



Best Practice Guide

“Mijn API op DjustConnect”



Om deel te kunnen nemen aan de API-economie moeten bedrijven data ter beschikking stellen. Op dit moment is de beste datadeeltechnologie het gebruik van API's. Deze starterkit maakt bedrijven wegwijs in het wat, waarom en hoe van deze technologie.

Ook DjustConnect is een API-gebaseerd platform. Deze gids licht dan ook heel concreet toe hoe API's beschikbaar te maken in de ConnectShop.



Inhoudstafel

1	Datadelen in de agrovoedingssector, en het waarom van DjustConnect.....	3
2	Identificatie.....	5
3	Je API op DjustConnect - snelgids.....	6
3.1	Stap 1: Registreren op DjustConnect.....	6
3.2	Stap 2: Je maakt je data beschikbaar via API.....	6
3.3	Stap 3: Publicatie en management van je API.....	6
3.4	Stap 4: Start met het verwaarden van je API's.....	7
4	De API-checklist: DjustConnect helpt ze je afvinken.....	8
5	Online leermateriaal.....	10
6	Case studies: getuigenissen van de ondernemings specifieke cases.....	11
7	Datadelen via DjustConnect DATA-AANBIEDER.....	12
7.1	API ontwikkeling.....	12
7.2	Configuratie in DjustConnect.....	15
7.3	Partner API (optioneel).....	17
7.4	Push API (Events).....	17
8	Datadelen via DjustConnect DATA-AFNEMER.....	21
8.1	Applicatie configureren in het DjustConnect Portaal.....	21
8.2	Integratie met de DjustConnect partner API.....	25
8.3	Data gebruiken.....	26
9	API starterkit.....	28
9.1	Wat is een Web API?.....	28
9.2	Waarom een Web API?.....	31
9.3	Web API's & Klantgerichtheid.....	33
9.4	Web API's & Beveiliging.....	33
9.5	Web API's & Performantie.....	35
9.6	Web API's & Onderhoudsvriendelijkheid.....	36
9.7	API Management.....	36



1 Datadelen in de agrovoedingssector, en het waarom van DjustConnect

In de agrovoedingssector produceren we voeding én data. Vanaf het moment dat een product het landbouwbedrijf verlaat, wordt het een deel van een uitgebreide agrovoedingsketen. Elke schakel in de keten voegt tijdens de verdere verwerking van deze producten data toe. Elke schakel bezit bijgevolg data met een eigen potentiële waarde. Door het aggregeren of analyseren van data, kunnen bestaande producten slimmer gemaakt worden of nieuwe digitale producten ontwikkeld worden. Resultaat daarvan is beter advies, slimmere technologieën, optimale service of minder administratie.

De waarde van data neemt dus toe naarmate je data deelt. De kost om je data te ontsluiten, rendeert vaak niet als je de data slechts met 1 speler gaat delen. Eens de 1^{ste} stappen gezet richting data-economie, ontgroeit je al snel het klassieke unilateraal model van data delen en kom je in een spaghetti aan dataconnecties terecht. Hierdoor wordt het moeilijker om het overzicht te bewaren, neemt het aantal en de diversiteit van contracten rond data delen toe en kan je het gevoel krijgen de controle over je data kwijt te zijn. Deel je daarenboven niet enkel je bedrijfseigen data, maar ook data die je beheert van jouw klanten landbouwers, is het ook jouw taak om hen individueel om actieve toestemming te vragen. Een hele klus.

Een datadeelplatform als DjustConnect kan hier een oplossing zijn. Datadeelplatformen leveren infrastructuur en diensten om het datadelen voor bedrijven gemakkelijker te maken. Hierdoor beperk je de eigen investeringen en word je ontzorgd, zodat je als bedrijf in alle vertrouwen data kan delen. DjustConnect biedt jouw bedrijf:

- 1 centrale datamarkt waarop je je data beschikbaar maakt voor de hele sector.
- 1 'open' platform waarlangs het data transport veilig en efficiënt verloopt.
- 1 centraal dashboard waarmee je 100 % de controle behoudt over jouw data en zelf beslist met wie je de data deelt en waarom. Ook de individuele toestemmingen van de individuele landbouwers worden via het DjustConnect dashboard verzameld.
- 1 centraal contract met duidelijke spelregels voor het data delen, conform de geldende wetgeving en Code of Conduct voor data delen in landbouw.

DjustConnect heeft dus als doelstelling om datadelen op een veilige en efficiënte manier mogelijk te maken met respect voor boer en tuinder. Door gebruik te maken van DjustConnect kunnen data-aanbieders hun data ter beschikking stellen van data-afnemers met de toestemming van land- en tuinbouwers. Deze data wordt via API ter beschikking gesteld, en komt dan in de ConnectShop van DjustConnect terecht, waardoor ook nieuwe klanten de weg naar deze databronnen kunnen vinden, en zeer vlot toegang tot deze data kunnen vragen.

1 Iemand heeft een idee voor een nieuwe app of toepassing en is daarvoor op zoek naar data.

In de ConnectShop van DjustConnect bekijkt hij het aanbod van de verschillende databronnen en registreert zich als data-afnemer. Hij kiest de data die hij nodig heeft en in één klik komt zijn vraag terecht bij de aanbieder ervan.



2 Deze data-aanbieder is akkoord met het idee.

Hij geeft zijn toestemming aan de data-afnemer via zijn DjustConnect-dashboard. Hiermee is de deal ook meteen juridisch geregeld. Maar voor de data op transport gezet worden, moet de boer nog akkoord gaan.

3 De boer beoordeelt de data-aanvraag in zijn persoonlijk DjustConnect dashboard.

Als hij ja zegt, komt het datatransport op gang. De boer kan op elk moment met één klik in zijn dashboard dit transport stoppen.



4 De nieuwe app of toepassing is er. De app combineert de data op een slimme manier. Hierdoor krijgt de app-gebruiker beter advies of wordt hij administratief ontzorgd.

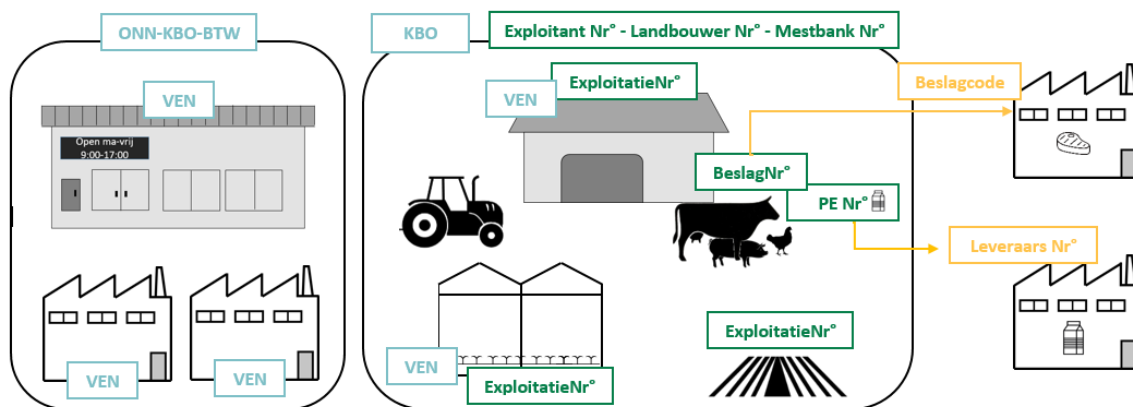


2 Identificatie

Bij datadelen is de unieke identificatie van de actoren en herkomst van de data (Het primaire punt waar de gegevens zijn gegenereerd of verzameld.: de stal, de koe, de sensor, de machine, etc.) de onderwerpen waarover de gegevens gaan essentieel, zodat alle data-gebruikers (aanbieders, afnemers en eigenaars) de garantie hebben waaraan de data is gelinkt.

In de agrovoedingssector is dit vrij complex. Enerzijds zijn er veel verschillende ketens en teelten en de daaraan gelinkte informatie, anderzijds worden de actoren en herkomst van de data door verschillende organisaties vaak op een verschillende manier benoemd.

De unieke identificatie van de landbouwer, het landbouwbedrijf, de percelen, de landbouwmachines, etc. is dus het vertrekpunt om aan datadelen te kunnen doen. Los van het unieke klantnummer dat in elk boekhoudprogramma gebruikt wordt, zijn er vele min of meer officiële nummers in omloop om een landbouwer (of een bepaalde activiteit) te identificeren. Denk maar aan bijvoorbeeld landbouwnummer, ondernemingsnummer, beslagnummer, productie-eenheidsnummer, locatienummer, exploitatienummer, leveraarsnummer, Dit probleem speelde al bij eerdere vormen van data delen, maar doordat deze uitwisseling beperkt bleef tot bedrijven binnen de directe



domeinomgeving, kon geval per geval een beperkte omzettingssleutel gecreëerd worden (vb. leveraarsnummer naar PE nummer).

Een unieke identificatie is dus een noodzakelijke stap is om data over bedrijven heen verder te kunnen combineren en correleren. Aan de hand van een inputworkshop met de bedrijven zullen alle gebruikte nummers in beeld gebracht worden teneinde een mapping te maken. Deze mapping kan dan ofwel door de bedrijven zelf gebruikt worden om het identificatienummer dat zij gebruiken om te zetten naar een consensus identificatienummer. Maar DjustConnect biedt deze mapping ook als dienst aan, en kan ervoor zorgen dat de door bedrijf X aangeleverde identificatie x te transformeren naar een ander identificatienummer y dat gebruikt wordt door een bedrijf Y, en zo zeker de juiste boeren, data-herkomst of bedrijven te identificeren.



