

Vendit Public API

Enkele zaken die niet direct duidelijk zijn bij het gebruik van de Public API

Find

De meest belangrijke entiteiten hebben een Find entry point. Met name (deze lijst kan in de toekomst uitgebreid worden):

- Customers/Find
- Offers/Find
- Orders/Find
- Products/Find
- PurchaseOrders/Find
- Repairs/Find
- Suppliers/Find
- TradeIns/Find
- Transactions/Find

Deze hebben allemaal een generieke werking. In de swagger documentatie is de bruikbare informatie helaas een beetje verborgen waardoor de werking in eerste opzicht vaag lijkt.

Om de beschikbare velden te zien moet er in Swagger op "Schema" geklikt worden naast "Example Value". Daarna moeten er nog een aantal nodes van een soort treeview open geklikt worden om daadwerkelijk bruikbare informatie te krijgen. Zie de rode circels in dit voorbeeld van Orders:

Example Value Schema

description: Collection of criteria filters

fieldFilters

nullable: true
List of criteria, note that the weight (cost) of a call increases with each added criteria Filter

CustomerOrderHeaderCustomerOrderHeaderCollectionInt32OrderFieldsFilter

description: Filter criteria for a search (Find) operation

field

integer(\$int32)
example: 7
Members

Name	Value	Description
CompanyName	101	Type = String
CustomerNumber	103	Type = String
Email	104	Type = String
AddressTypeDescription	105	Type = String
GenderDescription	106	Type = String
MobilePhoneNumber	107	Type = String
LastLinePhoneNumber	108	Type = String

Op de volgende pagina is de volledige schema weergegeven (gebruik deze niet als referentie, er zullen in de toekomst velden worden toegevoegd, de Swagger documentatie is leidend).

Example Value **Schema**

```
{
  description: Collection of criteria filters
  fieldFilters:
    nullable: true
    List of criteria, note that the weight (cost) of a call increases with each added criteria Filter
    CustomerOrderHeaderCustomerOrderHeaderCollectionInt32OrderFieldsFilter:
      description: Filter criteria for a search (Find) operation
      field:
        Integer($int32)
        example: 7
        Members
        Name Value Description
        CompanyName 101 Type = String
        CustomerNumber 103 Type = String
        Email 104 Type = String
        AddressTypeDescription 105 Type = String
        GenderDescription 106 Type = String
        MobilePhoneNumber 107 Type = String
        LandlinePhoneNumber 108 Type = String
        Website 109 Type = String
        FirstName 110 Type = String
        LastName 111 Type = String
        MiddleName 112 Type = String
        TitleDescription 113 Type = String
        IbanNumber 114 Type = String
        City 115 Type = String
        CountryCode 116 Type = String
        CountryName 117 Type = String
        DefaultAddress 118 Type = Boolean
        HouseNumber 119 Type = String
        HouseNumberSuffix 120 Type = String
        StreetName 121 Type = String
        Zipcode 122 Type = String
        TitleId 123 Type = Int32
        KbcDossierNumber 150 Type = String
        OrderStatusDescription 151 Type = String
        EmployeeName 152 Type = String
        TurnoverEmployeeName 153 Type = String
        OrderNumber 154 Type = String
        EmployeeId 155 Type = Int32
        Barcode 201 Type = String
        FrameNumber 202 Type = String
        ItemNumber 203 Type = String
        ProductDescription 204 Type = String
        ProductNumber 205 Type = String
        ProductSubDescription 206 Type = String
        SerialNumber 207 Type = String
        BrandName 208 Type = String
        ActivationDate 400 Type = DateTime
        DeliveryDate 401 Type = DateTime
        OrderDate 402 Type = DateTime
        OrderStatusDate 403 Type = DateTime
        OrderTypeDescription 404 Type = String
        StockStatus 405 Type = Int32
        OrderStatusEnumValue 407 Type = Int32
        OrderStatusId 408 Type = Int32
        CustomerId 409 Type = Int32
        ContactId 410 Type = Int32
        CountryId 411 Type = Int32
        DeliveryAddressId 412 Type = Int32
        DeliveryContactId 413 Type = Int32
        InvoiceAddressId 414 Type = Int32
        InvoiceContactId 415 Type = Int32
        OrderTypeId 416 Type = Int32
        KbcLeaseInfoId 417 Type = Guid
        TurnoverEmployeeId 418 Type = Int32
        LastModified 524 Type = DateTime
        BarcodeId 560 Type = Int32
        OrderDetailId 705 Type = Int64
        AddressId 1028 Type = Int32
      Enum:
      value: > Array [ 58 ]
      value: > {...}
      nullable: true
      example: SearchText
      value2: > {...}
      nullable: true
      example: 999
      filterComparison:
        Integer($int32)
        example: 11
        Members
        Name Value Description
        Equals 0 =
        NotEquals 1 <>
        GreaterThan 2 >
        LessThan 3 <
        GreaterOrEqual 4 >=
        LessOrEqual 5 <=
        Like 6 LIKE
        IsNull 7 IS NULL
        IsNotNull 8 IS NOT NULL
        StartsWith 9 LIKE '{0}%'
        EndsWith 10 LIKE '%{0}'
        Contains 11 LIKE '%{0}%'
        Between 12 BETWEEN
        NotLike 13 NOT LIKE
        NotStartsWith 14 NOT LIKE '{0}%'
        NotEndsWith 15 NOT LIKE '%{0}'
        NotContains 16 NOT LIKE '%{0}%'
        NotBetween 17 NOT BETWEEN
        Disable 255 Disabled Placeholder (not for use in public API)
      Enum:
      paginationOffset: > Array [ 19 ]
      operator:
        Integer($int32)
        default: 0
        Members
        Name Value Description
        And 0 Intersect all results with each other (only return common results).
        Or 1 Merge all results adding items from each filter.
      Enum:
      > Array [ 2 ]
}
```

