Technisch Ontwerp

Content Publicatiesysteem

Klant

Provincie Zeeland

Contactpersoon

A.P. Allemekinders

Versie

1.0

Datum

20-12-2022

Auteurs

Judith Steijne

Erik Boode

Inhoudsopgave

Inleiding **3**
Webservice *3*
Gebruikte technieken/producten *3*

Azure Architectuur **4**
Azure App Service *4*
Azure Storage *5*
Azure CDN 5
Azure App Registration *5*

SharePoint Architectuur **6**
Sitecollecties *6*
Handmatige invoer *6*
Inhoudstypes *6*
Metadataspecificaties *7*
Power Automate stroom *10*

Autorisatie **11**
Services onderling *11*
Binnenkomende verzoeken *11*
SharePoint Online *12*

API Services **14**
AppSettings.json Settings *14*
Azure Storage Table 'Settings' *14*
Import *15*
Nieuwe bestanden en metadata uploaden naar SharePoint Online 15
Gewijzigde bestanden uploaden naar SharePoint Online 16
Gewijzigde metadata opslaan in SharePoint Online 17
Genereren ObjectId voor nieuw (SharePoint) bestand 18
Zoeken *19*
Vinden bestand adhv ObjectId 19
Vinden metadata adhv ObjectId 19
Export *21*
Nieuwe bestanden en metadata uploaden naar Azure Storage Container 21
Gewijzigde bestanden & metadata uploaden naar Azure Storage Container 22
Verwijderde bestanden en metadata in Azure Storage Container weggooien 22

Naslagwerk **24**

Bijlage: Inhoud Metadatatabel Bestandssoort **25**

Inleiding

De basis is een content publicatie systeem in SharePoint Online waarin bestanden worden opgeslagen op diverse locaties. Daarmee kan onderscheid worden gemaakt tussen bestanden alleen intern beschikbaar (met diverse veiligheidsniveau ’s) en bestanden extern publiekelijk beschikbaar (geen beveiliging). Bestanden welke extern beschikbaar moeten zijn zullen elke X minuten via een webjob worden gekopieerd naar een Azure Storage Container. Deze Storage Container is via een CDN beschikbaar voor gebruik in de website en eventueel ook in een ander extern systeem.

Verder worden tijdens het bijwerken van de Azure Storage Container per nieuw/gewijzigd/verwijderd object een attenderingsbericht gestuurd naar de Elastic Search Connector service. Deze functionaliteit binnen deze service wordt ontwikkeld door Provincie Zeeland.

Webservice

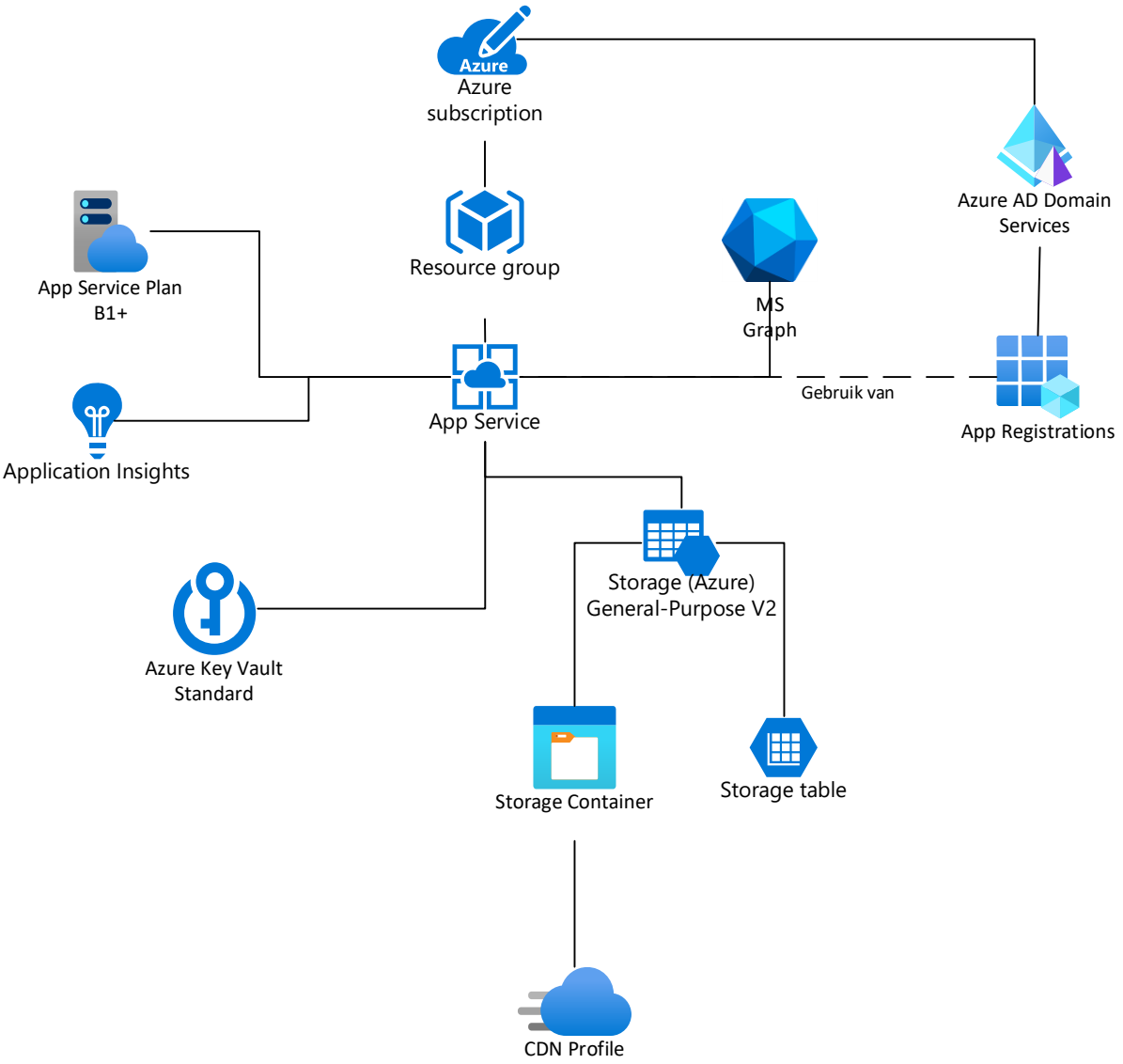
De webservice zal de volgende taken uitvoeren. De uren zijn gebaseerd op onderstaande taken:

- Opslaan nieuwe bestanden en bijbehorende metadata in SharePoint.
- Bijwerken bestanden en bijbehorende metadata in SharePoint
- Nieuwe & Gewijzigde bestanden van SharePoint uploaden naar Azure Storage Container
- Metadata (column info) van bestanden lezen uit SharePoint en opslaan als XML in Azure Storage Container
- Verwijderde bestanden & metadata weggooien uit Azure Storage Container
- Ophalen metadata van een bestand uit SharePoint adhv uniek ID
- Ophalen SharePoint URL voor bestand adhv uniek ID
  Of
- Ophalen content van een bestand uit SharePoint adhv uniek ID

Gebruikte technieken/producten

- Azure Storage Container
- Azure CDN profiles
- Azure Functions en/of Azure App Service (REST webservice in C#)
- Microsoft Graph
- SharePoint Online

Azure Architectuur



Binnen de te gebruiken Azure Subscription zullen twee resource groups gemaakt worden. Deze twee resource groups zullen identiek zijn qua inrichting, waarbij één wordt gebruikt als Test/Acceptatie omgeving en één voor Productie.