3 Je API op DjustConnect - snelgids

Data beschikbaar maken op DjustConnect gebeurt via een API. Een API bouwen en beschikbaar maken op het platform kan heel snel gaan. De volgende 4 stappen moeten doorlopen worden. In de verdere hoofdstukken kan je meer technische informatie vinden over deze stappen.

3.1 Stap 1: Registreren op DjustConnect

Registreren op DjustConnect verloopt via een tweestaps-identificatie. Hiervoor gebruikt het platform het CSAM-systeem, waarbij de identificatie via elektronische identificatie (eID of itsme) gebeurt. Dit is een heel veilig, waardoor we exact kunnen weten wie wie is op DjustConnect. Enkel zo kunnen we een correcte en veilige service garanderen voor het beheer van consents en API's.

3.2 Stap 2: Je maakt je data beschikbaar via API

Als data-aanbieder maak je data beschikbaar via API op het platform. Deze API komt terecht in de ConnectShop, waar data-gebruikers een aanvraag kunnen doen om deze data te mogen gebruiken. Alle partijen tekenen het DjustConnect-contract bij gebruik van het platform. Maar data-aanbieders en -afnemers kunnen daarnaast ook onderling afspraken maken over de voorwaarden voor het delen van deze data. Ook voor het gebruik van DjustConnect dient een jaarlijkse kost betaald te worden door de data-aanbieders en afnemers. De prijs hangt af van de rol, de hoeveelheid databronnen en het aantal land- en tuinbouwers waarvan data beschikbaar is in de bron. De voordelen van het gebruik van het platform, wegen sterk op tegen alle eigen investeringen die je moet doen als je er nog niet mee aan de slag bent.

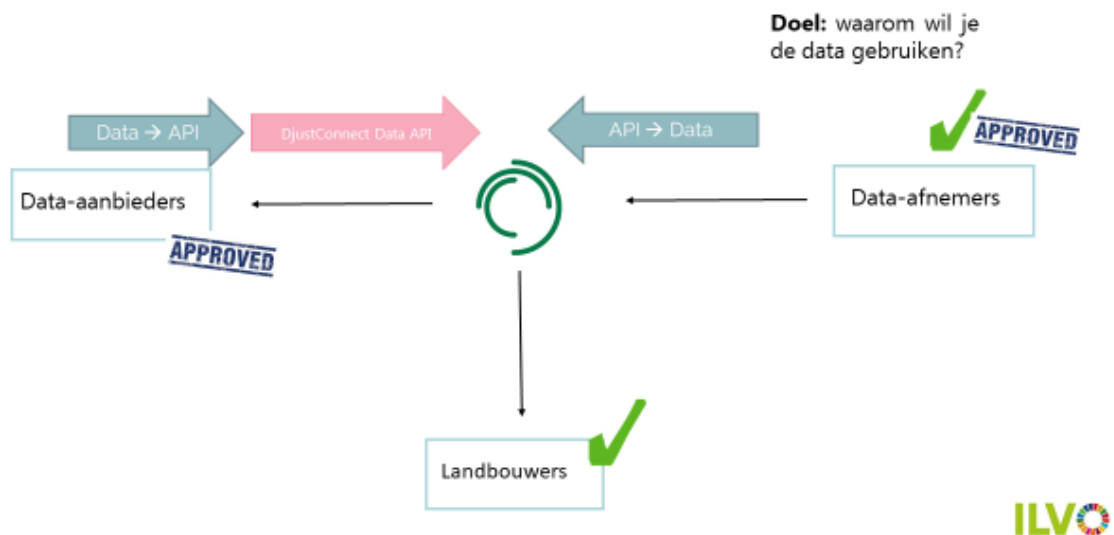
Hoe je als data-aanbieder je API moet bouwen, of als data-afnemer een API kan gebruiken, lees je in verdere hoofdstukken van deze gids.

Bij het bouwen van een API, moet er steeds bekeken worden welke standaarden en ontologie voor de data gebruik zal worden. Voor melk bv. kan je als eenheid zowel liter als volume gebruiken, maar is het wel cruciaal dat aanbieder en afnemer de gegevens op dezelfde manier lezen. DjustConnect verplicht geen ontologie, maar we gaan graag samen op zoek naar bestaande standaarden en mogelijkheden. Standaardisatie maakt het aan de kant van de afnemer veel gemakkelijker om data te integreren

3.3 Stap 3: Publicatie en management van je API

Eens de data-aanbieder de API heeft gebouwd, kan deze gepubliceerd worden in de ConnectShop van het DjustConnect-platform. DjustConnect zorgt voor het API-management, die reguleert wie toegang heeft tot je API.

De DjustConnect-infrastructuur is zo opgezet dat jouw API wordt gedupliceerd, wat een bijkomende veiligheidsstap geeft. Een van de grootste voordelen aan het gebruik van DjustConnect is het consent management. Data wordt pas doorgegeven, na akkoord van de data-aanbieder, voor zover elke individuele land- en tuinbouwer toestemming geeft om hun data te delen voor de specifieke doelstelling die de data-afnemer aangeeft in de data-aanvraag.



Alle personen die toegang hebben tot deze data zijn geïdentificeerd a.d.h.v. eID. Toegang vragen tot de API kan enkel na authenticatie, ofwel via Mutual SSL, waarbij certificaten uitgewisseld worden die aan beider zijden gevalideerd moeten worden, of via OAuth2, waarbij eerste keer met user name en paswoord wordt gewerkt, en daarna token-gebaseerd

3.4 Stap 4: Start met het verwaarden van je API's

Nu je API beschikbaar is op de [ConnectShop](#) van het DjustConnect-platform, kan je starten met zoeken naar klanten voor deze data. Voor het werken aan een digitale bedrijfsstrategie, nadenken over de waarde van data en het opzetten van digitale verdienmodellen, is veel [materiaal](#) beschikbaar via DjustConnect, en is het ook mogelijk om begeleiding te krijgen in de vorm van workshops.



4 De API-checklist: DjustConnect helpt ze je afvinken

Bij het ter beschikking stellen van een API aan (keten)partners, zijn er een aantal zaken cruciaal. Belangrijke aspecten zijn: veiligheid, authenticatie, throttling, specificaties, events, versiebeheer, consent management, ontologie en standaarden.

Het Vlaamse datadeelplatform DjustConnect zorgt ervoor dat bedrijven veilig en eenvoudig data kunnen delen en gebruiken, met respect voor de landbouwer, die veelal data-eigenaar is. DjustConnect is een API-gebaseerd platform. Data-aanbieders kunnen data beschikbaar stellen via API's in de ConnectShop, waar geïnteresseerde partijen een aanvraag kunnen doen om deze data te gebruiken voor een digitale tool. De Connectshop zorgt dat meerdere geïnteresseerden de weg naar je beschikbare API vinden. Maar DjustConnect faciliteert ook de data-uitwisseling zelf op technisch vlak net door bedrijven te ondersteunen in het afvinken van de API-checklist.

Zo verzorgt DjustConnect het API-management van je API. Hierbij neemt DjustConnect het beheer van de API op zich. Concreet wil dit zeggen dat we jouw API beschikbaar maken, het gebruiksbeleid afdwingen en de toegang controleren. Daarenboven verzamelen we metadata over het gebruik van de API zodat je bv. inzicht krijgt in hoe frequent elke data-afnemer jouw API bevroegt.

Onderstaande checklist geeft aan waar je allemaal moet op letten en wat DjustConnect nog meer doet om alle gebruikers te ontzorgen.

Checklist	Belang	Hoe DjustConnect je helpt
Security	Beveiligde toegang tot jouw data	DjustConnect zorgt voor een veilige verbinding en geen direct contact tussen data-aanbieder en data-afnemer.
Authenticatie	Is de gebruiker wel diegene die hij beweert te zijn?	Data-aanbieders valideren het inkomende SSL client certificaat van DjustConnect of OAuth2 wordt voorzien als authenticatie.
Throttling	Is een beveiliging op het systeem, zodat er maar een maximale load aan aanvragen op je systeem kan krijgen.	In DjustConnect kan, desgewenst, het aantal opvragingen per tijdseenheid beperkt worden.
Specificaties	Dit is technische documentatie voor de developers.	OpenAPI specificaties voor de beschikbare API's worden beschikbaar gesteld op het developer portal van DjustConnect.
Events (optioneel)	De data-aanbieder verwittigt de afnemers	DjustConnect biedt de mogelijkheid aan om, via events, data-afnemers op

	wanneer nieuwe gegevens beschikbaar zijn vb. nieuwe labo analyse beschikbaar.	de hoogte te brengen van updates van de beschikbare gegevens.
Versiebeheer	Elke versie van de API heeft een eigen versie nummer.	DjustConnect laat toe dat nieuwe versies van een API beschikbaar komen, maar dwingt contractueel af dat de oude versie minstens 3 maand samen met de nieuwe versie beschikbaar blijft voor bestaande data-afnemers. Daarenboven heeft DjustConnect een strikt beleid rond het overzetten van consent tussen opeenvolgende versies.
Consent management	Is de manier waarop de toestemmingen van de data-eigenaar op het gebruik van de API afgedwongen worden.	Toegang tot een API via DjustConnect kan er enkel komen na het verkrijgen van de juiste toestemmingen met name die van de aanbieder van de API en – indien de landbouwer eigenaar is van de data – die van de landbouwer. Elke landbouwer en data-aanbieder heeft een DjustConnect-dashboard met een overzicht van alle data-aanvragen, waarop hij deze toestemmingen kan geven of intrekken. Dit mechanisme zit vervat in de juridische contracten die elke gebruiker aanvaardt.
Ontologie en standaarden	Data standaarden zijn gedocumenteerde afspraken over representatie, formaat, definitie, structurering, overdracht, manipulatie, gebruik en beheer van gegevens. Een ontologie is een datamodel ontwikkeld binnen een domein, dat alle entiteiten beschrijft als ook de relaties tussen de entiteiten onderling weergeeft.	DjustConnect verplicht geen ontologie. Maar ontologie en standaardisatie maken het aan de kant van de afnemer veel gemakkelijker om data te integreren. Ze verhogen hierdoor vaak ook het gebruik en dus de waarde van de API. Daarom gaan we graag samen op zoek naar bestaande standaarden en mogelijkheden.

