Visma Business Cloud

ISV edition Happy Hour

November 2021

18th November 2021

Starting 12:30

VISMA

Agenda

- Introduction
- GraphQL API
- Roadmap
- Q&A

Introduction

Alexandra

Organization

R&D

Customer Success (CS)

Business Development (BD)

VBC R&D organization

+40 employees

- Front-end
- Back-end
- UX designers
- Specialists, BAs/QAs and Leads

6 countries and 12 locations

- Norway: Oslo, Fredrikstad, Trondheim
- Sweden: Malmö, Stockholm
- Denmark: Copenhagen
- Finland: Tampere, Turku
- Romania: Timișoara, Sibiu
- Lithuania: Vilnius, Kaunas

Visma Business Cloud - key points

All the benefits of cloud delivery

Same data model
Same business logic

Easy migration

Modern and user-friendly interface

Design and Data Extension capabilities

Flexible and efficient system

Integrated Reporting and Budgeting (OSR) & Payroll

Integration to Connect and all Visma Auto-services

State-of-the art API

Visma
Business Cloud

Last but not least:
VBC will let us reuse the whole value chain of consultants and support out-of the box

VISMA

Functionality and integrations

The following functionality and integrations are in place as of now:

- Designing layouts
- Accounting
- Purchase to Pay (P2P) and Order to Cash (O2C) processes
- Visma Connect: authentication
- Visma.net Admin: administration of users and access
- AutoInvoice: incoming and outgoing invoices, order exchange
- Workflow and Approval: receiving invoices, booking and approval
- AutoPay: incoming and outgoing payments
- AutoCollect: reminders and debt collection
- OneStop Reporting: reporting and budgeting
- GraphQL API

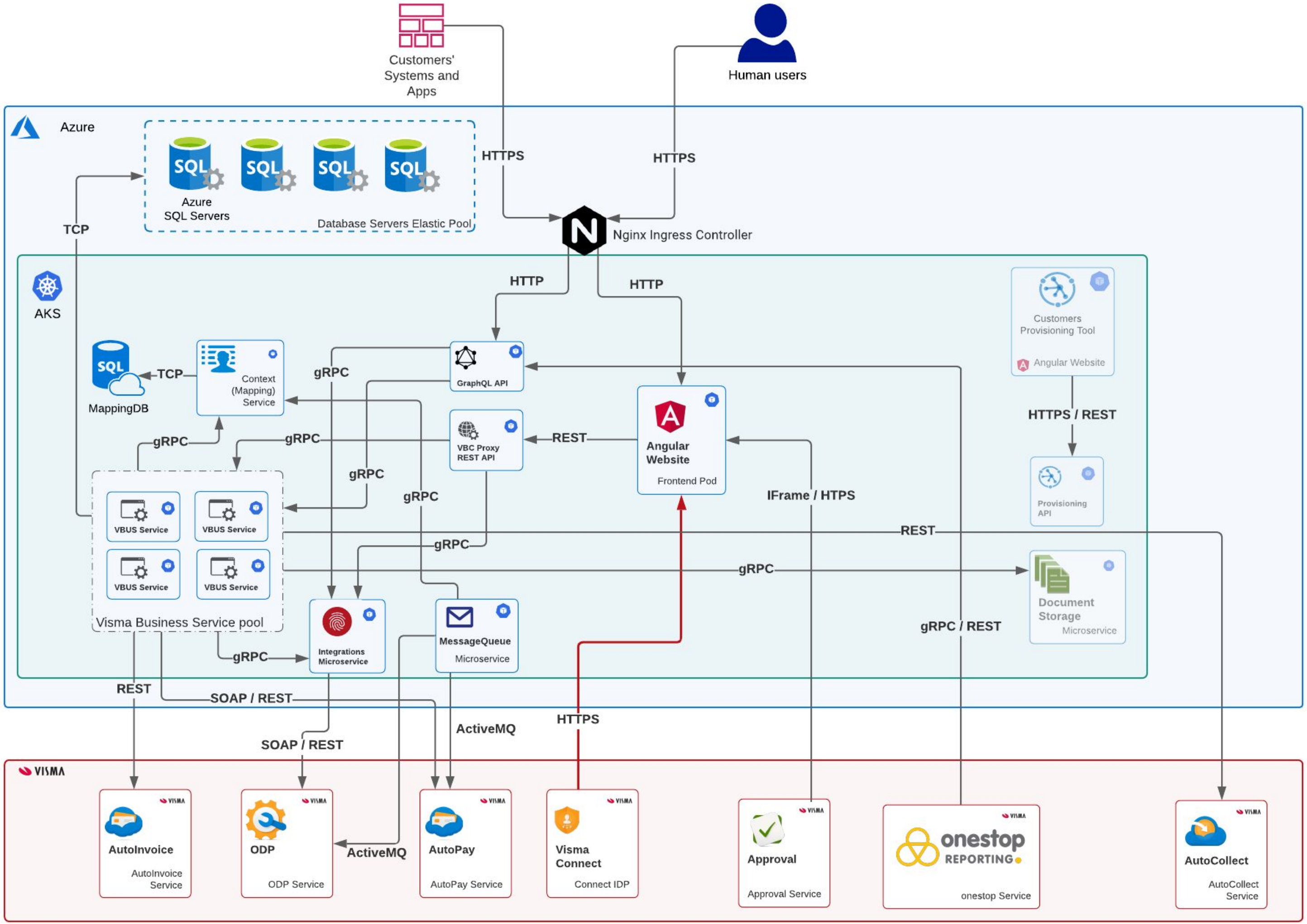
Deployments

Releasing new version to production every week!

Deployments and hotfixes without downtime

Measuring pNPS

Architecture

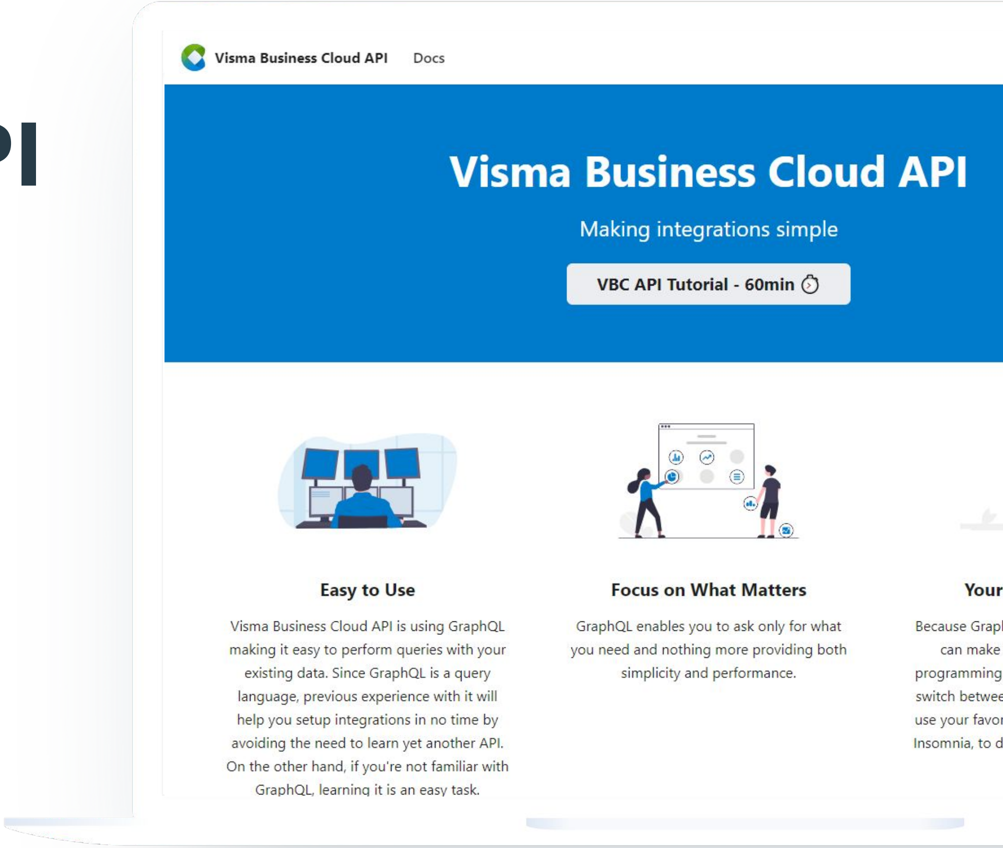


Visma Business Cloud GraphQL API

GraphQL API is our official state-of-the-art API for integrators. It lets API consumers fetch exactly the fields and objects they need in one single request, thus reducing response time.