Azure App Service

De REST API zal draaien in een Azure App Service. Deze App Service heeft een App Service Plan, voor productie minimaal S1 (afhankelijk van het aantal aanroepen). Voor monitoring wordt gebruik gemaakt van Application Insights, hier kunnen alerts op worden ingesteld zodat beheer een melding krijgt indien een service storing ervaart.

Om te kunnen verbinden met MS Graph wordt gebruik gemaakt van een App Registration. De gegevens hiervan staan in een Azure Key Vault om de beveiliging te waarborgen. Daarnaast zullen andere mogelijke ‘secrets’ ook in deze Key Vault bewaard worden (bv; connectionstrings).

De App Service zal lees en schrijfrechten krijgen op de Storage Container, voor bestanden, en de Storage Table, voor ObjectIds. Dit zal worden gedaan middels een Managed Identity, daarmee is de toegang rechtstreeks gekoppeld aan de lifecycle van de App Service.

Het is mogelijk toegang tot de API af te schermen door specieke IPs te whitelisten of de API Service te koppelen aan een VNet. Daarmee kunnen (kwaadwillende) externe partijen de API niet benaderen. Dit is voornamelijk handig als er geen thuisgebruikers zijn.

De App Service is te benaderen via een custom domain, voor gebruik in oa Unit 4.
Productie: https://content.zeeland.nl
Acceptatie: https://content.acc.zeeland.nl

Azure Storage

In een Azure Storage account zal een container worden gemaakt voor het plaatsen van de publiek toegankelijke bestanden. Deze kunnen rechtstreeks of via de CDN worden opgevraagd.

Om het mogelijk te maken zeer snel bestanden op te zoeken adhv ObjectIds zullen deze, samen met de diverse SharePoint Ids worden opgeslagen in een Table Storage. De PartitionKey is hierbij het ObjectId. Daarnaast zullen alle bestanden in de storage container een index tag krijgen met het ObjectId, daarmee kunnen deze snel worden teruggevonden binnen de container.

Om de kosten laag te houden wordt gebruik gemaakt van Hot en Cool tier binnen de storage container. Bestanden die lange tijd (instelbaar, voorstel: 90 dagen) niet zijn opgevraagd worden automatisch verplaatst naar de Cool tier. Wanneer zo'n bestand toch wordt opgevraagd wordt deze automatisch weer naar de Hot tier verplaatst. Er wordt geen gebruik gemaakt van de Archive tier.

Azure CDN

Door een CDN Profiel te koppelen aan een Storage Account kan een custom domain endpoint beschikbaar worden gesteld. Bestanden kunnen daarmee via een nette URL opgevraagd worden.

Productie: https://contentpub.zeeland.nl
Acceptatie: https://contentpub.acc.zeeland.nl

Hiervoor zijn een geldige SSL-certificaten benodigd in Azure op dit custom domain; een subdomain certificaat. De certificaten worden door IT beheer Provincie Zeeland besteld en aangeleverd. Er wordt gebruikt gemaakt van een Standaard Verizon CDN. Indien dit later niet blijkt te voldoen aan het aantal requests kan deze worden geüpgraded naar Premium.

Azure App Registration

Om te kunnen verbinden met Microsoft Graph en SharePoint Online dient een App Registration gemaakt te worden in de Azure AD omgeving. Deze App Registration dient minimaal de volgende permissions te krijgen:

- Sites.Read.All (Application Permission)
- Files.ReadWrite.All (Application Permission)

Deze rechten worden verleend op de gehele omgeving, waarbij de service alles kan benaderen.

SharePoint Architectuur

Omwille van schaalbaarheid, betrouwbaarheid en snelheid is ervoor gekozen om de data in SharePoint op te slaan op meerdere sitecollecties. Dit is het hoogst mogelijke niveau om onderscheid in aan te brengen. Per rechtensituatie (op basis van de in het functioneel ontwerp beschreven metadata-velden Bron en Classificatie) wordt een sitecollectie aangemaakt. Er wordt gebruik gemaakt van een site template, zodat eenvoudig meerdere sitecollecties aangemaakt kunnen worden (per bron-classificatie-combinatie één). In de beschrijving van de SharePoint architectuur wordt uitgegaan van de naamgeving conform standaard objecttypes van SharePoint, bijvoorbeeld “Document” in plaats van “Bestand”.

Sitecollecties

Elke sitecollectie bevat één documentenbibliotheek en één lijst voor externe referenties. De volgende sitecollecties zijn voor nu voorzien:

- Openbaar
- Intern openbaar
- Inkoopsysteem– Vertrouwelijk
- Financieel systeem- Vertrouwelijk
- Vastgoedsysteem – Vertrouwelijk
- Verkoopfacturensysteem – Vertrouwelijk
- Zaaksysteem - Vertrouwelijk
- Drop Off

N.B. Op dit moment wordt nog geen gebruik gemaakt van objectclassificatie ‘Geheim’, dit is geen onderdeel van de scope. Het technisch ontwerp is zodanig opgezet dat het mogelijk is om deze objectclassificatie in de toekomst toe te voegen.

Handmatige invoer

Voor de selecte groep medewerkers die gebruik kunnen maken van de handmatige invoer functionaliteit wordt een aparte sitecollectie ingericht (nice-to-have) die dient als een zogenaamde ‘Drop Off Library’. Het voordeel hiervan is dat de handmatige invoerder geen schrijfrechten nodig heeft op de verschillende sitecollecties en hij/zij niet hoeft na te denken in welke sitecollectie het bestand geüpload moet worden. De geüploade data wordt vervolgens met behulp van een Flow automatisch gerouteerd naar de juiste sitecollectie adhv ingevoerde metadata (zie § Power Automate stroom).

Inhoudstypes

Het metadatamodel wordt op alle sitecollecties toegepast met behulp van inhoudstypes. Het inhoudstype wordt in de centrale galerie aangemaakt en automatisch gedistribueerd naar de verschillende sitecollecties.