Met het aanbieden van deze diensten willen we datadelen in de Vlaamse agrovoedingssector een stap vooruitbrengen. DjustConnect faciliteert, maar het is aan de bedrijven om de API's te bouwen.



5 Online leermateriaal

DjustConnect biedt online leermateriaal aan in de vorm van de webinarreeks: We produceren voeding... en DATA! waarin we ons samen met experts en ervaringsdeskundigen verdiepen in volgende topics:

- **Sessie 1: De data-economie: hoe kan ik meer uit (mijn) data halen?**
 - Wat is data-economie? Joris Relaes, ILVO
 - Geven zonnepanelen en windenergie voldoende kracht om een data-economie aan te zwengelen? Ervaringen uit de energiesector. Dieter Jong, Re.Alto
 - Hoe kan het data deelplatform DjustConnect u hierbij ondersteunen? Stephanie Van Weyenberg, ILVO
 - Opportuniteit of uitdaging? Bedrijven getuigen: Koen Uyttenhove, IOT manager AVR en Founder ODYC DIHAVR en Ben Vanpoucke, IT manager Belgian Pork Group
- **Sessie 2: Hoe kan ik een verdienmodel opbouwen voor datadelen?**
 - Wat hebben airBNb, vinted en een koekjesfabriek gemeen? Een data verdienmodel! – Wim Vanhaverbeke, Surrey Business School
 - Zelf aan de slag? Met deze tools geeft u vorm aan uw digitale strategie of data business case – Bart Minne en Eva Maes, ILVO
 - Data valorisatie via DjustConnect – Stephanie Van Weyenberg, ILVO
 - Bedrijven delen hun eigen ervaringen uit de praktijk: Tammo Van Leeuwen, CEO Fooddesk
- **Sessie 3: Wat zijn API's en waarom zou ik ze gebruiken?**
 - "API's" zo'n typische IT-term? Waarom moet u als CEO toch weten wat dit is? – Stephane Jans, In The Pocket
 - Uw API's op DjustConnect – Stephanie Van Weyenberg, ILVO
 - Bedrijven delen hun eigen ervaringen uit de praktijk: Thierry Bruyneel, partner bij Delaware, Ellen Verhasselt, International Business Development bij SmartWithFood
- **Sessie 4: Spelregels voor datadelen, hoe zit het juridisch?**
 - Getouwtrek om data? Een goed juridisch kader zorgt ervoor dat dit niet hoeft. – Natalie Bertels, KU Leuven
 - Code of Conduct voor datadelen in landbouw – Jürgen Vangeyte, ILVO
 - DjustConnect = 100% controle over je data – Stephanie Van Weyenberg, ILVO
 - Bedrijven delen hun eigen ervaringen uit de praktijk: Dirk Coucke, CEO United Experts

(Her)bekijk de webinarreeks hier: [Webinarreeks: We produceren voeding... en DATA!](#)



6 Case studies: getuigenissen van de ondernemings specifieke cases

Er werden heel wat cases gerealiseerd op DjustConnect. Op enkele van deze cases zoomen we dieper in in dit [artikel](#).
Over 3 cases maakten we een filmpje met getuigenissen van de verschillende spelers, die hun kant van het werken met DjustConnect toelichten.

Milcobel



<https://youtu.be/REOZKsrwKfo>
Lees hier het [artikel](#).

WatchItGrow & AVR Machinery



<https://youtu.be/d884T9eEKhU>
Lees hier het [artikel](#).

Klimrek klimaatscan



<https://youtu.be/xjEm7PtThdo>
Lees hier het [artikel](#).



7 Datadelen via DjustConnect DATA-AANBIEDER

Om veilig datadelen, mét toestemming van het landbouwbedrijf, op DjustConnect mogelijk te maken, zijn er enkele stappen en randvoorwaarden om data via een API te kunnen aanleveren:

- Enkel HTTP GET requests worden ondersteund
- Een server SSL certificaat, bvb. van Letsencrypt, is vereist
- Het inkomende SSL client certificaat van DjustConnect valideren of OAuth2 authenticatie voorzien
- API-documentatie onder de vorm van een Open API Spec (Swagger file)
- Als een landbouwbedrijf toestemming moet verlenen:
 - o Een extra endpoint aanmaken: *farm-id*
 - o Zich ervan verzekeren dat elke endpoint als request parameter of in de response body een landbouwbedrijf identificatie bevat

7.1 API ontwikkeling

Enkel HTTP GET requests worden op dit moment ondersteund door DjustConnect. Om er zich van te verzekeren dat enkel DjustConnect toegang krijgt tot de data van de aangeboden API zijn er twee authenticatie methodes voorzien: enerzijds mutual SSL waarbij enkel calls gemaakt met een bepaald SSL certificaat toegestaan worden of anderzijds via oauth2 authenticatie. Optioneel kunnen ook nog extra HTTP header parameters verplicht worden gemaakt. Deze moeten dan wel medegedeeld worden aan de datagebruikers opdat de gebruikers deze headers kunnen toevoegen.

Om het inkomend DjustConnect SSL certificaat te valideren in een Asp.NET Core C# applicatie draaiende in IIS moeten volgende stappen uitgevoerd worden:

1. Web applicatie toevoegen aan IIS, domein configureren, SSL server certificaat toevoegen en https binding configureren
2. In het IIS *Features* paneel, ga naar *SSL Settings*, *Require SSL* aanzetten en *Client certificates* op *Accept* zetten
3. In C# (zie onderstaande code)
 - a. Implementeer de `IAuthorization` interface als een `RequireCertificateAttribute`
 - b. Gebruik `[RequireClientCertificate]` als attribute voor elke controller klasse

```

public class RequireClientCertificateAttribute : Attribute, IAuthorizationFilter
{
    private const string DJUSTCONNECTCERT_THUMB =
"7C7CCBBA57C04702CCB71F9DD7BAAA56F96EED46";
    public void OnAuthorization(AuthorizationFilterContext context)
    {
        X509Certificate2 cert =
context.HttpContext.Connection.ClientCertificate;
        if (cert == null)
        {
            context.Result = new UnauthorizedResult();
        }
        else if (!cert.Verify() || !IsValid(cert))
        {
            context.Result = new StatusCodeResult(403);
        }
    }

    private bool IsValid(X509Certificate2 cert)
    {
        return string.Equals(cert.Thumbprint, DJUSTCONNECTCERT_THUMB,
StringComparison.InvariantCultureIgnoreCase);
    }
}

```

7.1.1 Farm-id endpoint

Om te weten te komen welke landbouwbedrijven hun toestemming moeten geven voor het gebruik van de aangeleverde data, is er een aparte endpoint nodig die DjustConnect op regelmatige basis kan aanroepen. We raden aan om deze endpoint uit te sluiten van de Open API Spec van de aangeleverde API.

Het contract waaraan de *farm-id* endpoint moet voldoen is het volgende:

- POST request met als body een JSON-object met volgende properties:
 - o *ResourceUrl*: volledige url van de endpoint in kwestie
 - o *FarmIds*: ids van de landbouwbedrijven waarvoor DjustConnect wenst te weten of de endpoint hiervoor data ter beschikking heeft. Deze is leeg wanneer *All* gelijk is aan True
 - o *All*: geeft aan of één van de datagebruikers data voor alle beschikbare landbouwbedrijven wenst op te halen. Wanneer dit True is dan wordt als resultaat de volledige lijst van beschikbare landbouwbedrijf nummers verwacht als antwoord.
- Response bestaande uit een JSON-lijst van landbouwbedrijfnummers

Voorbeeld code voor een *farm-id* endpoint:

```

[ApiExplorerSettings(IgnoreApi = true)]
[RequireClientCertificate]
[Route("api/[controller]")]
[ApiController]
public class FarmIdController : ControllerBase
{
    IDatabaseContext _context;

    public FarmIdController(IDatabaseContext context)
    {
        _context = context;
    }
    [HttpPost]
    [ProducesResponseType(typeof(IEnumerable<string>)), (int)HttpStatusCode.OK]
    public IActionResult Post([FromBody]FarmIdRequest req)
    {
        var resultset = new HashSet<string>(_context.Farms.Select(f => f.Kbo));
        if (req.All)
        {
            return new JsonResult(resultset);
        }
        else
        {
            return new JsonResult(resultset.Intersect(req.FarmIds));
        }
    }
}
public class FarmIdRequest
{
    public string ResourceUrl { get; set; }
    public IEnumerable<string> FarmIds { get; set; }
    public bool All { get; set; }
}

```

7.1.2 OpenAPI Spec

Om de datagebruikers te kunnen voorzien van voldoende informatie over de aangeleverde API wordt er een OpenAPI Spec bestand, ook wel gekend als een Swagger file, verwacht. In het geval van .NET kan deze gegenereerd worden door het gebruik van [NSwag](#) of [Swashbuckle](#). De informatie aanwezig in de OpenAPI Spec wordt weergegeven in het developer portaal (<https://developer.djustconnect.be/>).

