

Features

OData	2
Pagination	2
Filtering	2
Selecting	3
Sorting	3
Webhooks	4
SubscriptionType	5

OData

The OData functionality is a standardized protocol that you can find more information about on the internet. In this document, we will go through some basics.

Pagination

For every request to an endpoint you make you receive a limited number of objects (50 objects/call). If you have more than 50 objects the pagination function allows you to change the “page” and view the next 50 objects that were not received in the first request, or you can change the “pagesize” so to receive more objects per call.

As an example if you have 70 customers, and you make a GET request for **/customers**, you will receive a meta-object telling you that there are 70 customers, 2 pages and 50 customers on this page, that will mean that the 20 other customers are on the next page.

A solution would be to either, make another request using the endpoint:

```
/customers?$page=2
```

Or

```
/customers?$pagesize=70
```

The “pagesize” limit is 1000, so if you have a several thousand of object you would need to combine these two parameters:

```
/customers?$pagesize=1000&$page=4
```

Filtering

The filtering function allows you limit the number of objects to the ones containing the requirement you define.

If you want to have all vouchers on a specific date:

```
vouchers?$filter=ChangedUtc gt 2016-12-31T23:59:59.00Z
```

For searching properties that contains a specific string:

```
customers?$filter=contains(Name, 'bakery')
```

Selecting

The “Select” function allows you to limit the properties in the response object, as an example:

```
articles?$select=Name, Number, NetPrice, StockBalance
```

Will send you a response of only these properties per every article.

Sorting

Sorting is an “Orderby” function that allows you to sort by the properties of the object:

```
customers?$orderby=Name
```

Will provide you the customers in alphabetical order.

As an example:

```
articles?$select=name&$orderby=name
```

Will limit the properties to only show article names, and then order them alphabetically.

Webhooks

The webhooks function allows you to subscribe to changes made in eAccounting. The changes will be sent to you as an HTTP message to the "Endpoint" you defined. To define where and what changes you to receive you will have to send a POST request to the webhooks endpoint, the following is an example:

POST https://eaccountingapi-sandbox.test.vismaonline.com/v2/webhooks

```
{
  "Endpoint": "<URL to where you want to receive the notifications>",
  "Description": "this is a description",
  "Active": true,
  "Subscriptions": [
    {
      "SubscriptionType": 1
    }
  ]
}
```

GET https://eaccountingapi-sandbox.test.vismaonline.com/v2/webhooks

```
{
  "Meta": {
    "CurrentPage": 1,
    "PageSize": 50,
    "TotalNumberOfPages": 1,
    "TotalNumberOfResults": 1,
    "ServerTimeUtc": "2020-03-24T10:21:37.649442Z"
  },
  "Data": [
    {
      "Id": "4fbecffc-7589-45f8-be70-2bafafeeb333",
      "Endpoint": "https://www.getpostman.com/oauth2/callback",
      "Description": "testing 1",
      "ClientId": "anourtest1",
      "AuthenticationKey": null,
      "AuthenticationValue": null,
      "SigningKey": null,
      "Active": true,
      "Subscriptions": [
        {
          "WebHookId": "4fbecffc-7589-45f8-be70-2bafafeeb333",
          "SubscriptionType": 1
        }
      ]
    }
  ],
}
```

```
"CreatedUtc": "2020-03-24T10:21:27.913Z",  
"ModifiedUtc": "2020-03-24T10:21:27.913Z"  
  }  
 ]  
}
```

DELETE
<https://eaccountingapi-sandbox.test.vismaonline.com/v2/webhooks/4fbecffc-7589-45f8-be70-2bafafeeb333>

<empty response>

SubscriptionType

None = 0, Article = 1, Customer = 2, CustomerInvoice = 3, CustomerInvoiceDraft = 4, Order = 5, Supplier = 6, SupplierInvoice = 7, SupplierInvoiceDraft = 8, Quote = 9, Voucher = 11, WebshopOrder = 12 = ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9', '11', '12']