Het inhoudstype wordt aangemaakt met de naam “Publicatieobject” en wordt gebaseerd op het bovenliggende standaard inhoudstype “Document” en is beschikbaar in de documentenbibliotheek. Het inhoudstype “Externe referentie” wordt gebaseerd op het bovenliggende standaard inhoudstype “Item” en is alleen beschikbaar in de lijst Externe referenties.

In onderstaande tabel staat beschreven waar de gewenste metadata gedefinieerd wordt. De metadata velden ('kolommen') uit standaard SharePoint inhoudstype kunnen niet worden aangepast qua type of naamgeving.

Inhoudstype	Bovenliggend inhoudstype	Metadataveld (* = verplicht veld)
Systeem (standaard SP)	-	Type Gemaakt nvt Gewijzigd nvt Gemaakt door nvt Gewijzigd door nvt
Item (standaard SP)	Systeem	Titel *
Document (standaard SP)	Item	Naam *
Publicatieobject	Document	Auteur * Bron gemaakt door Bron datum gemaakt * Bron gewijzigd door Bron datum gewijzigd * Bestandsextensie * Bestandformaat * Bestandssoort * Bewaartermijn * Bron * Classificatie * Datum publicatie * Datum vernietiging * Object ID * Zeester documenttype Zeester kenmerk
Externe Referentie	Item	Applicatie * Referentie * Soort referentie * Document *

Metadataspecificaties

Voor de metadatavelden waarbij een keuze gemaakt dient te worden uit een voorgeschreven lijst, wordt gebruik gemaakt van de centrale SharePoint 'Term Store' (managed metadata). Het voordeel hiervan is dat de keuzewaardes centraal beheerd worden en wijzigingen dus automatisch op alle sitecollecties verwerkt worden en eventuele termwijzigingen ook op reeds getagde content doorgevoerd wordt.

In onderstaande tabel staan de velden die het betreft beschreven met bijbehorende keuzewaardes.

Metadataveld	Keuzewaardes
Bestandssoort	Zie bijlage voorbeeld van de inhoud van de metadatatabel *Per waarde 1 tekstveld waarin "code-omschrijving" wordt geplaatst*
Bron	• Inkoopsysteem • Financieel systeem • Vastgoedsysteem • Verkoopfacturensysteem • Zaaksysteem
Classificatie	• Intern openbaar • Openbaar • Vertrouwelijk
Applicatie	• Diesis Billing • P8 • Proactis • Unit4 Financials • Zaaksysteem.nl

De keuzewaarden voor de hierboven genoemde metadatavelden kunnen door de functioneel applicatiebeheerder via de SharePoint interface worden beheerd, als er meer keuzewaardes bij komen kan de functioneel beheerder deze zelf onderhouden.

Ter referentie staan in onderstaande tabel alle metadatavelden (Kolommen) gespecificeerd.

Weergave naam	Veldnaam	Beschrijving *(wordt getoond)*	Type	Verplicht	Standaardwaarde
Type	DocIcon	-	Systeem	Nvt	Automatisch
Gemaakt	Created	-	Datum	Nvt	Automatisch
Gewijzigd	Modified	-	Datum	Nvt	Automatisch
Gemaakt door	Author	-	Persoon	Nvt	Automatisch
Gewijzigd door	Editor	-	Persoon	Nvt	Automatisch
Naam	LinkFilename	-	Bestand	Ja	Bestandsnaam van het geüploade bestand
Titel	Title	Titel van het publicatieobject	Eén tekstregel	Ja	
Auteur	Auteur	Auteur van het publicatie object, betreft een organisatienaam.	Eén tekstregel	Ja	Provincie Zeeland
Bestandsextensie	BestandsExtensie	Extensie van het bestand.	Éen tekstregel	Ja	pdf

Bestandsformaat	Mimetype	Bestandsformaat van het bestand.	Één tekstregel	Ja	application/pdf
Bestandssoort	BestandsSoort	Soort bestand.	Managed metadata	Ja	
Bewaartermijn	BewaarTermijn	Bewaartermijn van het publicatieobject, in dagen.	Getal	Ja	
Bron	Bron	Verwijzing naar bron van het object.	Managed metadata	Ja	
Classificatie	Classificatie	Classificatie van het publicatieobject conform informatiebeleid.	Managed metadata	Ja	
Bron gemaakt door	BronGemaaktDoor	De naam van degene die het bestand oorspronkelijk heeft gemaakt in het bronsysteem.	Één tekstregel	Nee	
Bron datum gemaakt	BronGemaaktOp	Datum van oorspronkelijke creatie van het bestand in het bronsysteem.	Datum	Ja	
Bron gewijzigd door	BronGewijzigdDoor	De naam van degene die het bestand oorspronkelijk heeft gewijzigd in het bronsysteem.	Één tekstregel	Nee	
Bron datum gewijzigd	BronGewijzigdOp	Datum laatst gewijzigd in het bronysteem.	Datum	Ja	
Datum publicatie	DatumPublicatie	Datum van publicatie (=startdatum bewaartermijn).	Datum	Ja	Huidige datum
Datum vernietiging	DatumVernietiging	Datum vernietiging.	Datum	Ja	Berekend: Datum Publicatie + Bewaartermijn
Object ID	ObjectID	Unieke code voor publicatieobject.	Eén tekstregel	Nee	*(wordt automatisch gevuld)*
Zeester documenttype	ZeesterDocumentType	Oorspronkelijk documenttype in Zeester.	Één tekstregel	Nee	
Zeester kenmerk	ZeesterKenmerk	Oorspronkelijk kenmerk in Zeester.	Eén tekstregel	Nee	
Applicatie	Applicatie	Welke referentie bestaat er met interne applicaties.	Managed metadata	Nee	
Referentie	Referentie	Referentienummer in de gekoppelde applicatie.	Eén tekstregel	Nee	

Soort referentie	SoortReferentie	Wat voor soort referentie is het.	Eén tekstregel	Nee	

Power Automate stroom

Hoewel het grootste gedeelte van de bestanden automatisch geüpload zullen worden richting SharePoint kan het in een klein aantal gevallen voorkomen dat bestanden handmatig geüpload worden. Om de beperkte groep gebruikers niet onnodige rechten/toegang te geven is het voorstel om met een 'Drop Off' documentbibliotheek te gaan werken. Hier worden de bestanden handmatig geüpload en van de benodigde metadata voorzien. Vervolgens kijkt de te maken stroom naar de combinatie van ObjectBron en Objectclassificatie om zo via een te creëren koppeltabel de juiste site/bibliotheek te kunnen selecteren. Deze koppeltabel zal handmatig aangevuld moeten worden wanneer er een nieuw ObjectBron en/of Objectclassificatie ontstaat. Vervolgens wordt er op de doellocatie een nieuw bestand aangemaakt.