Functionality that is in place as of now:

- Authentication
- Read
- Mutations (Insert, Update, Delete)
- Pagination, filtering, and sorting
- Aggregates



Visma Business Cloud API Docs

Visma Business Cloud API

Making integrations simple

VBC API Tutorial - 60min

Easy to Use

Visma Business Cloud API is using GraphQL making it easy to perform queries with your existing data. Since GraphQL is a query language, previous experience with it will help you setup integrations in no time by avoiding the need to learn yet another API. On the other hand, if you're not familiar with GraphQL, learning it is an easy task.

Focus on What Matters

GraphQL enables you to ask only for what you need and nothing more providing both simplicity and performance.

Your

Because Grapl
can make
programming
switch betwee
use your favor
Insomnia, to d

VISMA

GraphQL API

Marius

Topics

- What is GraphQL
- What’s in VBC GraphQL
- Live Demos
- Error handling
- Application setup
- Authentication

What is GraphQL

GraphQL

Query manipulation language

A runtime for fulfilling queries

Schema language

- Objects, types (scalar, enumeration, union, input) & interfaces, lists, arguments, etc.

Queries & mutations

- Fields, arguments, variables, fragments, etc.

GraphQL vs REST API

```
GET https://sample.com/person/1
```

```
{
  "firstName": "John",
  "lastName": "Doe",
  "email": "john.doe@gmail.com",
  "address": 42
}
```

```
GET https://sample.com/address/42
```

```
{
  "line1": "Harald Halfdansson, 13",
  "postCode": 1234,
  "city": “Oslo"
}
```

```
POST https://sample.com/graphql
```

```
{
  "query" : "{
    person {
      firstName
      lastName
      address {
        city
      }
    }
  }"
}
```

```
{
  "data" : {
    "person" : {
      "firstName" : "John",
      "lastName" : "Doe",
      "address" : {
        "city" : "Oslo"
      }
    }
  }
}
```

GraphQL Advantages

Declarative approach (focused on data)

Performance (fetch only what you need)

Tooling (schema exploration, live validation and autocomplete, etc.)

Rapid prototyping

GraphQL Disadvantages

Web caching is difficult (does not rely on HTTP caching)

Difficult for smaller applications

Does not support file uploading

GraphQL resources

Introduction to GraphQL
https://graphql.org/learn/

Relay documentation
https://relay.dev/docs/guided-tour/
https://relay.dev/graphql/connections.htm

What’s in VBC GraphQL

VBC GraphQL

Visma Business Cloud API for integrations

Based on the *Relay* specification (not fully compatible with)

Dynamically generated schema from the VBC data model

Ability to query & mutate the entire data (system and company data)

Authentication with Visma Connect

Access rights from Visma Business

Features

- ✓ **Queries**

  Rich support for reading everything you need.

- ✓ **Mutations**

  Inserts, updates, deletes.

- ✓ **Aggregates**

  Backend computation of aggregate functions (sum, average, min, max, count, etc.)

- ✓ **Pagination**

  Both forwards and backwards. Default pages of 1000 records if none is specified.

- ✓ **Filtering and sorting**

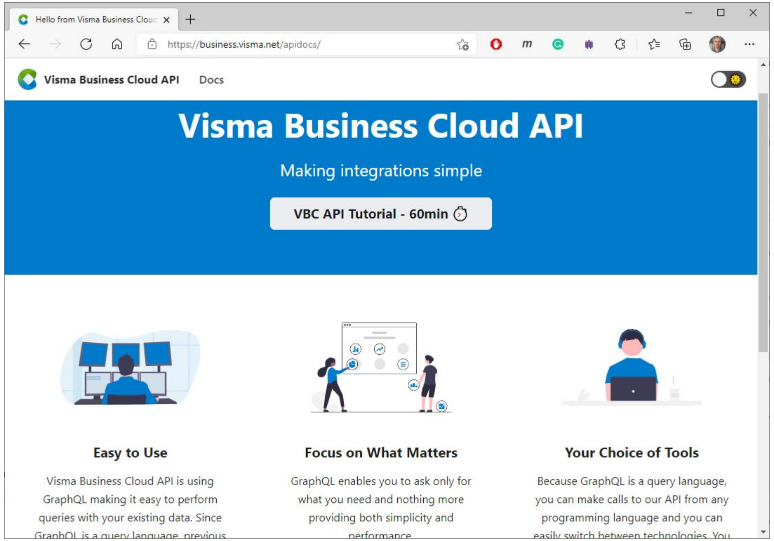
  Define complex filter expressions and specify sorting order.

- ✓ **Fragments, named queries, parameters**

  Various GraphQL features.

Documentation

https://business.visma.net/apidocs



Hello from Visma Business Cloud
https://business.visma.net/apidocs/
Visma Business Cloud API
Docs

Visma Business Cloud API

Making integrations simple

VBC API Tutorial - 60min

Easy to Use

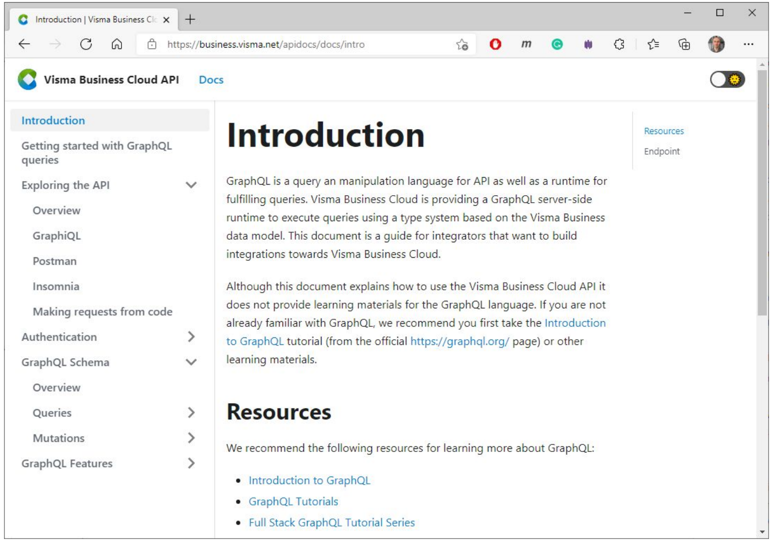
Visma Business Cloud API is using GraphQL making it easy to perform queries with your existing data. Since GraphQL is a query language, previous

Focus on What Matters

GraphQL enables you to ask only for what you need and nothing more providing both simplicity and performance

Your Choice of Tools

Because GraphQL is a query language, you can make calls to our API from any programming language and you can easily switch between technologies. You



Introduction

GraphQL is a query an manipulation language for API as well as a runtime for fulfilling queries. Visma Business Cloud is providing a GraphQL server-side runtime to execute queries using a type system based on the Visma Business data model. This document is a guide for integrators that want to build integrations towards Visma Business Cloud.

Although this document explains how to use the Visma Business Cloud API it does not provide learning materials for the GraphQL language. If you are not already familiar with GraphQL, we recommend you first take the Introduction to GraphQL tutorial (from the official https://graphql.org/ page) or other learning materials.