7.2 Configuratie in DjustConnect

7.2.1 Partner Details

Naam	
<input type="text" value="en Name"/>	^
<input type="text" value="nl Name"/>	
<input type="text" value="fr Name"/>	

Doel	
<input type="text" value="en Purpose"/>	^
<input type="text" value="nl Purpose"/>	
<input type="text" value="fr Purpose"/>	

Applicatie URL	<input type="text"/>
Naam data aanvrager	<input type="text"/>
Application logo	<input type="text"/>
Rol	<input type="text"/>
Certificaat	<input type="text"/>

Aan te leveren informatie voor het tabblad *Partner details*, deze informatie wordt geconfigureerd door een DjustConnect beheerder:

- Naam van de dataleverancier
- Purpose: Bijkomende informatie over de dataleverancier
- Url van de dataleverancier

Indien de optioneel te gebruiken partner API van DjustConnect ook aangesproken wordt, dan kan op het tabblad *Partner details* de *Primaire sleutel* die bij elke API call meegegeven moet worden als een HTTP header met sleutel: *DjustConnect-Subscription-Key* terug gevonden worden en het SSL certificaat (.cer bestand) geconfigureerd worden.

7.2.2 API Configureren

Om een API te configureren moet er eerst algemene informatie ingegeven worden:

- API Naam
- API Url: basis url van de API bv. <https://wa-sample-api.azurewebsites.net/>
- OpenAPI Spec URL: URL naar de OpenAPI JSON Specificatie bv. <https://wa-sample-api.azurewebsites.net/swagger/v1/swagger.json>
- Url naar de *farm-id* endpoint: bv. <https://wa-sample-api.azurewebsites.net/FarmId>

Nieuwe API

API Naam *

De naam van uw API

API URL *

De basis-URL van uw API

Open API Spec URL

De locatie van de Open API-specificatie waarin uw API wordt uitgelegd

Landbouwbedrijf ID API URL

URL van de landbouwbedrijf ID API endpoint die de Datahub zal gebruiken om de aanwezigheid van bepaalde landbouwbedrijf IDs na te gaan

Annuleren Nieuwe API aanmaken

Vervolgens moet er gekozen worden tussen SSL Certificaat Authenticatie en OAuth2 authenticatie. Bij OAuth2 Authenticatie moeten er extra parameters geconfigureerd worden. Bij SSL Certificaat Authenticatie moet zich men ervan verzekeren dat men het DjustConnect certificaat valideert.

Naast het suggereren van enkele tags waaronder de API terug te vinden zal zijn in het Developer portaal van DjustConnect (<https://developer.djustconnect.be/>), is de laatste en misschien wel belangrijkste stap het configureren van de verschillende endpoints van de API.

Voor elke van de databronnen moet er aangegeven worden:

- Welke toestemming er nodig is voor het gebruik van de databron:
 - o Geen toestemming nodig
 - o Enkel van de dataleverancier
 - o Zowel van de dataleverancier als van het landbouwbedrijf
- Het type landbouwidentificator dat gebruikt wordt (KBO, Leveraarsnummer, ...)
- Locatie van de landbouwidentificator:
 - o Request parameter: als de ID zich in de request url bevindt dan geef je de naam van de parameter zoals deze gekend is in de OpenAPI Spec
 - o JSON-pad: als de ID's zich in de body van de JSON-response bevinden dan moeten er twee parameters ingevuld worden:
 - JSON-pad naar het object: Als de response een lijst van JSON-objecten is, dan is het JSON-pad: \$
 - JSON-pad naar Landbouwbedrijf-ID: Als het response object in de lijst bv. een *kbo* property heeft dan is het JSON-pad: \$.*kbo*
- Beschrijving van de databron

Indien er JSON-paden gedefinieerd moesten worden dan kunnen deze uitgetest worden op <https://jsonpath.com/>.

Na indiening voor goedkeuring zal een DjustConnect beheerder de API publiceren zodat de datagebruikers toestemming kunnen beginnen vragen om de verschillende databronnen binnen de API te beginnen gebruiken.

7.2.3 Toestemming verlenen

Wanneer datagebruikers de gepubliceerde databronnen waarvoor toestemming van de dataleveranciers nodig is, wensen te gebruiken dan komen deze aanvragen terecht in het [Toegangsbeheer](#).

DETERMINE ACCESS RIGHTS TO THESE DATA REQUESTS

Consumer	Purpose	Resource	Requested on	End date	All farms	Status	Review date	Actions
Djustconnect Consumer samples	Consumer web-application in ASP.NET Core for the api's in the Djustconnect Provider Sample	Get a count of cattle in- or outside for specified kbo	24-09-2020	30-12-2030	No	Approved	24-09-2020	✓ ✗
Djustconnect Consumer samples	Consumer web-application in ASP.NET Core for the api's in the Djustconnect Provider Sample	Get an overview of total count of cattle grouped per farm	24-09-2020	30-12-2030	No	Approved	24-09-2020	✓ ✗

In de actie kolom kan al dan niet toestemming verleend worden.

7.3 Partner API (optioneel)

Alle acties en overzichten in de DjustConnect Portal kunnen ook uitgevoerd worden met behulp van de [Partner API](#).

Voor dataleveranciers zijn de relevante endpoints:

- *ProviderAPI*: aanmaken, configureren en publiceren van een nieuwe API
- *ResourceAccessRequest*: Goedkeuren/afkeuren van data aanvragen

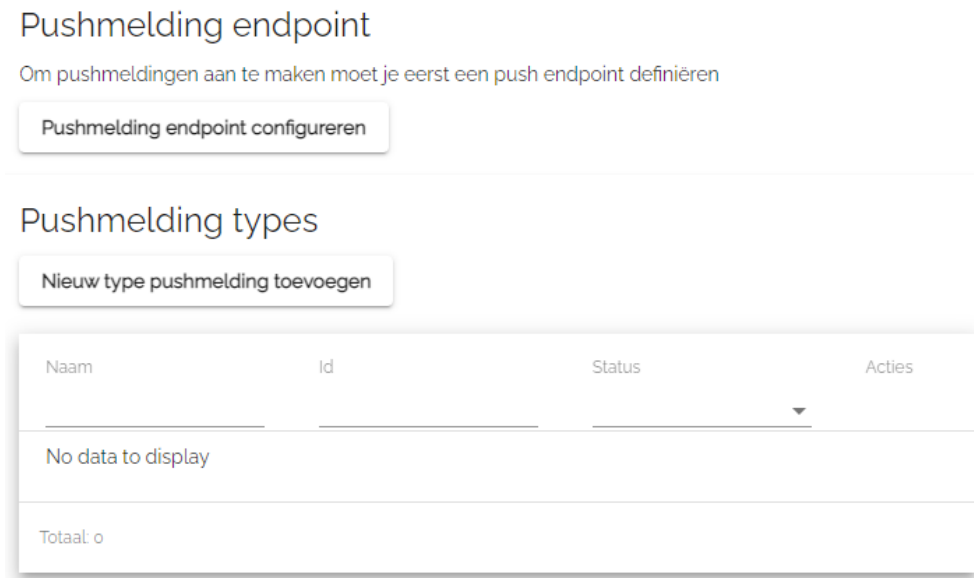
7.4 Push API (Events)

Indien gewenst kunnen er ook pushevents aangemaakt worden door dataleveranciers in het DjustConnect portaal. Dit is vooral nuttig wanneer datagebruikers in realtime op de hoogte willen gebracht worden van nieuwe data of veranderingen in de huidige data. We denken hierbij aan bv. nieuwe laboresultaten. Hierbij kan gekozen worden om de resultaten in één tijd met het pushbericht mee te geven of om het ID van het nieuwe of aangepaste datapunt door te sturen waarbij dan de data, indien gewenst, door de datagebruiker via een web API opgehaald kan worden. DjustConnect gebruikt

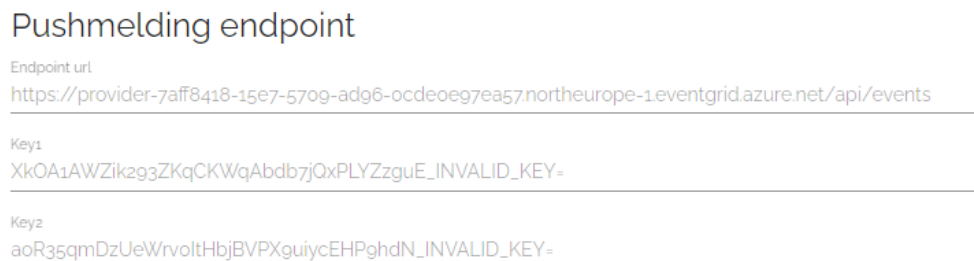
onderliggend Azure Event Grid, dus de tooling en documentatie hiervoor, aangeboden door Microsoft, kan ook steeds geraadpleegd worden.

7.4.1 Pushmelding Configureren

De eerste stap om een pushmelding te configureren is het configureren van een pushmelding endpoint door op "Pushmelding endpoint configureren" te klikken. Hierbij worden 3 zaken aangemaakt: een endpoint url en 2 keys.



Na klikken op "Pushmelding endpoint configureren" krijgt men dan zoiets:




Naast het aanmaken van een pushmelding endpoint moeten er ook pushmelding types geconfigureerd worden. Dit komt conceptueel overeen met de configuratie van een API en is dus ook gelijkaardig. De in te vullen velden zijn:

1. Naam van het eventtype
2. Een beschrijving van het eventtype
3. Documentatie url of document. Hiervoor is geen standaard documentatie formaat maar de bedoeling is om het verstuurd JSON-formaat te verduidelijken op een webpagina of in een document.
4. Aangeven of er al dan niet toestemming verleend moet worden alvorens een pushmelding naar de datagebruiker verstuurd mag worden. Indien ook het landbouwbedrijf zijn toestemming moet verlenen, dan is er ook hier een *farm-id* endpoint vereist. Deze kan net als elke andere web API gebruikmaken van SSL of OAuth autorisatie en eventueel extra vereiste headers.