Omdat de wens ook is om nieuwe versies van bestanden handmatig te kunnen uploaden zullen er twee extra velden aangemaakt moeten worden in de 'Drop off' bibliotheek. Deze kolommen zullen het objectID en het unieke id van het bestand op de doellocatie bevatten. Met behulp van deze velden kan de stroom in het geval van een nieuwe versie van het bestand het oorspronkelijke bestand op de doellocatie overschrijven.

Stroom stappen:

1. Trigger op de 'Drop Off' documentbibliotheek, wanneer een bestand is aangemaakt of is aangepast;
2. Om ervoor te zorgen dat de stroom niet bij elke aanpassing van een kolom getriggerd wordt zal deze enkel geactiveerd worden wanneer de kolom 'Compleet' de waarde 'ja' heeft;
3. Uitlezen van aangemaakt bestand (+ metadata);
4. Koppeltabel uitlezen met behulp van ObjectBron en Objectclassificatie;
5. API aanroepen om ObjectId te selecteren;
6. Kopiëren bestand naar doelbibliotheek;
7. Metadata uit bronlocatie en ontvangen ObjectId toevoegen aan gekopieerde bestand in doellocatie;
8. Mail naar degene die het bestand aangemaakt heeft met de bevestiging dat deze succesvol is aangemaakt;
9. Nadat het bestand succesvol verwerkt is op de doellocatie krijgt dit document de status 'verwerkt' waardoor deze in de standaard weergave niet zichtbaar is. Wanneer een nieuwe versie van het bestand geüpload wordt dient dit originele bestand overschreven te worden;
10. Als laatste stap in de stroom wordt de waarde uit de kolom 'Compleet' voor het betreffende bestand teruggezet op 'nee'.

Autorisatie

Alle onderdelen van de synchronisatie services maken gebruik van de door Microsoft aangeraden best practices.

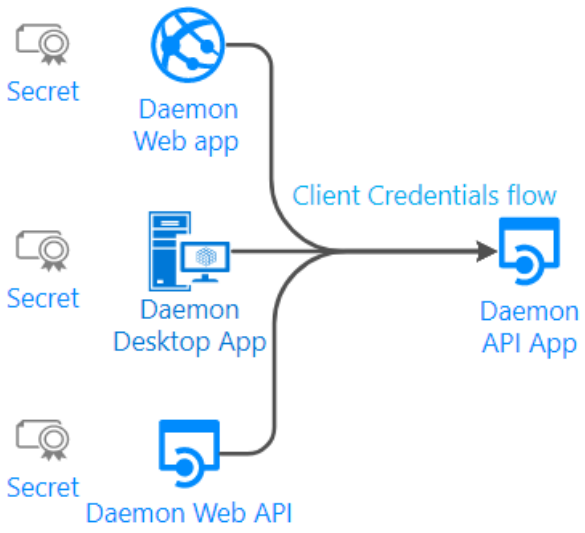
Services onderling

De diverse services zullen onderling authentiseren door middel van een automated Managed Identity, deze wordt toegewezen vanuit de Azure Portal. Elke service in Azure krijgt hiermee een eigen identity waarmee deze toegang kan krijgen tot andere Azure onderdelen. Andere systemen of gebruikers kunnen deze identiteit niet gebruiken.

Om te kunnen verbinden met SharePoint Online wordt gebruik gemaakt van Microsoft Graph. Authenticatie met Graph gebeurt op basis van een App Registration. Via deze App Registration wordt door oAuth verbonden naar Graph en kunnen acties worden uitgevoerd in SharePoint Online.

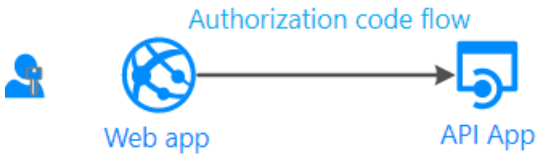
Binnenkomende verzoeken

Alle binnenkomende verzoeken naar de services moeten geauthentiseerd worden via oAuth dmv de App Registration. Externe services zoals BizTalk kunnen aanmelden, via Microsoft, met een App ID en Certificaat (of App Secret). Daarmee kunnen deze externe services zich zonder tussenkomst van persoon authentiseren.



Deze services zullen, na authenticatie, overal toegang tot hebben. Zij mogen elk bestand toevoegen, opvragen of wijzigen.

Om het mogelijk te maken voor eindgebruikers om bestanden op te vragen van de services zal een basale WebUI worden ingericht; deze website doet niets meer dan de gebruiker inloggen in het Azure AD met zijn/haar account en daarna de gevraagde call uitvoeren het opvragen van een bestand of bestand-metadata.



Deze WebUI zal geen vormgeving bevatten; er wordt alleen doorgestuurd naar het bestand in SharePoint, of metadata in JSON-formaat getoond. De enige reden voor deze UI is om de authenticatie voor de gebruiker eenvoudig te houden. Er wordt gebruik gemaakt van MSAL v2 voor het starten en afronden van de authenticatie flow.

SharePoint Online

Het autorisatiemodel zoals beschreven in het functioneel ontwerp wordt in SharePoint ingericht met SharePoint-groepen en (standaard) machtigingsniveaus.

In onderstaande tabel staat beschreven welke groepen welke machtigingen heeft op welke site(s).

SharePoint-groep	Leden	Sitecollectie	Machtigingsniveau
Handmatige invoer	CPS Handmatige invoer	Drop Off	Bijdragen
Raadplegen openbaar	CPS Raadplegen openbaar	Openbaar Intern openbaar	Lezen
Raadplegen Inkoopsysteem – Vertrouwelijk	CPS Raadplegen Inkoopsysteem – Vertrouwelijk	Proacties - Vertrouwelijk	Lezen
Raadplegen Financieel systeem - Vertrouwelijk	CPS Raadplegen Financieel systeem - Vertrouwelijk	Unit4 Financials - Vertrouwelijk	Lezen
Raadplegen Vastgoedsysteem - Vertrouwelijk	CPS Vastgoedsysteem - Vertrouwelijk	P8 Financials - Vertrouwelijk	Lezen
Raadplegen Verkoopfacturensysteem - Vertrouwelijk	CPS Raadplegen Verkoopfacturensysteem - Vertrouwelijk	Diesis Billing - Vertrouwelijk	Lezen
Raadplegen Zaaksysteem - Vertrouwelijk	CPS Raadplegen Zaaksysteem - Vertrouwelijk	Zaaksysteem - Vertrouwelijk	Lezen
Functioneel beheer	CPS Functioneel beheer	alle	Volledig beheer
Technisch beheer	CPS Technisch beheer	alle	Volledig beheer

SharePoint kent de volgende standaard machtigingsniveaus.