Resources

We recommend the following resources for learning more about GraphQL:

- Introduction to GraphQL
- GraphQL Tutorials
- Full Stack GraphQL Tutorial Series

Testing the queries

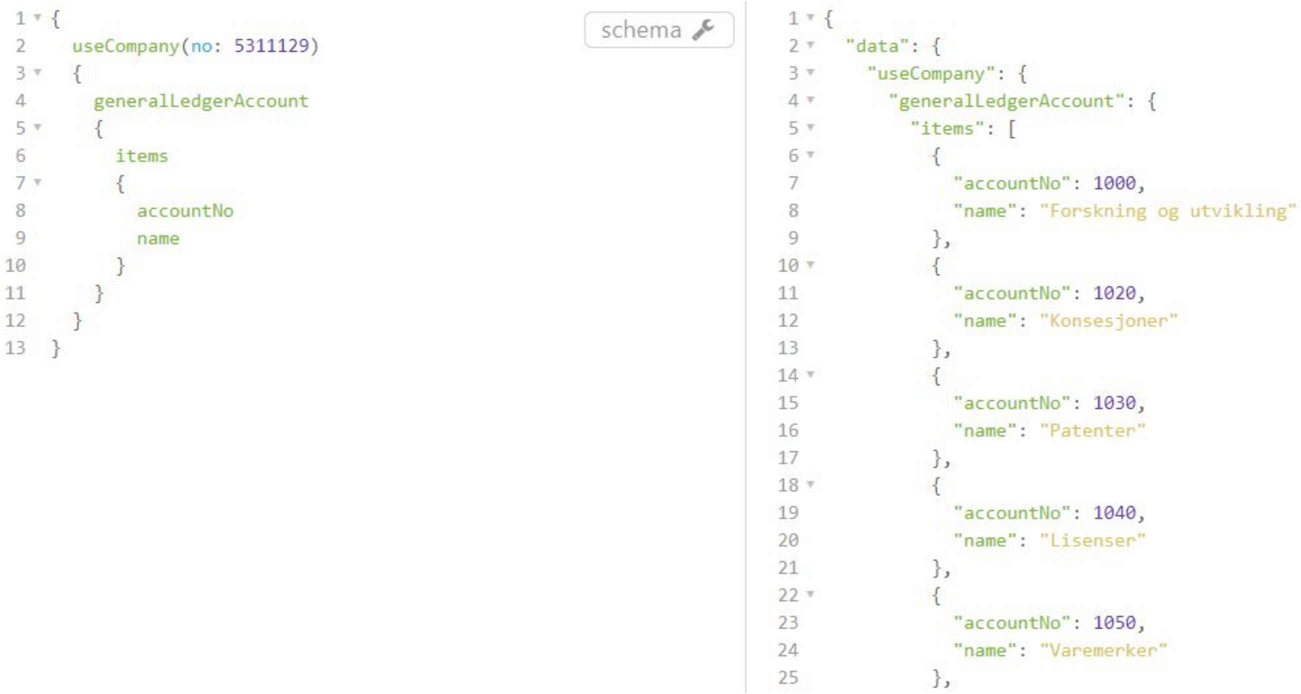
GraphiQL
https://docs.business.visma.net/graphiql/

Postman
https://www.postman.com/

Insomnia
https://insomnia.rest/

Demos

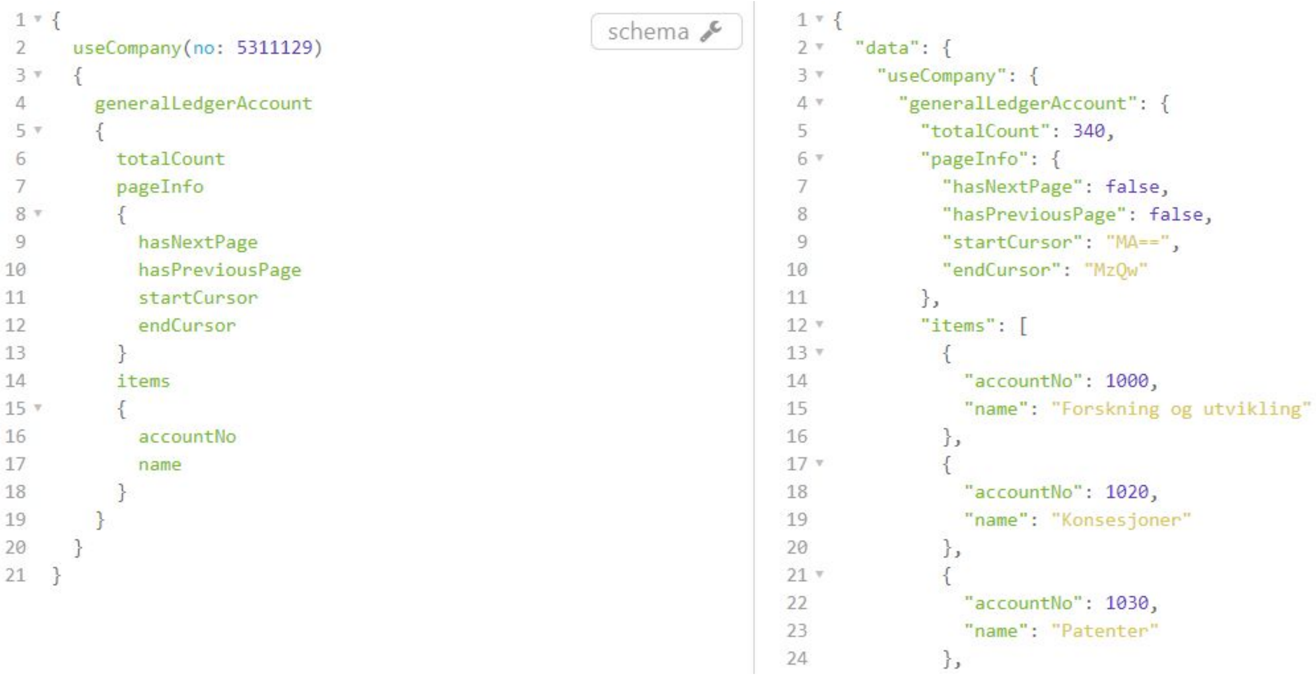
Connections



```
{
  useCompany(no: 5311129)
  {
    generalLedgerAccount
    {
      items
      {
        accountNo
        name
      }
    }
  }
}
```

```
{
  "data": {
    "useCompany": {
      "generalLedgerAccount": {
        "items": [
          {
            "accountNo": 1000,
            "name": "Forskning og utvikling"
          },
          {
            "accountNo": 1020,
            "name": "Konsesjoner"
          },
          {
            "accountNo": 1030,
            "name": "Patenter"
          },
          {
            "accountNo": 1040,
            "name": "Lisenser"
          },
          {
            "accountNo": 1050,
            "name": "Varemerker"
          },
```

