND ri.CategoryId = '458ac9fe-f77c-4cc1-b47f-d5f857413ac1'
AND a.OpdrachtTypeId IN ('837a73eb-f637-4b14-b2eb-ca082a06168f', 'ee2ba447-96b4-44ce-aae4-605083a525cf', '4f225325-1e2d-4662-8e6d-fa073bad94a2')
```

In dit voorbeeld zijn de Ids opgenomen van de OpdrachtTypes:

- Actie
- Prioriteit
- Akkoord

Zie ook de tabel in de "OpdrachtTypes" paragraaf.

Aantal Snags binnen een project waarin een bepaalde medewerker is gekoppeld aan een bepaalde vraag

```
SELECT count(ri.id)
FROM Ronde v
join Rondeltem ri ON ri.Rondeld = v.Id
join Antwoord a on a.RondeltemId = ri.Id
WHERE v.ProjectId = '2501A64A-2B6F-4B65-B130-DE0244E5DBA7'
AND ri.CategoryId = '458ac9fe-f77c-4cc1-b47f-d5f857413ac1'
AND a.OpdrachtTypeld IN ('60619ca6-054d-4e0f-a2f2-e3b78f7a1e89')
AND a.OnderwerpIndex = 1 AND a.VraagIndex = 2 AND a.OpdrachtIndex = 3
and a.Antwoord = 'Naam medewerker'
```

In dit voorbeeld is ervan uit gegaan dat Onderwerp 1, Vraag 2, Opdracht 3 een Opdracht van het type “Toegewezen aan” is.

Alle snags waarbij onder onderwerp met index 1 een Opdracht van het Type Akkoord met nee is beantwoord

```
SELECT *
FROM Rondeltem ri
JOIN Antwoord a ON a.RondeltemId = ri.Id
JOIN GekozenOptie opt ON opt.AntwoordId = a.Id
WHERE ri.OnderwerpIndex = 1
AND ri.OpdrachtTypeld = '4f225325-1e2d-4662-8e6d-fa073bad94a2'
AND ri.Antwoord = 'Nee'
```