Machtigingsniveau	Beschrijving
Volledig beheer	Heeft volledig beheer.
Ontwerpen	Kan weergeven, toevoegen, bijwerken, goedkeuren en aanpassen.
Bewerken	Kan lijsten toevoegen, bewerken en verwijderen; kan lijstitems en bestanden weergeven, toevoegen, bijwerken en verwijderen.
Bijdragen	Kan lijstitems en bestanden weergeven, toevoegen, bijwerken en verwijderen.
Lezen	Kan pagina's, lijstitems en bestanden weergeven en bestanden downloaden.

Indien gewenst kunnen er aangepaste machtigingsniveaus aangemaakt worden (als afgeleide van een van de standaard machtigingsniveaus).

API Services

De API-services zijn in drie delen op te splitsen; import, export en zoeken. Alle endpoints van deze services worden gehost in dezelfde Azure App Service en hebben derhalve dezelfde aanroep URL, basis foutafhandeling en authenticatie methodes. Logging in de gehele app service wordt afgehandeld door Azure App Insights.

De code wordt geschreven in C#, op basis van .Net 6.0. Er wordt zo veel mogelijk gebruik gemaakt van bestaande OpenSource libraries.

De basisstructuur volgt het Generic Repository Pattern; er zijn controllers, repositories en services. Een controller ontvangt een HTTP aanroep, handelt de autorisatie af en vraag aan een repository om de actie(s) uit te voeren. De repository omvat alle logica en voert deze uit, mogelijk met behulp van services (bv voor File Storage in Azure). Hiermee wordt een nette 'seperation of concern', een scheiding van technische onderdelen, gehouden en kunnen 'units of work', diverse technische onderdelen, eenvoudig worden toegevoegd of aangepast.

AppSettings.json Settings

De volgende instellingen worden opgeslagen in de AppSettings.json, deze waarden kunnen worden overschreven via de App Service Configuration in de Azure Portal.

Naam	Type	Omschrijving
ClientID	GUID	ID van de AppRegistration
CertificateThumbprint	string	Thumbprint van AppRegistration certificaat
TenantID	GUID	GUID van Azure Tenant
ObjectIdsTable	string	Naam van Azure Storage Container
AppSettingsTable	string	Naam van Azure Storage Table
StorageTableConnectionString	string	Koppeling naar connectionstring in key vault
FileStorageConnectionString	string	Koppeling naar connectionstring in key vault
CallbackURL	string	URL voor de service na CDN update
LocationMapping	JSON	Mapping tussen classificatie + bron en SharePoint locatie
MetadataMapping	JSON	Mapping tussen metadata velden en SharePoint kolommen + default waarde
[…]		

Azure Storage Table 'Settings'

Sommige instellingen moeten worden aangepast door de applicatie, deze zullen worden opgeslagen in een Azure Storage Table. Hierbij wordt een uniek ID gegenereerd als PartitionKey en zijn de namen van de instellingen ook de RowKeys.

Naam	Type	Omschrijving
SequenceNumber	int	Volgnummer voor ObjectId
LastSynchronisationNew	datetime	Laatste synchronisatie nieuwe documenten
LastSynchronisationChanged	datetime	Laatste synchronisatie gewijzigde documenten
LastSynchronisationDeleted	datetime	Laatste synchronisatie verwijderde documenten
[…]		

Import

Nieuwe bestanden en metadata uploaden naar SharePoint Online

Kleine bestanden kunnen gecombineerd worden toegevoegd via

URL	Input	Output
[PUT] /files/	{ "Content": "base64 byte[]", "Metadata": { "MimeType": "string", "FileName": "string", "FileExtension": "string", "SourceCreatedOn": "datetime: iso8610", *"SourceCreatedBy": "string: displayname",* "SourceModifiedOn": "datetime: iso8610", *"SourceModifiedBy": "string: displayname",* "AdditionalMetadata": { "Author": "string", "Title": "string", "DocumentType": "string:enum", *"ZeesterDocumentType":"string",* *"ZeesterReference": "string",* "RetentionPeriod": "int", "Classification": "string: enum", "PublicationDate": "datetime: iso8610", "ArchiveDate": "datetime: iso8610", "Source": "string" }, *"ExternalReferences": [{* *"ExternalApplication": "string",* *"ExternalReference": "string",* *"ExternalReferenceType": "string: enum"* *}]* } } Schuingedrukte waardes zijn niet verplicht.	**Bestand is correct toegevoegd** 200: {ObjectId} **Gebruiker heeft zich niet aangemeld** 401: Unauthorized **Gebruiker heeft geen toegang** 403: Forbidden **Server error** 500: {error}

Grote bestanden (> 1GB) moeten apart worden geüpload door eerst de volledige content te uploaden. Op deze wijze kan de verzender gebruik maken van zgn 'multipart/form-data', waarmee meer data verzonden kan worden dan rechtstreeks in de body via het bovenstaande endpoint. Na succesvol uploaden kan de metadata worden toegevoegd via het endpoint voor wijzigen metadata.

URL	Input	Output
[PUT] /files/new/{source}/{classification}	"<contents of file>"	**Bestand is correct geupload** 200: {ObjectId} **Gebruiker heeft zich niet aangemeld** 401: Unauthorized

		Gebruiker heeft geen toegang 403: Forbidden **Server error** 500: {error}

Wanneer een request binnenkomt wordt de content tijdelijk opgeslagen op de schijf van de AppService (in geval van grote bestanden) of in het geheugen van de App Service (in geval van kleine bestanden). Via de Microsoft Graph wordt vervolgens een nieuw bestand gemaakt met de ontvangen content in de gewenste locatie. Deze locatie wordt bepaald adhv de classificatie van het bestand, bij grote bestanden is het dus van belang dat deze wordt meegegeven in de URL. Via de mapping uit AppSettings.json wordt de gewenste locatie opgezocht.

De ontvangen metadata wordt gemapped naar SharePoint kolommen adhv de Mapping in de AppSettings.json. Als dit mislukt, bijvoorbeeld omdat een waarde niet gemapped kan worden of omdat een verplichte waarde mist, wordt een 500 foutmelding geretourneerd. Deze foutmelding bevat een logisch Engelse foutbericht met de velden welke niet gemapped konden worden. Deze foutmelding wordt ook geregistreerd in de App Insights.

Wanneer beide stappen succesvol zijn doorlopen wordt een ObjectId gegenereerd via de ObjectIdGenerator code (zie ObjectId endpoint). Deze wordt samen met de eventueel ontvangen metadata opgeslagen bij het nieuwe gemaakte bestand in de SharePoint bibliotheek via Microsoft Graph.

Als er geen metadata mee is verstuurd, in geval van grote bestanden via '/files/new', wordt default placeholder metadata opgeslagen (waardes uit AppSettings.json). De verzender kan via het 'update metadata' endpoint deze metadata alsnog koppelen aan het bestand.