Na de configuratie kan deze gevalideerd worden en ingediend worden ter goedkeuring.

Here you can find all the details of the selected event type.


Name *

en Name 

Add a description

en Add a description  

Documentation URL

Documentation File 

Provider ...  Farm ID
 Beslag nu...  Farm id URL
 
 The location of the farm id call that DjustConnect will use to

Certificate Authorization **OAuth Authorization**

Add new header config

Back

Create

7.4.2 Pushmelding uitsturen

Om een pushmelding uit te sturen moet een JSON-bericht verstuurd worden met de juiste metadata. Dit kan door gebruik te maken van de door Microsoft Azure aangeleverde .NET bibliotheek: Microsoft.Azure.EventGrid die beschikbaar is op Nuget:

```
// Topic te kopiëren bij Mijn pushmeldingen -> Pushmelding endpoint
var topic = "https://provider-90d2a26d-YOUR-ENDPOINT-URL-HERE.northeurope-1.eventgrid.azure.net/api/events";
// Key te kopiëren bij Mijn pushmeldingen -> Pushmelding endpoint
var topicCredentials = new TopicCredentials("tLZY9Nvn2=YOUR-EVENT-ENDPOINT-KEY-HERE");
// Id te kopiëren bij Mijn pushmeldingen -> Pushmelding types
var eventType = "1a491352-e452-YOUR-EVENT-TYPE-ID-HERE";
var farmNumber = "0262172489";
var data = "Voorbeeld event, kan ook een object zijn";
var events = new List<EventGridEvent>
{
    new EventGridEvent
    {
        Id = Guid.NewGuid().ToString(),
        EventTime = DateTime.Now,
        EventType = eventType,
        Subject = farmNumber,
        Data = data,
        DataVersion = "1.0"
    }
};

using (var client = new EventGridClient(topicCredentials))
    await client.PublishEventsAsync(new Uri(topic).Host, events);
```

Of je kan je bericht zelf samenstellen en versturen als JSON. De request heeft dan de volgende parameters en eigenschappen:

- Het moet een POST request zijn naar de url die vermeld staat op het DjustConnect portaal;
- Er moet een extra header *aeg-sas-key* geconfigureerd worden met als waarde de vermelde key in het portaal;
- Het root element van de body is een array met één of meerdere pushmeldingen;
- Een pushmelding bestaat uit:
 - o id: unieke id voor elk event
 - o eventType: id van het eventtype, terug te vinden in het portaal
 - o subject: identificatienummer van het landbouwbedrijf waarover deze pushmelding gaat
 - o eventTime: tijdstip van het event
 - o data: JSON-waarde of object met de eigenlijke data
 - o dataVersion: versie van het gebruikte dataformaat

Meer lezen over Azure Event Grid:

- <https://docs.microsoft.com/en-us/azure/event-grid/>
- <https://docs.microsoft.com/en-us/dotnet/api/overview/azure/eventgrid>
- <https://social.technet.microsoft.com/wiki/contents/articles/53692.azure-eventgrid-submitting-from-postman-to-custom-topic.aspx>



8 Datadelen via DjustConnect DATA-AFNEMER

Om veilig datadelen, mét toestemming van het landbouwbedrijf, mogelijk te maken zijn er enkele extra stappen nodig om data van een REST API via DjustConnect te kunnen gebruiken:

- Client SSL certificaat, dit mag self-signed zijn
- Extra HTTP header: DjustConnect-Subscription-Key
- Toegangsverzoek tot API's in DjustConnect

In dit document wordt meer informatie verschaft over de stappen die nodig zijn en over enkele extra functionaliteiten die DjustConnect aanbiedt.

8.1 Applicatie configureren in het DjustConnect Portaal

8.1.1 Partner details

Hier vindt u de gegevens van de geselecteerde partner.

Name

Test Consumer

Purpose

Consumer for testing

Application Url

Primaire sleutel

.....



Secundaire sleutel

.....



Certificaat

Thumbprint huidig certificaat: ogF6CgA8CB66197BD78F6EBB4D072A7A58B25873



Opslaan

Aan te leveren informatie voor het tabblad *Partner details*, deze informatie wordt geconfigureerd door een DjustConnect beheerder:

- Applicatie naam
- Purpose: doel van de applicatie
- Applicatie URL: eventuele url naar de applicatie, deze wordt gebruikt om vanuit het data-aanvraag overzicht van de landbouwbedrijven te laten doorlinken naar jullie applicatie

Zelf in te vullen informatie voor het tabblad *Partner details*:

- Certificaat: Publieke sleutel van het SSL certificaat (.cer bestand) dat zal gebruikt worden om API's aan te roepen.

Op het tabblad *Partner details* vind je ook de *Primaire sleutel* die bij elke API call meegegeven moet worden als een HTTP header met sleutel: *DjustConnect-Subscription-Key*.

8.1.2 Toegangsbeheer

De einddatum van uw toegang:

Toegang tot databronnen (wordt goedgekeurd door de provider):

- /api/FarmData/complex - GET - Sample Provider API x
- /api/FarmData/simple - GET - Sample Provider API x
- /api/OAuth/IfarmId - GET - Sample Provider API x Returns a number. - Sample Provider API x
- Returns a word. - Sample Provider API x Returns values. - Sample Provider API x

Toegang Landbouwbedrijf data (wordt goedgekeurd door de eigenaar van het landbouwbedrijf):

PE nummer

Zoeken

Nummers toevoegen Alles verwijderen

100 x 200 x 300 x 13 x 12 x 12 x 45 x 789 x

Aanvraag indienen

In het tabblad *Toegangsbeheer* kan het toegangsverzoek geconfigureerd worden. Deze bestaat uit:

- Einddatum (optioneel): aangezien Focus geen op voorhand vastgelegde einddatum heeft is dit voor jullie niet van toepassing;
- Toegang tot databronnen: kies hier welke API's u wenst te consumeren;
- Toegang Landbouwbedrijfsdata: kies hier welke type identificatienummer (KBO-nummer, productie-eenheidsnummer, beslagnummer, ...) u wenst te gebruiken. In het geval van Focus is dit KBO. Vul hier ook de KBO-nummers in tot de welke u toegang wenst, om te testen kan u manueel enkele KBO-nummers invullen. In productie raden we aan om hiervoor gebruik te maken van de hiervoor voorziene partner API: *ConsumerAccess*.

8.1.3 Toegang landbouwbedrijf

Hieronder vindt u de huidige status (al dan niet reeds gebruiker geregistreerd) van alle landbouwbedrijven waarvoor u toegang heeft gevraagd.

ID ↑ Filter ID	Status -- G. ▾
100	Geen KBO gevonden
12	Geen KBO gevonden
13	Geen KBO gevonden
200	Geen KBO gevonden
300	Geen KBO gevonden
45	Geen KBO gevonden
789	Geen KBO gevonden

Items per page: 10 1 - 7 of 7 |< < > >|

In het tabblad *Toegang landbouwbedrijven* kan de status van de landbouwbedrijven waarvoor toegangsverzoek verzonden is, geraadpleegd worden. Dit geeft een indicatie voor welke landbouwbedrijven er reeds een gebruiker geregistreerd is.

8.1.4 Toegang API's

Hieronder vindt u de consentstatus van alle API's van alle dataleveranciers waarvoor u toegang heeft gevraagd.

Databron ↑ Filter Databron	API Filter API	Dataleverancier Filter Dataleverancier	API Status -- Geen ▾
/api/FarmData/complex - GET	Sample Provider API	Test Provider	Goedgekeurd
/api/FarmData/simple - GET	Sample Provider API	Test Provider	Goedgekeurd
/api/OAuth/ffarmIdl - GET	Sample Provider API	Test Provider	Goedgekeurd
Returns a number.	Sample Provider API	Test Provider	Goedgekeurd
Returns a word.	Sample Provider API	Test Provider	Goedgekeurd
Returns values.	Sample Provider API	Test Provider	Goedgekeurd

Items per page: 10 1 - 6 of 6 |< < > >|

In het tabblad *Toegang API's* kan er nagekeken worden of er voor alle aangevraagde API's bij de verschillende dataleveranciers toestemming gekregen is.

8.1.5 Toegang API's/landbouwbedrijven

Hieronder vindt u per API de consentstatus van alle landbouwbedrijven waarvoor u toegang heeft gevraagd.

ID Filter ID	Databron ↑ Filter Databron	KBO Status -- Geen -- ▾	API Status -- Geen .. ▾	Consent Status -- Geen -- ▾
100	/api/FarmData/com	Geen KBO gevonden	Goedgekeurd	Niet van toepassing
12	/api/FarmData/com	Geen KBO gevonden	Goedgekeurd	Niet van toepassing
13	/api/FarmData/com	Geen KBO gevonden	Goedgekeurd	Niet van toepassing
200	/api/FarmData/com	Geen KBO gevonden	Goedgekeurd	Niet van toepassing
300	/api/FarmData/com	Geen KBO gevonden	Goedgekeurd	Niet van toepassing
45	/api/FarmData/com	Geen KBO gevonden	Goedgekeurd	Niet van toepassing
789	/api/FarmData/com	Geen KBO gevonden	Goedgekeurd	Niet van toepassing
100	/api/FarmData/sim	Geen KBO gevonden	Goedgekeurd	Niet van toepassing
12	/api/FarmData/sim	Geen KBO gevonden	Goedgekeurd	Niet van toepassing
13	/api/FarmData/sim	Geen KBO gevonden	Goedgekeurd	Niet van toepassing

In het tabblad *Toegang API's/landbouwbedrijven* kan er nagegaan worden of een landbouwbedrijf toestemming verleend heeft voor het ophalen van zijn data bij een databron.