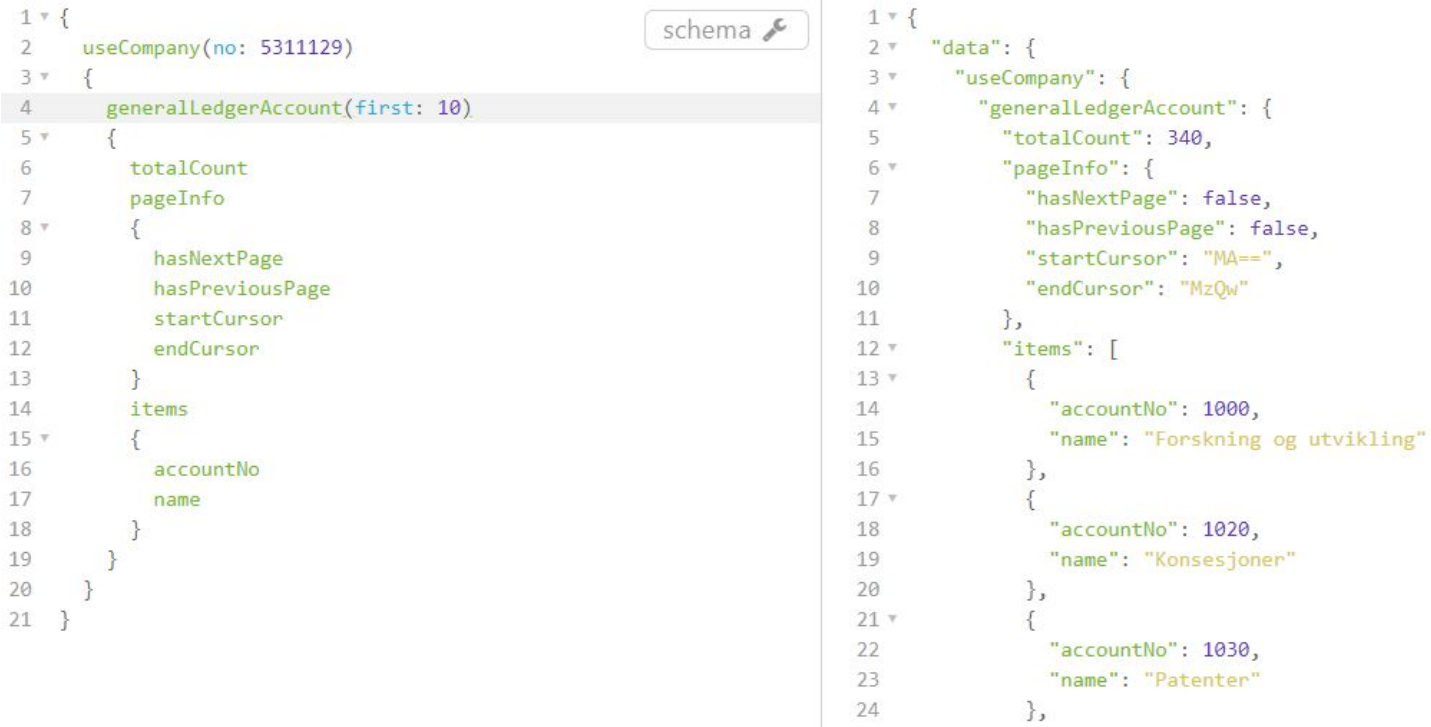
Connections



```
{
  useCompany(no: 5311129)
  {
    generalLedgerAccount
    {
      totalCount
      pageInfo
      {
        hasNextPage
        hasPreviousPage
        startCursor
        endCursor
      }
      items
      {
        accountNo
        name
      }
    }
  }
}
```

```
{
  "data": {
    "useCompany": {
      "generalLedgerAccount": {
        "totalCount": 340,
        "pageInfo": {
          "hasNextPage": false,
          "hasPreviousPage": false,
          "startCursor": "MA==",
          "endCursor": "MzQw"
        },
        "items": [
          {
            "accountNo": 1000,
            "name": "Forskning og utvikling"
          },
          {
            "accountNo": 1020,
            "name": "Konsesjoner"
          },
          {
            "accountNo": 1030,
            "name": "Patenter"
          },
```

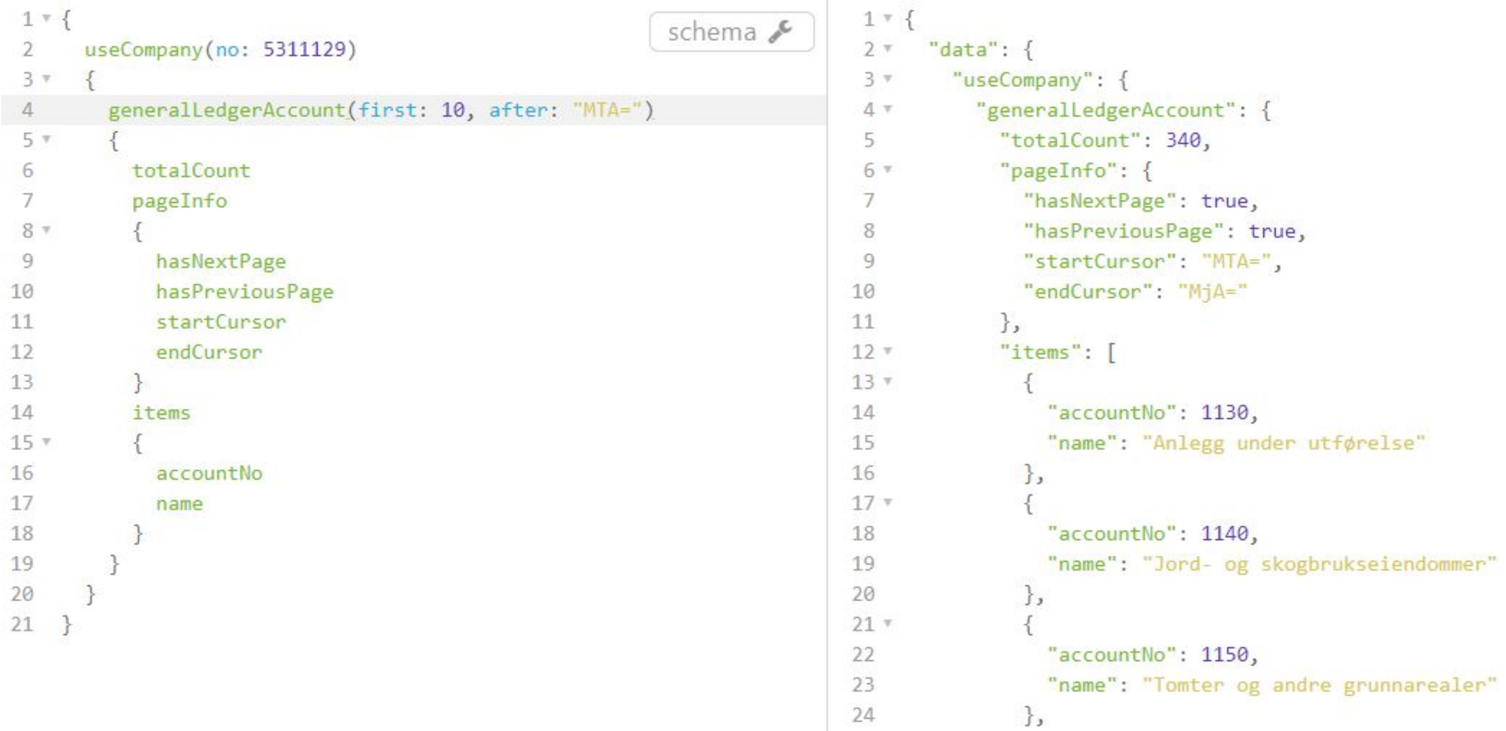
Pagination (first page)



```
{
  useCompany(no: 5311129)
  {
    generalLedgerAccount(first: 10)
    {
      totalCount
      pageInfo
      {
        hasNextPage
        hasPreviousPage
        startCursor
        endCursor
      }
      items
      {
        accountNo
        name
      }
    }
  }
}
```

```
{
  "data": {
    "useCompany": {
      "generalLedgerAccount": {
        "totalCount": 340,
        "pageInfo": {
          "hasNextPage": false,
          "hasPreviousPage": false,
          "startCursor": "MA==",
          "endCursor": "MzQw"
        },
        "items": [
          {
            "accountNo": 1000,
            "name": "Forskning og utvikling"
          },
          {
            "accountNo": 1020,
            "name": "Konsesjoner"
          },
          {
            "accountNo": 1030,
            "name": "Patenter"
          },
```