Als een van de stappen mislukt wordt het bestand uit SharePoint verwijderd (indien al geüpload) via Microsoft Graph en retourneert de server een 500 foutmelding met logisch Engelse foutbericht. Deze foutmelding wordt ook geregistreerd in de App Insights.

Gewijzigde bestanden uploaden naar SharePoint Online

URL	Input	Output
[POST] **/files/content/{ObjectId}** **/files/{ObjectId}/content**	"<contents of file>"	**Bestand is correct geüpdatet** 200: {ObjectId} **Gebruiker heeft zich niet aangemeld** 401: Unauthorized **Gebruiker heeft geen toegang** 403: Forbidden **ObjectId onbekend**

		404: Not found **Server error** 500: {error}

Wanneer een request binnenkomt wordt adhv het meegegeven ObjectId de SharePoint Ids van het bestand opgezocht in de table storage. Vervolgens wordt het bestand opgevraagd via Microsoft Graph. Als een van deze twee stappen het bestand niet kan vinden wordt een 404 foutmelding geretourneerd. Er wordt niet gecontroleerd of de meegegeven content anders is, deze wordt altijd opgeslagen als nieuwe versie van het bestand.

Wanneer het bestand is gevonden wordt de content tijdelijk opgeslagen op de schijf van de AppService (in geval van grote bestanden) of in het geheugen van de App Service (in geval van kleine bestanden). Via de Microsoft Graph wordt vervolgens een nieuwe versie van bestand gemaakt met de ontvangen content. Als deze stap mislukt retourneert de server een 500 foutmelding met logisch Engelse foutbericht. Deze foutmelding wordt ook geregistreerd in de App Insights.

Gewijzigde metadata opslaan in SharePoint Online

URL	Input	Output
[POST] **/files/metadata/{ObjectId}** **/files/{ObjectId}/metadata**	{ "Metadata": { "MimeType": "string", "FileName": "string", "FileExtension": "string", "SourceCreatedOn": "datetime: iso8610", *"SourceCreatedBy": "string: displayname",* "SourceModifiedOn": "datetime: iso8610", *"SourceModifiedBy": "string: displayname",* "AdditionalMetadata": { "Author": "string", "Title": "string", "DocumentType": "string:enum", *"ZeesterDocumentType":"string",* *"ZeesterReference": "string",* "RetentionPeriod": "int", "Classification": "string: enum", "PublicationDate": "datetime: iso8610", "ArchiveDate": "datetime: iso8610", "Source": "string" }, *"ExternalReferences": [{* *"ExternalApplication": "string",* *"ExternalReference": "string",* *"ExternalReferenceType": "string: enum"* *}]* } } Schuingedrukte waardes zijn niet verplicht.	**Metadata is correct geüpdatet** 200: {ObjectId} **Gebruiker heeft zich niet aangemeld** 401: Unauthorized **Gebruiker heeft geen toegang** 403: Forbidden **ObjectId onbekend** 404: Not found **Server error** 500: {error}

Wanneer een request binnenkomt wordt adhv het meegegeven ObjectId de SharePoint Ids van het bestand opgezocht in de table storage. Vervolgens wordt het bestand opgevraagd via Microsoft Graph. Als een van deze twee stappen het bestand niet kan vinden wordt een 404 foutmelding geretourneerd. Er wordt niet gecontroleerd of de meegegeven content anders is, deze wordt altijd opgeslagen als nieuwe metadata van het bestand.

Wanneer het bestand is gevonden wordt voor alle meegegeven metadata de waarde gemapped naar de bijbehorende SharePoint kolommen adhv de Mapping in de AppSettings.json. Vervolgens worden deze kolommen geüpdatet via Microsoft Graph.

Als een waarde niet is meegegeven wordt deze niet geüpdatet. Als een waarde leeg moet worden gemaakt moet deze expliciet als lege string / datetime min / 0 waarde (dus niet NULL) worden meegegeven. Als een van de verplichte waarde een lege string bevat wordt een foutmelding gegeven. Als er geen metadata is meegegeven wordt er niets aangepast.

Genereren ObjectId voor nieuw (SharePoint) bestand

URL	Input	Output
[PUT] **/files/ObjectId/**	{ "SiteId": "guid", "ListId": "guid", "ItemId": "int" }	**ObjectId is correct gemaakt** 200: {ObjectId} **Gebruiker heeft zich niet aangemeld** 401: Unauthorized **Gebruiker heeft geen toegang** 403: Forbidden **Server error** 500: {error}

Als een nieuw bestand is aangemaakt in SharePoint wordt dit endpoint aangeroepen met de SharePoint Ids van het nieuwe bestand. Wanneer een bestand rechtstreeks is toegevoegd aan SharePoint start een PowerAutomate Flow welke dit endpoint aanroept. Wanneer een bestand wordt toegevoegd via het REST endpoint wordt dit endpoint niet aangeroepen maar wordt de onderliggende code rechtstreeks uitgevoerd, aangezien beide endpoints in dezelfde service zitten.

Het ObjectId heeft het format *ZLD{jaar}-{volgnummer}*, waarbij volgnummer wordt opgehoogd voor elk ontvangen bestand. Dit volgnummer wordt opgeslagen in de Azure Storage Table voor settings. Daarnaast wordt elk ontvangen bestand opgeslagen in de Azure Storage Table met ObjectId en de ontvangen SharePoint Ids. Voor het ontvangen bestand wordt ook de DriveId en DriveItemId opgehaald via Microsoft Graph en opgeslagen voor verder gebruik in het systeem.

Het nieuwe volgnummer wordt ook opgeslagen in de Azure Storage Table ‘settings’.

Azure Storage Table - ObjectIdentifiers

Kolom	Type	Opmerkingen

ObjectId	ZLD{jaar}-{volgnummer}	PartitionKey
SiteId	string	Rowkey Combinatie is uniek
ListId	String	
ItemId	String	
DriveId	String	Combinatie is uniek
DriveItemId	string	

Zoeken

Vinden bestand adhv ObjectId

URL	Input	Output
[GET] **/files/content/{ObjectId}** **/files/{ObjectId}/content**		**Bestand is gevonden** 200: {SharePoint url} OF {content of file} **Gebruiker heeft zich niet aangemeld** 401: Unauthorized **Gebruiker heeft geen toegang** 403: Forbidden **ObjectId onbekend** **Bestand niet gevonden in SharePoint** 404: Not found **Server error** 500: {error}

Wanneer een request binnenkomt wordt adhv het meegegeven ObjectId de SharePoint Ids van het bestand opgezocht in de storage table. Vervolgens wordt het bestand opgevraagd via Microsoft Graph. Als een van deze twee stappen het bestand niet kan vinden wordt een 404 foutmelding geretourneerd.