Voorbeeld

Als er bijvoorbeeld gezocht moet worden op alle orders die door medewerker “Jan” zijn ingeschoten, dan kan dat op deze manier:

```
{
  "fieldFilters": [
    {
      "field": 152,
      "value": "Jan",
      "filterComparison": 11
    }
  ],
  "paginationOffset": 0,
  "operator": 0
}
```

Field 152 komt overeen met “EmployeeName”

FilterComparison 11 komt overeen met “Contains”

paginationOffset 0 zal de resultaten 1 tot 100 ophalen, om meer op te halen kun je dezelfde call nogmaals aanroepen met paginationOffset 100.

De operator heeft in dit geval geen effect (we hebben slechts 1 criteria) maar je kunt meerdere filters opgeven en dan zullen ze allemaal met And of Or afgehandeld worden. Het is op dit moment nog niet mogelijk om een complexe decision-tree query te maken met gedeeltelijk and en gedeeltelijk or.

Vaak gevraagd

Waarom GetMultiple geen details ophaalt.

De database is het minst schaalbare onderdeel in de keten van services. GetWithDetails haalt een soort boom-structuur op, terwijl Getmultiple zich beperkt tot alleen de stam.

Het uitvoeren van een opdracht in de database kost een bepaalde tijd. Het aantal opdrachten dat een database tegelijkertijd uit kan voeren is beperkt. Grote opdrachten kunnen zorgen voor langere wachttijden voor andere opdrachten.

Door een grote opdracht op te splitsen in veel kleine opdrachten kunnen andere opdrachten van andere processen nog tussentijds verwerkt worden. In het ergste geval kunnen grote requests voor een time-out zorgen, en dat is helemaal zonde want al het werk dat de database tot de time-out heeft gedaan gaat dan ook nog eens verloren.

Het klopt dat er meer overhead mee gepaard gaat, maar deze overhead komt voornamelijk op het schaalbare deel van het systeem (de load-balanced servers die de requests afhandelen) en in veel mindere mate op de database zelf. Deze kan tussen de GetWithDetails transacties ook vele andere belangrijke transacties verwerken (zoals het registreren van nieuwe verkopen, orders etc...).