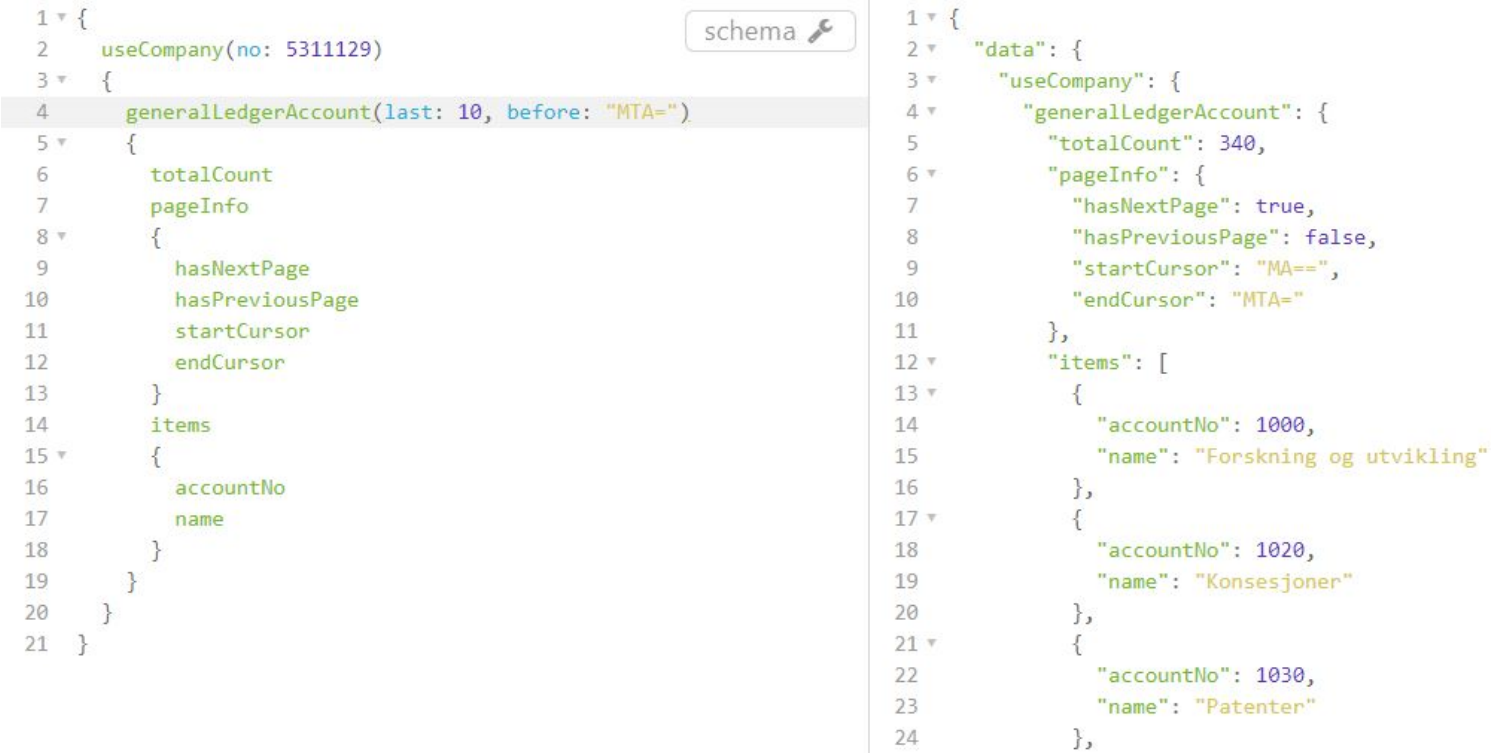
Pagination (next page)



```
{
  useCompany(no: 5311129)
  {
    generalLedgerAccount(first: 10, after: "MTA=")
    {
      totalCount
      pageInfo
      {
        hasNextPage
        hasPreviousPage
        startCursor
        endCursor
      }
      items
      {
        accountNo
        name
      }
    }
  }
}
```

```
{
  "data": {
    "useCompany": {
      "generalLedgerAccount": {
        "totalCount": 340,
        "pageInfo": {
          "hasNextPage": true,
          "hasPreviousPage": true,
          "startCursor": "MTA=",
          "endCursor": "MjA="
        },
        "items": [
          {
            "accountNo": 1130,
            "name": "Anlegg under utførelse"
          },
          {
            "accountNo": 1140,
            "name": "Jord- og skogbrukseiendommer"
          },
          {
            "accountNo": 1150,
            "name": "Tomter og andre grunnarealer"
          },
```

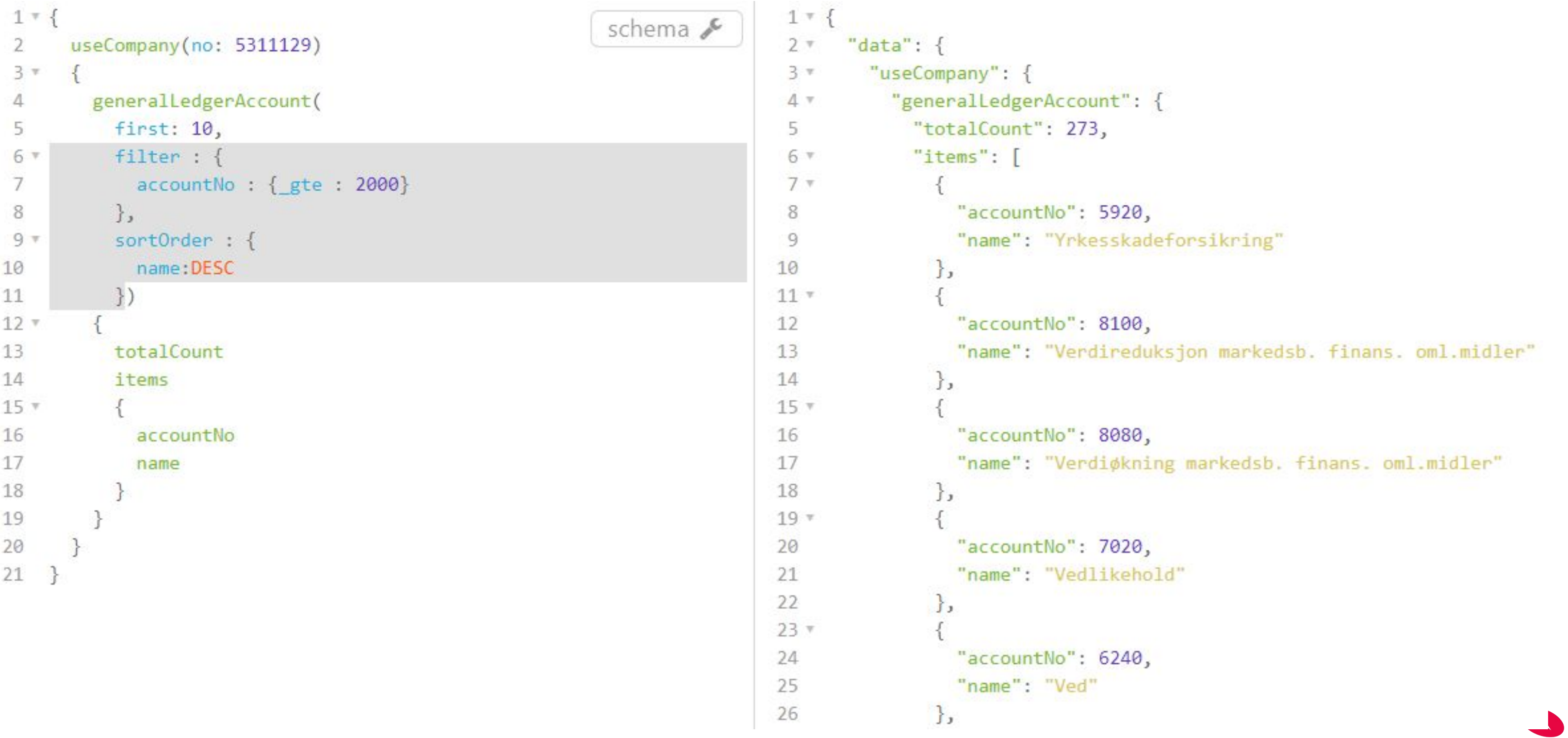
Pagination (previous page)



```
{
  useCompany(no: 5311129)
  {
    generalLedgerAccount(last: 10, before: "MTA=")
    {
      totalCount
      pageInfo
      {
        hasNextPage
        hasPreviousPage
        startCursor
        endCursor
      }
      items
      {
        accountNo
        name
      }
    }
  }
}
```

```
{
  "data": {
    "useCompany": {
      "generalLedgerAccount": {
        "totalCount": 340,
        "pageInfo": {
          "hasNextPage": true,
          "hasPreviousPage": false,
          "startCursor": "MA==",
          "endCursor": "MTA="
        },
        "items": [
          {
            "accountNo": 1000,
            "name": "Forskning og utvikling"
          },
          {
            "accountNo": 1020,
            "name": "Konsesjoner"
          },
          {
            "accountNo": 1030,
            "name": "Patenter"
          },
```