Versie 1:
Wanneer het bestand is gevonden wordt de SharePoint absoluut server URL van het bestand geretourneerd.

Versie 2:
Wanneer het bestand is gevonden wordt de volledige content ingelezen en vervolgens geretourneerd naar de aanvrager.
Deze versie wordt alleen gemaakt wanneer Versie 1 niet voldoende blijkt te zijn voor Unit4.

Vinden metadata adhv ObjectId

URL	Input	Output
[GET] **/files/metadata/{ObjectId}** **/files/{ObjectId}/metadata**		**Bestand is gevonden** 200: "ObjectId": "string", { "Metadata": { "MimeType": "string",

```
  "FileName": "string",
  "FileExtension": "string",
  "CreatedOn": "datetime: iso8610",
  "CreatedBy": "string: displayname",
  "ModifiedOn": "datetime: iso8610",
  "ModifiedBy": "string: displayname",
  "SourceCreatedOn": "datetime: iso8610",
  "SourceCreatedBy": "string: displayname",
  "SourceModifiedOn": "datetime: iso8610",
  "SourceModifiedBy": "string: displayname",
  "AdditionalMetadata": {
    "Author": "string",
    "Title": "string",
    "DocumentType": "string",
    "ZeesterDocumentType":"string",
   "ZeesterReference": "string",
    "RetentionPeriod": "int",
    "Classification": "string: enum",
    "PublicationDate": "datetime: iso8610",
    "ArchiveDate": "datetime: iso8610",
    "Source": "string"

  },
  "ExternalReferences": [{
    "ExternalApplication": "string",
    "ExternalReference": "string",
    "ExternalReferenceType": "string: enum"
  }]
 }
}
```


Gebruiker heeft zich niet aangemeld
401: Unauthorized

Gebruiker heeft geen toegang
403: Forbidden

ObjectId onbekend
Bestand niet gevonden in SharePoint
404: Not found

Server error
500: {error}

Wanneer een request binnenkomt wordt adhv het meegegeven ObjectId de SharePoint Ids van het bestand opgezocht in de storage table. Vervolgens wordt het bestand opgevraagd via Microsoft Graph. Als een van deze twee stappen het bestand niet kan vinden wordt een 404 foutmelding geretourneerd.

Wanneer het bestand is gevonden wordt de metadata van SharePoint gemapped naar de gewenste metadata van het endpoint via de Mapping uit de AppSettings.json. Dit metadata object wordt geretourneerd naar de aanvrager van het endpoint.

Export

Om bestanden uit SharePoint te synchroniseren naar de Storage Container wordt periodiek een WebJob gestart adhv een schedule. Deze schedule kan in Azure worden aangepast. Deze job zal de drie onderstaande endpoints aanroepen. Deze endpoints kunnen ook manueel worden gestart indien wenselijk. Nadat de synchronisatie succesvol is afgerond wordt de synchronisatie datum van de betreffende actie in de Azure Storage Table voor settings aangepast naar de datetime van 'nu'.

Mocht halverwege de synchronisatie iets fout gaan waardoor deze geheel afgebroken wordt, wordt de synchronisatie datum ingesteld op het laatst succesvol gesynchroniseerde bestand. Daarmee kan de synchronisatie op het juiste punt worden hervat als deze nogmaals wordt gestart. Indien een bestand/metadata faalt, wordt dit geregistreerd in de App Insights logging en loopt de synchronisatie verder met het volgende bestand.

XML-format metadata

```
<?xml version="1.0"?>
 <Document id="ObjectId">
  <MimeType>application/pdf</MimeType>
  <FileName>filename.pdf</FileName>
  <FileExtension>pdf</FileExtension>
  <CreatedOn>yyyy-MM-ddThh:mm:ssZ</CreatedOn>
  <ModifiedOn>yyyy-MM-ddThh:mm:ssZ</ModifiedOn>
  <SourceCreatedOn>yyyy-MM-ddThh:mm:ssZ</SourceCreatedOn>
  <SourceModifiedOn>yyyy-MM-ddThh:mm:ssZ</SourcemodifiedOn>
  <Title>Document titel</Title>
  <Author>Provincie Zeeland</Author>
  <DocumentType>Type</DocumentType>
  <ZeesterDocumentType>ZeesterType</ZeesterDocumentType>
  <ZeesterReference><ZeesterReference>
  <RetentionPeriod>90</RetentionPeriod>
  <Classification>Openbaar</Classification>
  <PublicationDate>yyyy-MM-ddThh:mm:ssZ</PublicationDate>
  <ArchiveDate>yyyy-MM-ddThh:mm:ssZ</ArchiveDate>
 </Document>
</xml>
```

Nieuwe bestanden en metadata uploaden naar Azure Storage Container

URL	Input	Output
[GET] **/export/new**		**Synchronisatie succesvol** 200: OK **Server error** 500: {error}

Via Microsoft Graph worden alle nieuwe bestanden sinds de laatste synchronisatie opgehaald. De

metadata wordt opgeslagen in XML formaat en de content van het bestand wordt gekopieerd naar de Storage Container.

Indien er geen metadata bekend is voor een document wordt deze niet meegenomen met de synchronisatie, er wordt dan vanuit gegaan dat het document nieuw is of dat iets is fout gegaan bij het toevoegen van dit document. Wanneer de metadata is aangevuld zal het document naar voren komen bij gewijzigde documenten en alsnog geüpload naar de CDN.

Per bestand wordt de CallbackURL, uit de AppSettings.json, aangeroepen om de service te laten weten dat het bestand is toegevoegd.

Het bestand krijgt in de Storage Container de naam **'{ObjectId}.{filename}'**.
Metadata behorende bij een bestand heeft de naam **'{ObjectId}.{filename}.xml'**
Het ObjectId wordt toegevoegd in de naam om te zorgen dat elk bestand een unieke naam in de container heeft.

Gewijzigde bestanden & metadata uploaden naar Azure Storage Container

URL	Input	Output
[GET] **/export/updated**		**Synchronisatie succesvol** 200: OK **Server error** 500: {error}

Via Microsoft Graph worden alle gewijzigde bestanden, content of metadata, sinds de laatste synchronisatie opgehaald. De metadata wordt opgeslagen in XML-formaat en de content van het bestand wordt gekopieerd naar de Storage Container, bestaande bestanden in de Storage Container worden hiermee overschreven.

Indien er geen metadata bekend is voor een document wordt deze niet meegenomen met de synchronisatie, er wordt dan vanuit gegaan dat het document nieuw is of dat iets is fout gegaan bij het toevoegen van dit document. Wanneer de metadata is aangevuld zal het document weer naar voren komen bij gewijzigde documenten en alsnog geüpload naar de CDN.

Per bestand wordt de CallbackURL, uit de AppSettings.json, aangeroepen om de service te laten weten dat het bestand is gewijzigd.