8.1.6 Status API's

Hieronder vindt u de huidige status van alle API's waarvoor u toegang heeft gevraagd.

Databron	Status
/api/FarmData/complex - GET	Ok
/api/FarmData/simple - GET	Ok
/api/OAuth/ifarmIdI - GET	Ok
Returns a number.	Ok
Returns a word.	Ok
Returns values.	Ok
Totaal: 6	

In het tabblad *Status API's* kan er nagegaan worden of een API bereikbaar is.

8.2 Integratie met de DjustConnect partner API

Om de integratie met andere systemen te faciliteren zijn er verschillende API endpoints voorzien in DjustConnect. De Swagger file van deze REST API kan geraadpleegd worden op <https://partnerapi.djustconnect.be/index.html>.

8.2.2 ConsumerAccess

De configuratie van het toegangsverzoek in de tabblad *ToegangsBeheer* kan ook via de *ConsumerAccess* endpoint gebeuren.

Als u de landbouwbedrijven waarvoor u toegang vraagt wil configureren raden we aan om eerst een GET request te doen op de *ConsumerAccess* endpoint en daarna de *farmIds* eigenschap van het JSON-object aan te passen op basis van de lijst van landbouwbedrijven aangesloten bij Focus en het volledige object terug te sturen met een POST request op dezelfde endpoint.

Het volledige toegangsverzoek kan ook aangemaakt worden door gebruik te maken van de *Resource* en *FarmIdType* endpoints.

8.2.3 FarmStatus, RarStatus en DarStatus (optioneel)

De informatie die beschikbaar is op de tabbladen *Toegang landbouwbedrijven*, *Toegang API's* en *Toegang API's/landbouwbedrijven* kan desgewenst ook via de *FarmStatus*, *RarStatus* en *DarStatus* API endpoints geraadpleegd worden.

8.2.4 Consumer/resource-health (optioneel)

Ook de informatie van het tabblad *Status API's* kan nagegaan worden via de endpoint *Consumer/resource-health*.

8.2.5 FarmMapping (optioneel)

Indien één van de API's die men wenst te gebruiken, gebruik maakt van een landbouwbedrijf-identificator die niet gekend is in het eigen systeem dan kan de vertaling hiervoor via de *FarmMapping* endpoint opgevraagd worden. Hiermee kan dus bv. een KBO-nummer vertaald worden naar een PE- of beslagnummer. Merk op dat in sommige gevallen meerdere waarden teruggegeven worden.

8.3 Data gebruiken

Eens dat de applicatie en het toegangsverzoek geconfigureerd en goedgekeurd zijn in DjustConnect, kan de eigenlijke data opgehaald worden bij de dataleverancier.

Alle technische informatie over de verschillende API's van dataleveranciers is terug te vinden op <https://developer.djustconnect.be/>. Kijken we bijvoorbeeld naar de API endpoint "[Inagro wateranalyses \(overzicht rapporten\)](#)" dan zien we volgende informatie:

- Url: `https://apim.djustconnect.be/water-analyse-rapporten-inagro/api/Document/lijst/{Ondernemingsnummer}`
- Parameter:
 - o *Ondernemingsnummer* van het type string
 - o *outdoor* van het type boolean
- Response:
 - o Lijst van DTODocument objecten:
 - *id*: int32
 - *datumRapport*: string
 - *staalnameDatum*: string

Om deze API te gebruiken moeten we, gebruikmakend van het SSL certificaat en een extra HTTP header, met als key "DjustConnect-Subscription-Key" en als waarde de Primaire sleutel, een HTTP GET request doen naar deze url met een ondernemingsnummer. Indien er toestemming verleend is, zal deze API dan een JSON-object bestaande uit een lijst van objecten met *id*, *datumRapport* en *staalnameDatum*.

Stappen voor aanmaak van een self-signed certificaat in een Windowsomgeving waarbij de datagebruiker een webapplicatie is die draait onder IIS op het domein `focus.boerenbond.be`:

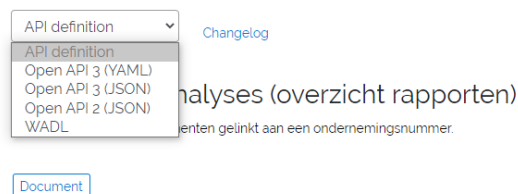
- 1) In een admin powershell op de server:
`$cert = New-SelfSignedCertificate -certstorelocation cert:\localmachine\my -dnsname focus.boerenbond.be`
- 2) In de Windows certificate store:
 - a. Exporteer het certificaat in de Windows certificate store één keer met de private key (pfx) en één keer zonder de private key (cer)
 - b. Right click the certificate -> Tasks -> Manage Private Keys -> Grant rights to the name of the IIS application pool user e.g. "IIS AppPool\focus.boerenbond.be"
 - c. Kopieer het certificaat naar de "Trusted Root Certification Authorities"
- 3) Het geëxporteerde cer bestand opladen in `portal.djustconnect.be` in het tabblad "*Partner details*"
- 4) Om het zelfde certificaat te gebruiken tijdens bv. de ontwikkeling kan je het certificaat importeren in de "Local machine certificate store"

Voorbeeld code C# code om een HttpClient object te initialiseren met het client SSL certificaat en de *DjustConnect-Subscription-Key* header:

```
var store = new X509Store("My", StoreLocation.LocalMachine);
store.Open(OpenFlags.ReadOnly);
var certificates = store.Certificates.Find(X509FindType.FindByThumbprint, "HIER DE THUMBPRINT VAN UW SSL CERTIFICAAT", false);
if (certificates.Count == 0)
{
    throw new InvalidOperationException($"Certificate not found for CN=focus.boerenbond.be in LocalMachine/My.");
}
var certificate = certificates[0];
var clientHandler = new HttpClientHandler();
clientHandler.ClientCertificates.Add(certificate);
clientHandler.ClientCertificateOptions = ClientCertificateOption.Manual;
var client = new HttpClient(clientHandler);
client.DefaultRequestHeaders.Add("DjustConnect-Subscription-Key", "HIER DE PRIMAIRE SLEUTEL VANUIT tabblad Partner details");
```

Eens geconfigureerd kan het *client* object gebruikt worden om de API endpoint aan te roepen. De code hiervoor kan handmatig geschreven worden of gegenereerd worden op basis van de specificaties die je kan downloaden op <https://developer.djustconnect.be>.

Water analyse rapporten Inagro



Meer informatie over het genereren van client code kan je hier terugvinden:

- <https://swagger.io/tools/swagger-codegen/>
- <https://stackoverflow.com/questions/54094688/generate-net-client-from-swagger>
- <https://medium.com/@unchase/how-to-generate-c-or-typescript-client-code-for-openapi-swagger-specification-d882d59e3b77>



9 API starterkit

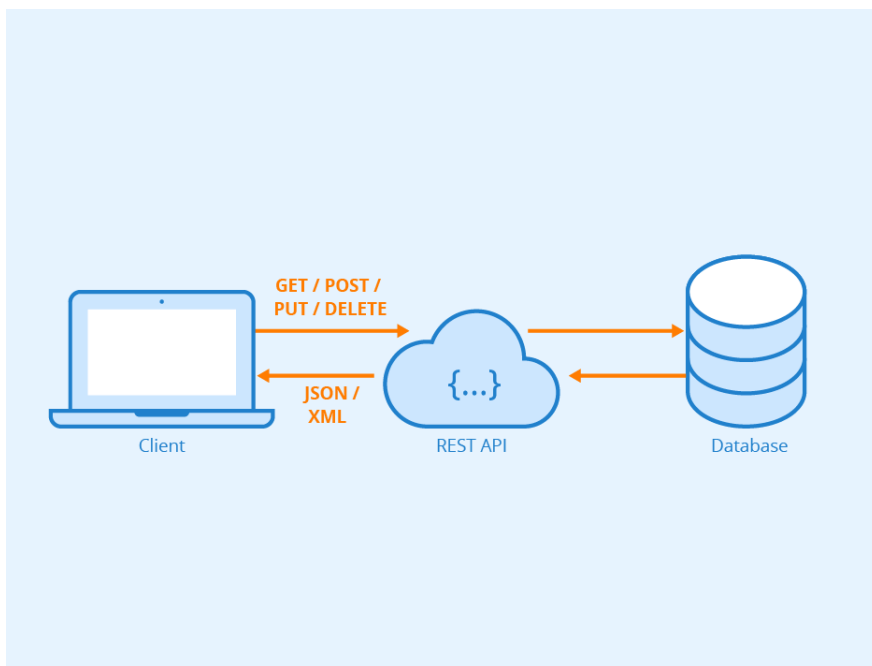
In deze API starterkit kan je algemene uitleg terugvinden over wat API's zijn, welke voordelen ze brengen, en wat je technisch meer van ze moet weten.

9.1 Wat is een Web API?

Voor het delen van de data via DjustConnect wordt gebruik gemaakt van een technologie genaamd REST gebaseerde web API's. De term API verwijst naar "Application Programming Interface". Een API is een set van methodes die een computerprogramma kan gebruiken om te communiceren met een ander computerprogramma. Enkele voorbeelden van API's zijn:

- De Android API waar mobiele ontwikkelaars gebruik van maken om mobiele applicaties te bouwen en meldingen op het scherm te tonen.
- De Windows API waar ontwikkelaars de visuele aspecten van programma's zoals Word en Excel kunnen programmeren maar ook bijvoorbeeld het openen van bestanden
- De Google Maps API die website bouwers kunnen integreren in hun websites om kaarten te tonen

De Google Maps API is een voorbeeld van een web API omdat deze gebruik maakt van het internet voor de communicatie tussen twee computerprogramma's. Zoals te zien is in Figuur 1 bestaat de communicatie bij een web API uit een client die eerst een commando stuurt naar de server via het internet waarna de server een antwoord zal terugsturen.