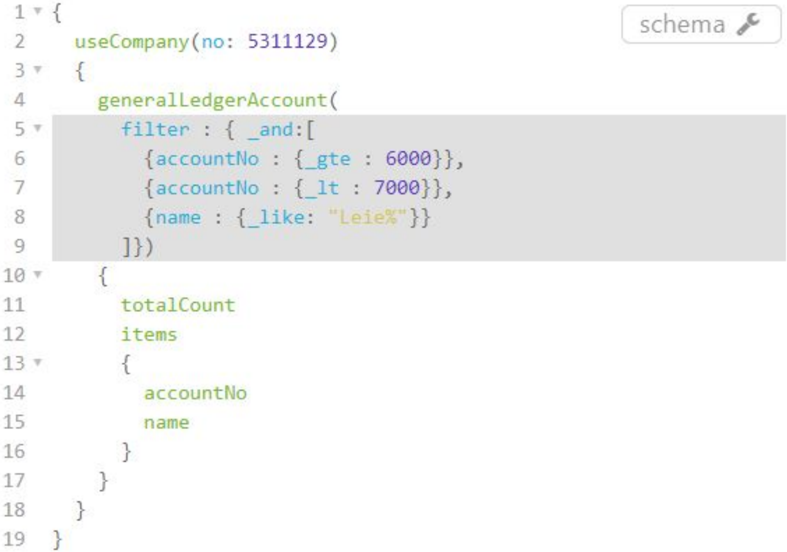
Filtering and sorting



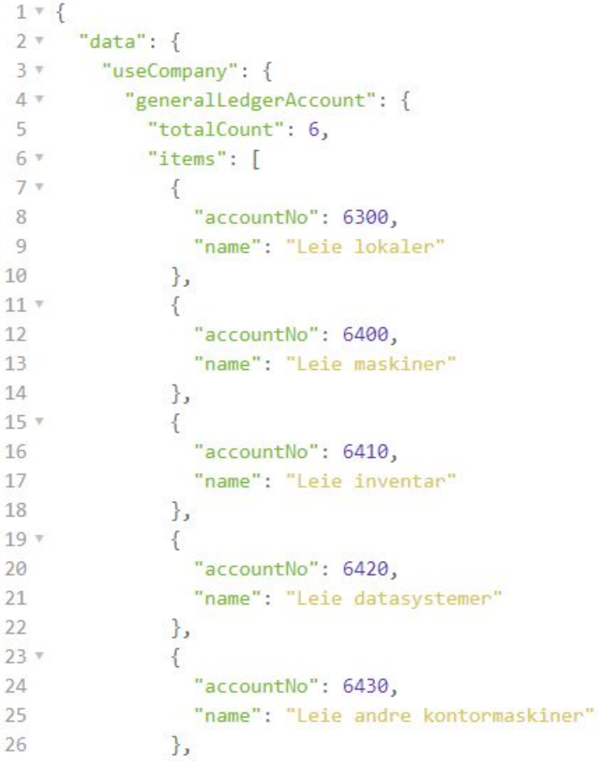
```
{
  useCompany(no: 5311129)
  {
    generalLedgerAccount(
      first: 10,
      filter : {
        accountNo : {_gte : 2000}
      },
      sortOrder : {
        name:DESC
      })
    {
      totalCount
      items
      {
        accountNo
        name
      }
    }
  }
}
```

```
{
  "data": {
    "useCompany": {
      "generalLedgerAccount": {
        "totalCount": 273,
        "items": [
          {
            "accountNo": 5920,
            "name": "Yrkesskadeforsikring"
          },
          {
            "accountNo": 8100,
            "name": "Verdireduksjon markedsb. finans. oml.midler"
          },
          {
            "accountNo": 8080,
            "name": "Verdiøkning markedsb. finans. oml.midler"
          },
          {
            "accountNo": 7020,
            "name": "Vedlikehold"
          },
          {
            "accountNo": 6240,
            "name": "Ved"
          },
```

Filtering and sorting

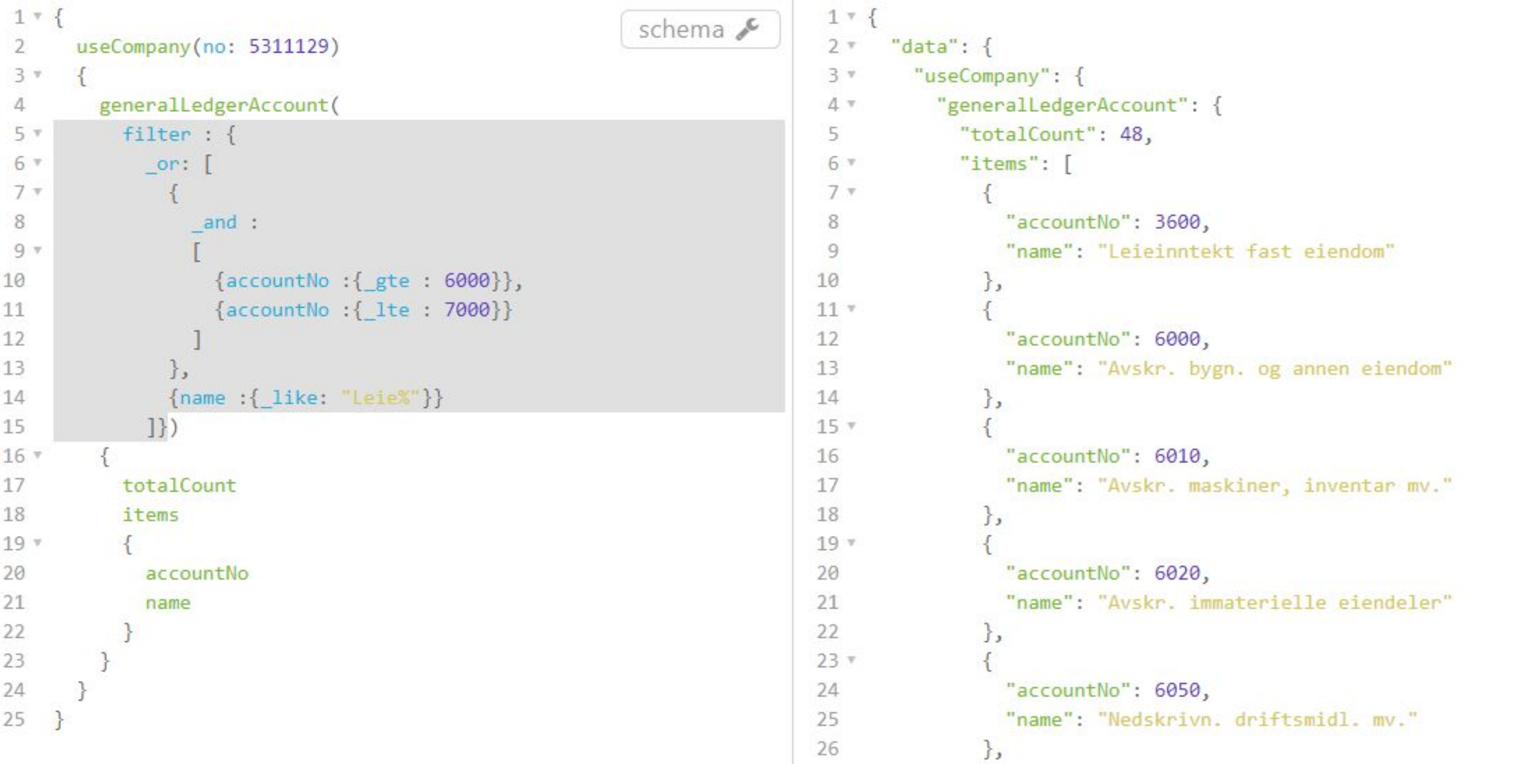


```
{
  useCompany(no: 5311129)
  {
    generalLedgerAccount(
      filter : { _and:[
        {accountNo : {_gte : 6000}},
        {accountNo : {_lt : 7000}},
        {name : {_like: "Leie%"}}
      ]})
    {
      totalCount
      items
      {
        accountNo
        name
      }
    }
  }
}
```



```
{
  "data": {
    "useCompany": {
      "generalLedgerAccount": {
        "totalCount": 6,
        "items": [
          {
            "accountNo": 6300,
            "name": "Leie lokaler"
          },
          {
            "accountNo": 6400,
            "name": "Leie maskiner"
          },
          {
            "accountNo": 6410,
            "name": "Leie inventar"
          },
          {
            "accountNo": 6420,
            "name": "Leie datasystemer"
          },
          {
            "accountNo": 6430,
            "name": "Leie andre kontormaskiner"
          },
```

