**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/customer

**Hoe krijg je producten?**

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

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

curl --location --request GET '[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/)customer/product' \

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

Voorbeeld:

<https://dst3-api.platformdst.nl/api/customer/product?language=en>

Hier wordt de taal als Engels opgegeven

**De overige endpoints:**

Om de endpoints te berijken moet je eerst de basis url voor de volgende aanroepen uit te voeren met in de header een geldige token zoals hierboven is voor gedaan.

/api/customer/product/count

Dit geeft als resultaat het getal van het aantal producten die jij mag zien en aan de ingevulde zoektermen voldoen.

De zoek termen zijn:

Language: de taal van het product,

supplierName: de code a supplier heeft bijvoorbeeld TREK,

countries: de landen vanuit het product je wilt zien. *Een array van strings*,

eanUpc: het ean upc nummer in het product,

itemNumber: het item number veld in het product,

statuses: de statussen waar het product in zit. *Een array van strings,*

productGroup: de artikelsubgroep tot waar het product toe behoord,

brand: het merk wat je vind in de general\_productinformation,

keyword: het keyword veld wat in het product zit,

dateModified: de datum waarop het product als laatst in aangepast,

modifiedSince: de datum sinds het product als laatst in aangepast.

Alle input velden zijn niet case sensitive.

Al deze zoek termen zijn ook mogelijk in het product endpoint van hierboven.

/api/customer/codelists

Hier krijg je de code en naam terug van alle codelijsten die in DST aanwezig zijn.

/api/customer/codelist/{codelistCode}

Met de meegestuurde codelistCode krijg je een array terug van de waardes die in die codelijst tabel zitten

**De multicustomer endpoints:**

/api/customer/product/clean

Deze werkt precies hetzelfde als het endpoint: /api/customer/product

Met 2 kleine aanpassingen:

1: De codelijst waardes in de versie van de internationalCode blijven staan en niet per taal gekeken wordt welke vertaling ernaast gezet

2: er word niet gekeken naar de autorisatie omdat het hier gaan om een multicustomer

/api/customer/authorization/gln

Als je de GLN in vult van een customer krijg je de autorisatie terug op de volgende manier:

Een array van:

gln,

supplierCode,

supplierName,

profileCode,

profileName,

customerCode,

customerName

/api/customer/all

Hier hoeft geen informatie mee gegeven te worden en je krijgt een array terug van alle customers