Figuur 1 Overzicht van de onderdelen en de data stromen bij een eenvoudige web API.

Door het gebruik van Web API's kan er, mits onderlinge toestemming, een data uitwisseling plaatsvinden zonder menselijke interventie. De grote voordelen hierbij zijn dat de uitgewisselde data steeds de meest actuele versie kan zijn, dat deze opgevraagd kan worden wanneer gewenst door de client en dat de data aangeleverd kan worden in een formaat dat door een computerprogramma geïnterpreteerd kan worden. Enkel voorbeeld toepassingen zijn hiervan zijn:

- Weergeven van het actuele geleverde vermogen van zonnepanelen
- Uitwisselen van facturatiegegevens met een boekhoudpakket
- Doorgeven van de GPS locatie van een voertuig of koe aan een digitale kaart

Technisch bekeken maken REST (REpresentation State Transfer) gebaseerde web API's gebruik van de HTTP-standaard en hebben een set aan richtlijnen waar deze aan moeten voldoen:

- Client-server architecture: een scheiding tussen de gebruiker van de data en de aanbieder van de data
- Stateless: de communicatie tussen de client en server gebeurt stateless, m.a.w. het communicatie protocol zelf houdt geen sessie status bij. De client moet deze dan ook steeds communiceren met de server.
- Cacheability: clients kunnen de antwoorden van de server gaan bijhouden (cachen), indien dit niet gewenst is moeten de antwoorden (responses) van de server dit aangeven.
- Layered system: een client maakt geen onderscheid of deze verbonden is met een tussenlaag of met de eind server. Het gebruik van tussenlagen kan de schaalbaarheid en de veiligheid van een systeem vergroten.
- Uniform interface: fundamenteel voor het ontwerp van een REST architectuur

- Individuele databronnen worden geïdentificeerd door de aanvraag (request) doormiddel van bv. URIs.
- Als een client een resource opgevraagd heeft bezit hij voldoende metadata om de resource aan te passen of te verwijderen
- Elke response bevat voldoende informatie om te beschrijven hoe deze uitgelezen moet worden. Het media type kan bv. gebruikt worden om de te gebruiken parser te bepalen.

HATEOAS (Hypertext as The Engine Of Application State): meegeven van voldoende informatie aan de client zodat deze op dynamische wijze de links naar de benodigde resources kan ontdekken. Toegepast op web API's krijgen we een systeem dat bestaat uit:

- Een basis URL (bv. <https://api.example.org>)
- Het gebruik van HTTP methodes zoals GET, POST, PUT, PATCH en DELETE
 - GET: ophalen van een resource van een server
 - POST: toevoegen van een nieuwe resource
 - PUT: aanpassing van een bestaande resource
 - PATCH: gedeeltelijke aanpassing van een bestaande resource
 - DELETE: verwijderen van een resource
- Het gebruik van HTTP status codes, bv:
 - 200: ok
 - 401: geen toestemming
 - 404: onbekend adres
 - 500: server fout
- Een manier om te linken naar de verschillende resources. Dit wordt meestal opgevangen door het ter beschikking stellen van een gebruiksvriendelijk overzicht van de verschillende endpoints van een API zoals bv. een OpenAPI Specification.
- Gebruik van gestandaardiseerde data formaten zoals JSON, XML, JSON-LD, ...

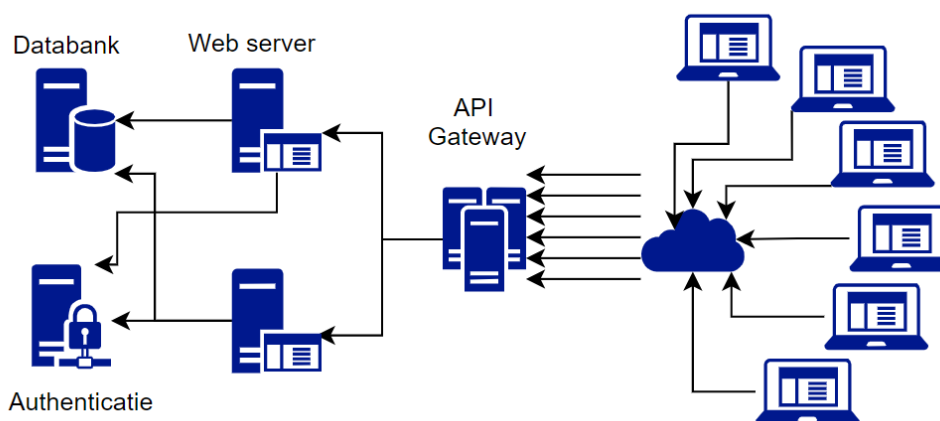
Meer lezen:

- https://en.wikipedia.org/wiki/Representational_state_transfer
- https://en.wikipedia.org/wiki/Web_API
- <https://restcookbook.com/>

9.2 Waarom een Web API?

Enkele voordelen van het gebruik van een Web API voor het delen van data zijn:

- Gebruik van een gestandaardiseerde manier van communiceren tussen de client en de server. Met als gevolg een wijde beschikbaarheid aan technologieën om data te bevragen.
- Doordat alle sessie informatie door de client bij elke request meegegeven worden is de code voor het maken van requests zeer leesbaar.
- Web API's maken het mogelijk om caching toe te passen. Caching gaat bij herhaaldelijk opvragen van min of meer statistische data een reeds voorberekend antwoord terugsturen. Hierdoor kan er efficiënt omgegaan worden met de server, client en communicatie capaciteit.
- De mogelijkheid om meerdere lagen te gebruiken laat toe om de performantie en veiligheidseigenschappen van de server te wijzigen zonder een impact te hebben op de client. In Figuur 2 zien we een voorbeeld architectuur waarbij de server bestaat uit een API Gateway waarmee de verschillende clients communiceren. Deze API Gateway communiceert dan het verzoek van de clients naar één van de web servers die op hun beurt weer de authenticatie server en databank server aanspreken.
- Het gebruik van open standaarden zoals HTTP methodes, HTTP status codes, OpenAPI Specification, data formaten zoals JSON. Dankzij het gebruik van deze standaarden is dit een zeer breed ondersteunde technologie.



Figuur 2 Voorbeeld architectuur waarbij gebruik gemaakt wordt van de mogelijkheid om meerdere lagen te gebruiken

9.2.1 Web API vs Web service

In plaats van het gebruik van een REST gebaseerde web API bestaan er nog andere alternatieven om data te delen via HTTP. Een historisch veel gebruikte methode hiervoor zijn SOAP (Simple Object Access Protocol) Web services. Het is een set van standaarden dat XML-berichten stuurt als request en XML-berichten als response terugstuurt. Enkele nadelen van SOAP zijn:

- Bandbreedte die tot 100% hoger ligt ten opzichte van een vergelijkbare REST API met als resultaat hoger data verbruik en hogere response tijden
- Minder flexibiliteit: enkel XML, sterke koppeling tussen client en server
- Meer tooling nodig om een server en een client te kunnen maken

Daar tegenover staat dat bij SOAP web services er een verplichte standaard specificatie is op basis waarvan er met de juiste tooling code voor verschillende programmeertalen kan gegenereerd worden. Bij REST gebaseerde web API's is dit niet verplicht maar kan dezelfde functionaliteit bekomen worden door het gebruik van een standaard zoals OpenAPI Specification en de bijbehorende tooling hiervoor.

Meer lezen:

- <http://dx.doi.org/10.5539/mas.v12n3p175>
- <https://raygun.com/blog/soap-vs-rest-vs-json/>
- <https://da-14.com/blog/ultimate-guide-api-architecture-rest-soap-or-graphql>

9.2.2 Web API vs GraphQL

In 2015 werd de GraphQL taal publiek bekend gemaakt door de ontwikkelaars van Facebook. Deze maakt het voor de client mogelijk om exact aan te geven welke data deze wenst te ontvangen van de server. Dus in tegenstelling tot een REST API waarbij het de server is die definitie van de data bepaalt is het bij een GraphQL de client die dit bepaald. Het voordeel hiervan is dat er bij GraphQL er steeds het optimale aantal requests en optimale volume aan data verzonden wordt naar de client. Het nadeel is dat een GraphQL geen gebruik kan maken van de standaard HTTP caching en dat er geen controle is over de mogelijke queries waardoor dit kan leiden tot een grote belasting van de database en web server. Bijkomend is deze standaard nog relatief jong waardoor de tooling hiervoor nog niet zo uitgebreid is.

