**Hoe moet de token worden opgehaald?**

Om een token op te vragen moet er een http POST van het type ”x-www-form-urlencoded” gedaan worden naar:

Voor non productie omgeving zoal bv. Acceptatie/Test:

<https://keycloak.platformdst.nl/auth/realms/DST-NonProd/protocol/openid-connect/token>

Productie:

<https://keycloak.platformdst.nl/auth/realms/DST/protocol/openid-connect/token>

Met de volgende parameters in:

* username = [je eigen gebruikersnaam]
* password = [je eigen wachtwoord]
* client\_id = dst3-api
* grant\_type = password

Een voorbeeld van een dergelijke aanroep in CURL:

curl --location --request POST '[https://keycloak.platformdst.nl/auth/realms/](https://keycloak.platformdst.nl/auth/realms/%22%20%5Co%20%22https%3A//keycloak.platformdst.nl/auth/realms/)DST-NonProd/protocol/openid-connect/token' \

--header 'Content-Type: application/x-www-form-urlencoded' \

--data-urlencode 'username=[replace with your username]' \

--data-urlencode 'password=[replace with your password]' \

--data-urlencode 'client\_id=dst3-api' \

--data-urlencode 'grant\_type=password'

In geval van een positief resultaat bevat de JSON een “access\_token” welke gebruikt moet worden om de DST 3.0 API aan te roepen.

Vervolgens kan er met dit token een aanroep gedaan worden naar de DST 3.0 API.

**De swagger:**

De API 3.0 OpenAPI definitie documentie voor de test en productie omgeving zijn te vinden op:

Test:

<https://dst3-api.test.platformdst.nl/api/docs/swagger-ui/index.html>

Productie:

<https://dst3-api.platformdst.nl/api/docs/swagger-ui/index.html>

Er zijn verschillende OpenAPI definities beschikbaar, voor elke rol afzonderlijk. Deze zijn hieronder weergegeven en dienen in het “explore” veld te worden ingevoerd om de juiste docs te tonen. Deze zijn alleen toegankelijk voor gebruikers met de juiste rol.

/api/docs/supplier

**Hoe post je een product?**

Een voorbeeld van een aanroep om producten te posten in CURL is hieronder weergegeven:

Eerst moet je een token ophalen zoals eerder beschreven daarna kan je de volgende call uitvoeren

curl --location --request POST '[https://dst3-api.test.platformdst.nl/api/](https://dst3-api.test.platformdst.nl/api/%22%20%5Co%20%22https%3A//dst3-api.test.platformdst.nl/api/)supplier/product' \

--header 'Authorization: Bearer [replace with the access token value from previous token request]'

In de body moet je een product stoppen in json format.

Mocht er iets fout zijn word aangegeven welke validatie fouten de artikelstandaart hanteerd.

**De overige endpoints:**

**Product endpoints:**

GET /api/supplier/product
Hier krijg je alle producten van jou als supplier

Je kan een pageble meegeven. Default is deze 20 per pagina

GET /api/supplier/product-update
Hier krijg je alle responses van verkeerd aangeleverde artikelen en de validatie fouten per artikel

Je kan een pageble meegeven. Default is deze 20 per pagina

GET /api/supplier/product-update/{id}

Stuur hier de id mee van de specifieke product-update die je wilt bekijken

Hier krijg je de fout(en) van een specifieke aanlevering

Als je als aanlevert via CSV zijn de komende endpoints relevant anders niet:

GET /api/supplier/importjob

Hier krijg je alle imports terug per bestand op de FTP server

GET /api/supplier/importjob/{id}

Stuur hier de id mee van de specifieke importjob die je wilt bekijken

Hier krijg je de specificaties van een import terug

GET /api/supplier/importjob/{id}/error-details

Stuur hier de id mee van de specifieke importjob die je wilt bekijken

Hier krijg je de validatie meldingen van een import terug

GET /api/supplier/product-update/importjob/{importJobId}

Stuur hier de id mee van de specifieke importjob die je wilt bekijken

Je krijgt alle product-updates terug die bij deze import horen.

**Supplier gegevens endpoints:**

Alle endpoints in deze catogorie kunnen ook in de UI van DST 3.0 worden uitgevoerd

GET /api/supplier/get

Hier krijg je je eigen gegevens terug zoals opgeslagen in DST

PUT /api/supplier/update

Hier kan je je gegevens updaten. Stuur in de body de geüpdatete gegevens mee

GET /api/supplier/ftpsettings

Hier krijg je de ftp-settings terug mocht je deze hebben

PUT /api/supplier/ftpsettings

Hier kan je je ftp-gegevens updaten. Stuur in de body de geüpdatete ftp-gegevens mee

**Profile endpoints**

Alle calls kan je ook uitvoeren in de UI

GET /api/supplier/profile/all

Hier krijg je al je profielen terug

GET /api/supplier/profile/{supplierProfileId}

Hier krijg je een specifiek profiel terug. De ID die je meestuurt in het pad bepaald welk profiel je te zien krijgt

GET /api/supplier/profile/all/{customergln}

Hier krijg je al je profielen terug waar deze customer van geautoriseerd is. De Id die je meestuurt bepaald op basis van welke customer je de profielen te zien krijgt

Dit kan niet in de UI

POST /api/supplier/profile/create

Hier moet je in de body de gegevens van een nieuw profiel mee sturen:

{

 "code": "string",

 "name": "string",

 "supplierAuthorization": "PROFILE\_BASED",

 "archived": true

}

PUT /api/supplier/profile/update/{id}

Hier moet je in de body precies hetzelfde sturen als hierboven alleen moet er nu in de url de id worden mee gestuurd van het profiel wat je wilt updaten

PUT /api/supplier/profile/authorization/{customerGln}/supplier-profile/{supplierProfileId}

Hier kan je een customer autoriseren voor een profiel. De Customer wordt meegegeven doormiddel van de GLN en het profiel op basis van de id van het profiel

PUT /api/supplier/profile/authorization/{customerGln}/supplier-profile/remove/{supplierProfileId}

Hier kan je een autorisatie verwijderen op basis van dezelfde functionaliteiten als hierboven

**Codelist endpoints**

Alle endpoints in deze catogorie kunnen ook in de UI van DST 3.0 worden uitgevoerd

GET /api/supplier/codelists

Hier krijg je een overzicht van alle codelijsten die er in DST 3.0 zitten

GET /api/supplier/codelist/{codelistCode}

Als je de code gebruikt kan de codelijst ophalen en krijg je de waarden van deze codelijst terug