Verwijderde bestanden en metadata in Azure Storage Container weggooien

URL	Input	Output
[GET] **/export/deleted**		**Synchronisatie succesvol** 200: OK **Server error** 500: {error}

Via Microsoft Graph worden alle verwijderde bestanden sinds de laatste synchronisatie opgehaald.

Voor deze bestanden wordt zowel de content als de xml metadata verwijderd en wordt de CallbackURL, uit de AppSettings.json, aangeroepen om de service te laten weten dat het bestand

verwijderd is.
Deze CallbackURL is de url naar de Elastic Search Connector op https://es-connector.zeeland.nl.

URL	Input	Output
[POST] **https://es-connector.zeeland.nl** **/{create|update|delete}/{ObjectId}**	{ "ObjectId": "string", { "Metadata": { "MimeType": "string", "FileName": "string", "FileExtension": "string", "CreatedOn": "datetime: iso8610", "ModifiedOn": "datetime: iso8610", "SourceCreatedOn": "datetime: iso8610", "SourceModifiedOn": "datetime: iso8610", "AdditionalMetadata": { "Author": "string", "Title": "string", "DocumentType": "string", "ZeesterDocumentType": "string", "ZeesterReference": "string", "RetentionPeriod": "int", "Classification": "string: enum", "PublicationDate": "datetime: iso8610", "ArchiveDate": "datetime: iso8610", } } } }	**Update succesvol** 200: OK **Geen token** 401: Unauthorized **Ongeldige token** 403: Forbidden **ObjectId onbekend / Bestand niet gevonden in Azure storage container** 404: Not found **Server error** 500: {error}

De body van de aanroep bevat de metadata in JSON format, gelijk aan de ontvangen JSON bij nieuw/wijzigen bestand maar dan exclusief de metadata die niet wordt meegestuurd bij synchronisatie services. In geval van verwijderen wordt er *geen* body meegestuurd.

Elastic Search + connector wordt gerealiseerd als een losstaand onderdeel gebaseerd op Docker containers binnen een K8s omgeving (eventueel gehost binnen de Azure stack).

De aanroep van de service zal dan ook als beveiliging geen gebruik maken van Azure / Microsoft identies maar van een token op basis van de zgn Bearer authentication, zie: https://swagger.io/docs/specification/authentication/bearer-authentication/.

De token is niet tijd gebonden en hoeft dus niet na een bepaalde tijdspanne ververst te worden. Gezien deze aanroep alleen vanuit de interne omgeving zal worden aangeroepen is dit geen probleem.

Naslagwerk

Pricing Overview—How Azure Pricing Works | Microsoft Azure

Azure Key Vault Overview - Azure Key Vault | Microsoft Learn

Introduction to Table storage - Object storage in Azure | Microsoft Learn

About Blob (object) storage - Azure Storage | Microsoft Learn

Map a custom domain to an Azure Blob Storage endpoint - Azure Storage | Microsoft Learn

What is a content delivery network (CDN)? - Azure | Microsoft Learn

Overview - Azure App Service | Microsoft Learn

Set a blob's access tier - Azure Storage | Microsoft Learn

Authentication and authorization - Azure App Service | Microsoft Learn

Quickstart: Register an app in the Microsoft identity platform - Microsoft Entra | Microsoft Learn

Verify scopes and app roles protected web API - Microsoft Entra | Microsoft Learn

driveItem: createUploadSession - Microsoft Graph v1.0 | Microsoft Learn

MIME types (IANA media types) - HTTP | MDN (mozilla.org)

Upload files in ASP.NET Core | Microsoft Learn

Bijlage: Inhoud Metadatatabel Bestandssoort

BIJ	Bijlage van een e-mail	D	YY000001	YY999999	REG
BIJ-BRIEF	Bijlage van brief/nota	D	YY000001	YY999999	REG
EMAIL	E-mail	D	YY000001	YY999999	REG
INT	Intern document	D	YY000001	YY999999	REG

FEZ-AMR	FEZ Bankafschriften AMRO	D	CA$Myy00001	CA$Myy99999	FAC
FEZ-BGR	FEZ Begroting	D	CBRYY00001	CBRYY99999	FAC
FEZ-BNG	FEZ Bankafschriften BNG	D	CBGyy00001	CBGyy99999	FAC
FEZ-BNGKKS	FEZ BNG Mobilteitsfonds KKS Bankafschrif	D	CKKSYY0001	CKKSYY9999	FAC
FEZ-BNK	FEZ Bankafschriften	D	BYY0000001	BYY9999999	FAC
FEZ-CDEX	FEZ Bankafschriften Dexia	D	CEXYY0001	CEXYY9999	FAC
FEZ-CDHWB	FEZ Bankafschriften Dexia Huurwaarborg	D	CHWBYY0001	CHWBYY9999	FAC
FEZ-CFORT	FEZ Bankafschriften Fortis	D	CFORTYY001	CFORTYY999	FAC
FEZ-CKILG	FEZBankafschriften Kredietfaciliteit ILG	D	CKILGYY001	CKILGYY999	FAC

FEZ-CVISA	FEZ Bankafschriften VISA	D	CVISAYY001	CVISAYY999	FAC
FEZ-EHS	FEZ Groenfonds EHS Bankafschriften	D	CEHSYY0001	CEHSYY0012	FAC
FEZ-FAC	FEZ CODAFACTUUR Externe Organisatie	D	CEyy000001	CEyy999999	FAC
FEZ-ILG	FEZ Groenfonds ILG Bankafschriften	D	CBILGYY001	CBILGYY999	FAC
FEZ-NCW	FEZ Groenfonds NCW Bankafschriften	D	CNCWYY0001	CNCWYY9999	FAC

INK	Inkomende Documenten	D	YY000001	YY999999	

INT	Intern document	D	YY000001	YY999999	REG

UIT	Uitgaande Post	D	YY000001	YY999999	

zee	ZEESTER registratie	D	ZEEyy00001	ZEEyy99999	r-zee
ZLD-FAC	CODAFACTUUR	D	CFYY000001	CFYY999999	FAC
ZLD-VPL	CODA Verplichting	D	CVYY000001	CVYY999999	FAC

Fin-FBEW	Financieel bewijsstuk (VPL)	D	CBYY000001	CBYY999999	FAC

EBIL	E-Billing: VK-ZLD-ALG	H	Z184000	Z185999	EBIL
EBIL-PVCR	E-Billing: VK-PVE-ALG-CR	H	183000	183999	EBIL
EBIL-SBOPV	E-Billing: VK-PVE-ALG	H	P181000	P182999	EBIL
EBILCR	E-Billing: VK-ZLD-ALG-CR	H	CR186000	CR186999	EBIL
PV-ZLD	E-Billing: VK-Personeelsvereniging	H	P170001	P179999	EBIL