Meer lezen:

- <https://graphql.org/>
- <https://da-14.com/blog/ultimate-guide-api-architecture-rest-soap-or-graphql>
- <https://www.javatpoint.com/graphql-advantages-and-disadvantages>
- <https://blog.logrocket.com/5-reasons-you-shouldnt-be-using-graphql-61c7846e7ed3/>

9.2.3 Web API vs (S)FTP

Een ander veelgebruikt alternatief voor web API's voor het delen van data is het delen van bestanden via (S)FTP. Hoewel data delen via (S)FTP zijn voordeel kan hebben bij het delen van grote volumes data zijn er ook enkele nadelen aan verbonden:

- Geen mogelijkheid om ad hoc de benodigde data te filteren tijdens de request.
- De asynchrone aard van de data uitwisseling verhindert de bevraging van variabele data en data voor gebruik in realtime applicaties.
- Moeilijker te bevragen voor klanten van bedrijven aangezien de SFTP poort geblokkeerd kan zijn op de firewall
- Data uitwisseling formaat is niet gestandaardiseerd waardoor client libraries niet gegenereerd kunnen worden

Meer lezen: <https://enterprisearchitecture.harvard.edu/data-exchange-mechanisms>

9.3 Web API's & Klantgerichtheid

9.3.1 Documentatie

Om de client applicaties (data gebruikers) te kunnen voorzien van voldoende informatie over de gemaakte web API is het essentieel om documentatie te genereren. Een veelgebruikte standaard hiervoor is de OpenAPI Specification, ook wel gekend als Swagger specificatie. Het is een programmeertaal agnostische standaard voor de beschrijving van HTTP API's, die mensen en computers toelaat om de mogelijkheden van een API te begrijpen. Dit laat toe om op basis van een specificatie bestand de mogelijkheden van een API weer te geven in een website zoals bv. op het [Developer portaal](#) van DjustConnect. In het geval van een .NET web API kan een specificatie bestand voor een web API gegenereerd worden door gebruik te maken van NSwag. Voor de andere veel gebruikte programmeertalen is er een vergelijkbare tool beschikbaar. Om de klantvriendelijkheid van een API te testen heeft het Support Centre for Data Sharing een tool ontwikkeld. Deze is beschikbaar op <https://api-friendliness.eudatasharing.eu/>. Een overzicht van de gebruikte criteria voor klantvriendelijkheid wordt hier weergegeven <https://eudatasharing.eu/index.php/technical-aspects/classification-scheme-developer-friendly-apis>.

Meer lezen:

- <https://github.com/OAI/OpenAPI-Specification>
- <https://docs.microsoft.com/en-us/aspnet/core/tutorials/getting-started-with-nswag>
- <https://swagger.io/tools/open-source/open-source-integrations/>

9.3.2 Data ontologie

Om verwarring te voorkomen bij het gebruik van een web API is het aangewezen om steeds zoveel mogelijk de meest gebruikte termen en eenheden te gebruiken voor de aangeleverde data. Idealiter wordt gebruikt gemaakt van een ontologie voor de data in kwestie. Wanneer deze niet beschikbaar is of niet helemaal gepast is voor de uit te wisselen data is het belangrijk om, indien er verwarring mogelijk is de gebruikte eenheden en de gebruikte terminologie te verduidelijken in de OpenAPI Specificatie.

9.4 Web API's & Beveiliging

De eerste stap voor het bouwen van een veilige REST API is ervoor zorgen dat de API enkel bereikbaar is via het HTTPS protocol. Tegenwoordig kan dit zeer eenvoudig door gebruik te maken van de tools ter beschikking gesteld door Let's Encrypt (<https://letsencrypt.org>).

Voor niet-publieke API's is de volgende stap ervoor zorgen dat er toegangscontrole is. Hiervoor bestaande verschillende methodes met elk hun voor- en nadelen, de meest gebruikte zijn:

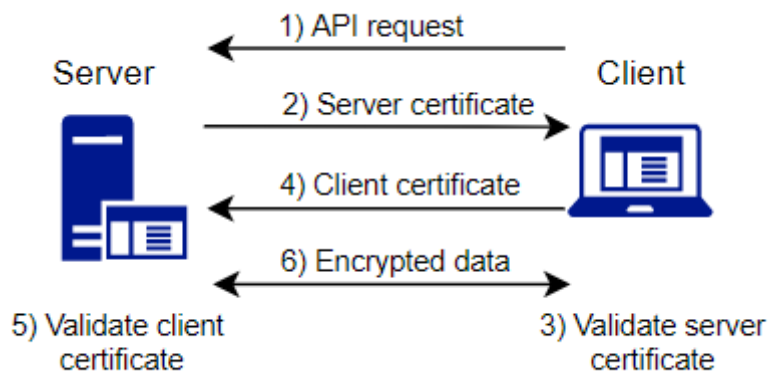
- Basic HTTP Authentication: Standaard gebruikersnaam/wachtwoord authenticatie
- API Keys: Hierbij wordt een key gegeven aan een reeds gekende gebruiker die dan bij verdere communicatie kan gebruikt worden. Grootste voordeel ten opzichte van Basic Auth is dat deze afzonderlijk teruggetrokken kan worden. Nadeel is dat indien de key niet op een veilige manier doorgegeven wordt deze door eender wie kan gebruikt worden.

- OAuth2: Maakt gebruik van een access token die aangemaakt wordt na succesvolle identificatie op basis van bv. gebruikersnaam/wachtwoord. Beste keuze voor de identificatie van gebruikers en het toekennen van de juiste permissies.
- OpenID Connect: Laag bovenop OAuth2 die het mogelijk maakt om de identificatie gegevens bij een autorisatie server te verifiëren.
- Mutual SSL (Client Certificate Authorisation): Hierbij gaat de server de client identificeren op basis van het gebruikte SSL certificaat. Dit is zeer schaalbaar en veilig mits het gebruikte client certificaat veilig bewaard wordt.

Voor de beveiliging van web APIs heeft DjustConnect 2 mogelijkheden voor data aanbieders voorzien:

- 1) Mutual SSL: hierbij is het de bedoeling dat data aanbieders bij elke request valideren dat het certificaat van de inkomende request afkomstig is van DjustConnect (stap 5 van Figuur 3).
- 2) OAuth2: de credentials en configuratie informatie hiervan kunnen door de data aanbieder zelf ingevuld worden in het DjustConnect portaal

Bovendien is het binnen DjustConnect ook nog mogelijk om van data gebruikers te verlangen van een API key mee te geven via een http header. Hierdoor kan een data aanbieder bijkomende autorisatie logica voorzien per data gebruiker.



Figuur 3 Overzicht van Mutual SSL authenticatie.

Meer lezen:

- https://cheatsheetseries.owasp.org/cheatsheets/REST_Security_Cheat_Sheet.html
- <https://blog.restcase.com/4-most-used-rest-api-authentication-methods/>

9.5 Web API's & Performantie

Enkele tips om de performantie van API's te verbeteren.

9.5.1 Payload beperken

Een veel gebruikte manier om de response grootte en response tijd te beperken is om paginering van de resultaten toe te passen. Bijvoorbeeld de partner API van DjustConnect gebruikt extra headers hiervoor in de response: X-Pages, X-PageNumber, X-PageSize en X-TotalCount. Om een bepaalde pagina op te halen of de pagina grootte aan te geven kan respectievelijk de PageNumber of de PageSize meegegeven worden in de url.

Een tweede belangrijk aspect is de optimalisatie van de organisatie van een API zodat deze voldoende maar niet teveel data meegeeft. Indien er verschillende data gebruikers zijn en de noden gekend zijn kan er eventueel gekozen worden om aparte endpoints te voorzien. Bij conflicterende of onbekende noden kan het een goed idee zijn om een extra request parameter te voorzien waarmee de data gebruikers de gewenste velden kunnen uitfilteren.

Ook tijdens het transport van de data kan de payload beperkt worden door gebruik te maken van compressie of compacte data formaten.

9.5.2 Caching

Door gericht gebruik te maken van caching kan de performantie van een API sterk verhoogt worden. Dit is natuurlijk sterk afhankelijk van de veranderlijkheid van de data. Om regelmatige bevraging van weinig veranderlijke data te voorkomen kan er ook gebruik gemaakt worden van pushmeldingen. Hiebij gaat de data leverancier een API van de data gebruiker gaan aanspreken wanneer er nieuwe data is. Dit scenario wordt ondersteunt in DjustConnect.

9.5.3 Applicatie tuning

Ook op applicatie niveau kan er soms nog winst geboekt worden op het vlak van performantie. Enkele manieren om dit te doen zijn:

- Connection pooling, hierdoor vermijdt men steeds een nieuwe databank verbinding te moeten maken
- Query tuning: optimalisatie van de databank queries
- Asynchroon wegschrijven van logging informatie, hierdoor moet men niet wachten op het voltooiën van de logging alvorens het antwoord terug te sturen

9.5.4 Rate limiting

Bij meerdere data gebruikers kan het nodig zijn om rate limiting in te voeren om te vermijden dat overmatige bevraging door één van de gebruikers impact heeft op de performantie van de API voor de andere gebruikers.

Meer lezen:

- <https://medium.com/rpdstartup/rest-api-performance-tuning-getting-started-7a6efefage20>
- <https://nordicapis.com/everything-you-need-to-know-about-api-pagination/>
- <https://blog.logrocket.com/caching-strategies-to-speed-up-your-api/>

9.6 Web API's & Onderhoudsvriendelijkheid

Net als bij het programmeren van andere websites is het belangrijk om enkele basis principes voor het schrijven van onderhoudsvriendelijke programma's in het achterhoofd te houden. Zaken zoals het KISS-principe en SOLID in geval van het gebruik van een object-georiënteerde programmeertaal zijn ook hier van tel. Ook het schrijven van documentatie, unit testen, integratie testen (Postman) en performantie testen (JMeter) kunnen bijdragen tot een grotere onderhoudbaarheid.

Meer lezen:

- <https://www.software.ac.uk/resources/guides/developing-maintainable-software>
- <https://bobbelderbos.com/2016/03/building-maintainable-software/>
- <https://www.postman.com/automated-testing/>
- <https://jmeter.apache.org/>

9.7 API Management

Om het beheer, de beveiliging en de performantie van web API's te vereenvoudigen bestaan er verschillende API management software aanbieders in de cloud of om te installeren op een eigen server. Enkele bekendere voorbeelden hiervan zijn:

- Apigee
- Azure API Management
- Dell Boomi
- Kong Enterprise
- Mulesoft

Meer lezen:

- https://en.wikipedia.org/wiki/API_management
- <https://www.softwaretestinghelp.com/api-management-tools/>