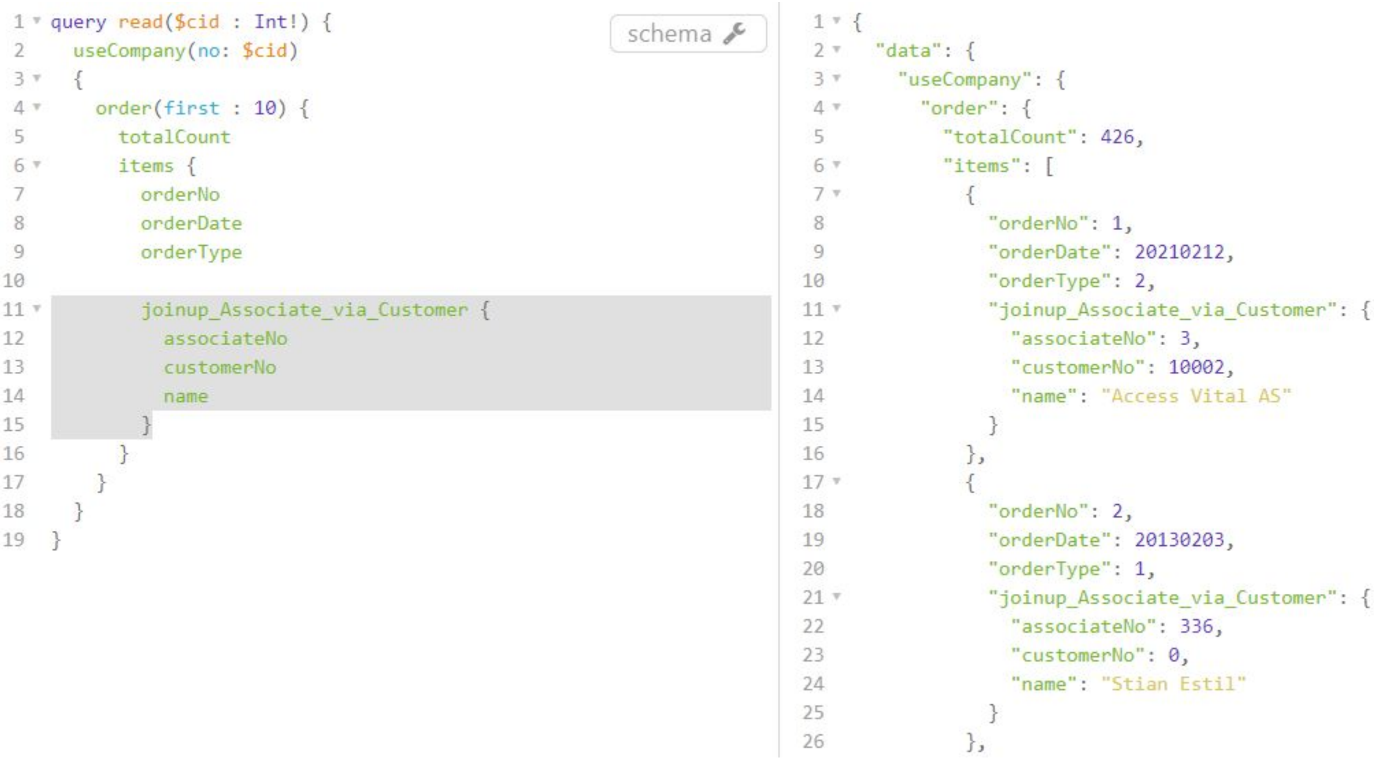
Filtering and sorting



```
{
  useCompany(no: 5311129)
  {
    generalLedgerAccount(
      filter : {
        _or: [
          {
            _and :
            [
              {accountNo :{_gte : 6000}},
              {accountNo :{_lte : 7000}}
            ]
          },
          {name :{_like: "Leie%"}}
        ]})
    {
      totalCount
      items
      {
        accountNo
        name
      }
    }
  }
}
```

```
{
  "data": {
    "useCompany": {
      "generalLedgerAccount": {
        "totalCount": 48,
        "items": [
          {
            "accountNo": 3600,
            "name": "Leieinntekt fast eiendom"
          },
          {
            "accountNo": 6000,
            "name": "Avskr. bygn. og annen eiendom"
          },
          {
            "accountNo": 6010,
            "name": "Avskr. maskiner, inventar mv."
          },
          {
            "accountNo": 6020,
            "name": "Avskr. immaterielle eiendeler"
          },
          {
            "accountNo": 6050,
            "name": "Nedskrivn. driftsmidl. mv."
          },
```

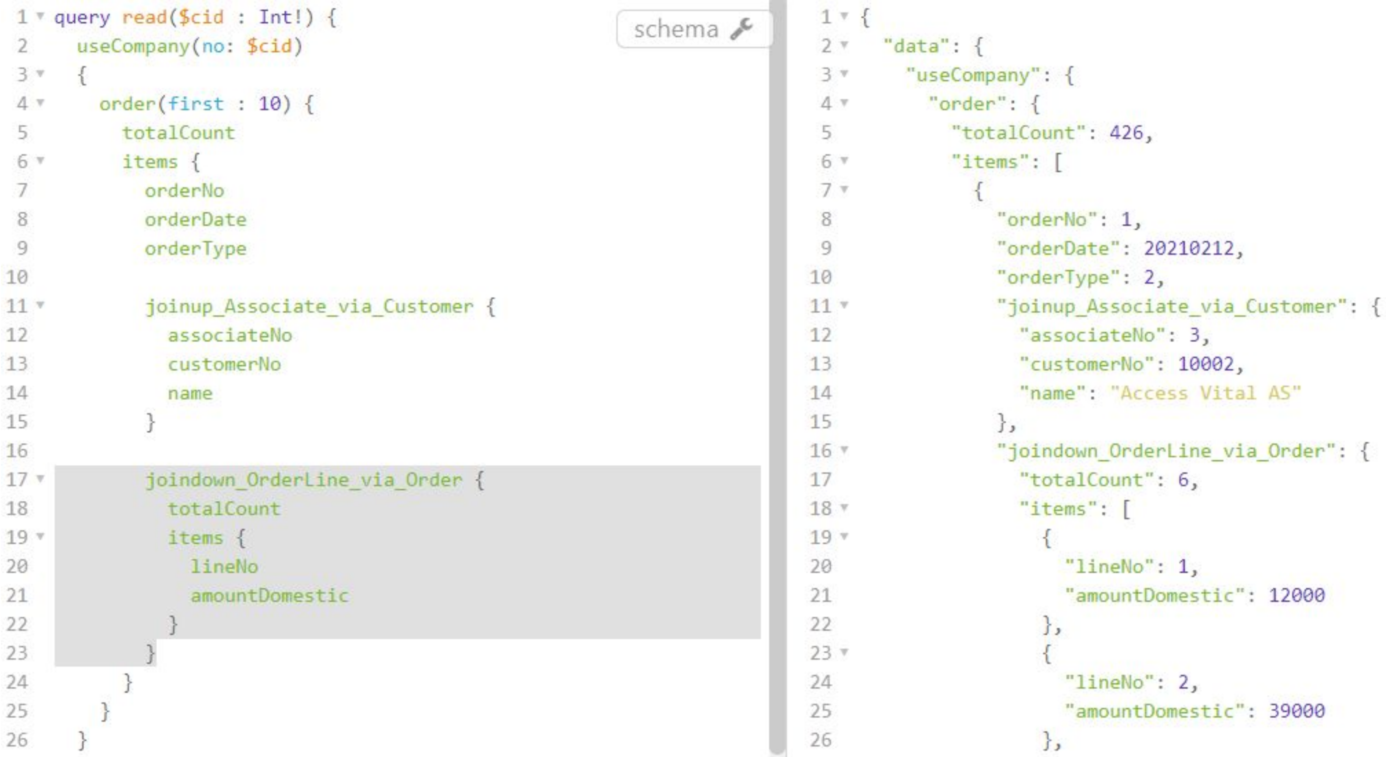
Joins



```
query read($cid : Int!) {
  useCompany(no: $cid)
  {
    order(first : 10) {
      totalCount
      items {
        orderNo
        orderDate
        orderType

        joinup_Associate_via_Customer {
          associateNo
          customerNo
          name
        }
      }
    }
  }
}
```

```
{
  "data": {
    "useCompany": {
      "order": {
        "totalCount": 426,
        "items": [
          {
            "orderNo": 1,
            "orderDate": 20210212,
            "orderType": 2,
            "joinup_Associate_via_Customer": {
              "associateNo": 3,
              "customerNo": 10002,
              "name": "Access Vital AS"
            }
          },
          {
            "orderNo": 2,
            "orderDate": 20130203,
            "orderType": 1,
            "joinup_Associate_via_Customer": {
              "associateNo": 336,
              "customerNo": 0,
              "name": "Stian Estil"
            }
          },
```

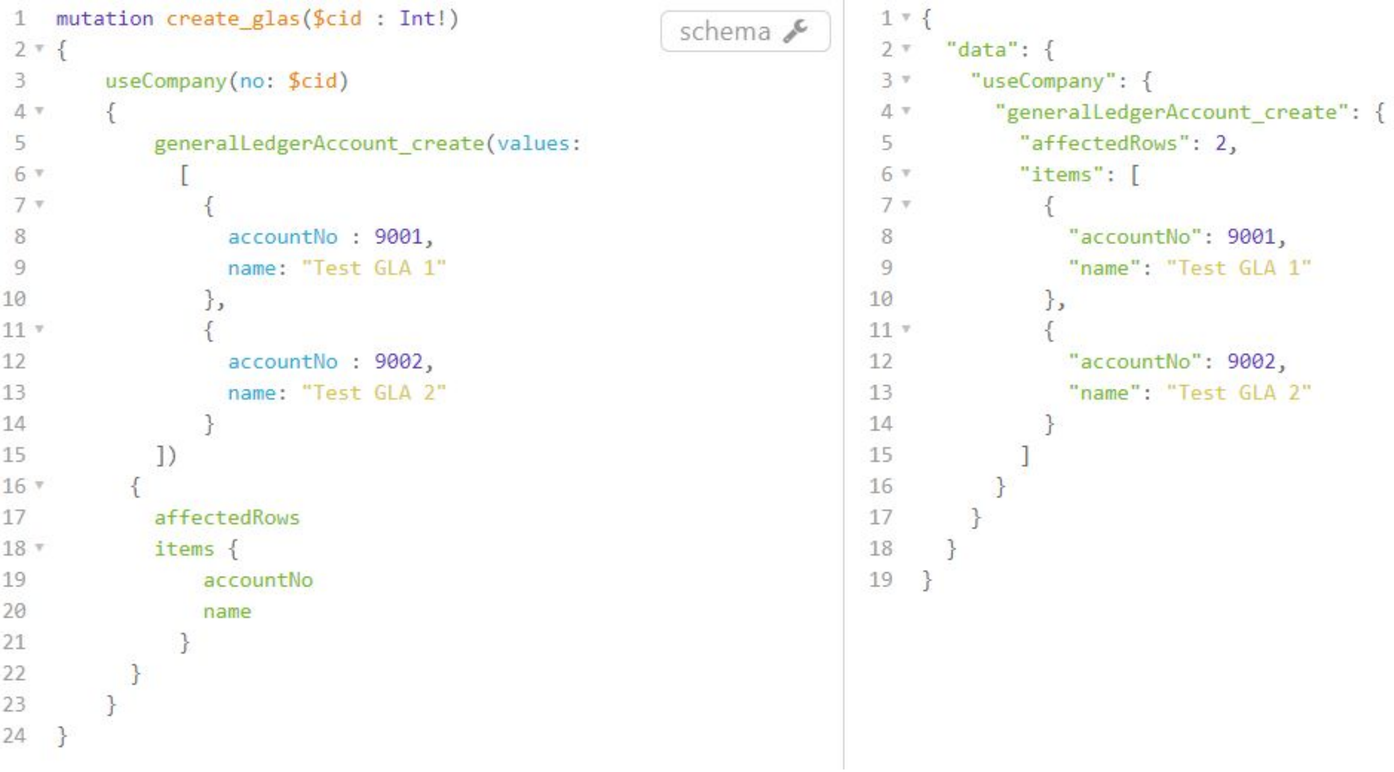
Joins



```
query read($cid : Int!) {
  useCompany(no: $cid)
  {
    order(first : 10) {
      totalCount
      items {
        orderNo
        orderDate
        orderType

        joinup_Associate_via_Customer {
          associateNo
          customerNo
          name
        }

        joindown_OrderLine_via_Order {
          totalCount
          items {
            lineNo
            amountDomestic
          }
        }
      }
    }
  }
```

```
{
  "data": {
    "useCompany": {
      "order": {
        "totalCount": 426,
        "items": [
          {
            "orderNo": 1,
            "orderDate": 20210212,
            "orderType": 2,
            "joinup_Associate_via_Customer": {
              "associateNo": 3,
              "customerNo": 10002,
              "name": "Access Vital AS"
            },
            "joindown_OrderLine_via_Order": {
              "totalCount": 6,
              "items": [
                {
                  "lineNo": 1,
                  "amountDomestic": 12000
                },
                {
                  "lineNo": 2,
                  "amountDomestic": 39000
                },
```

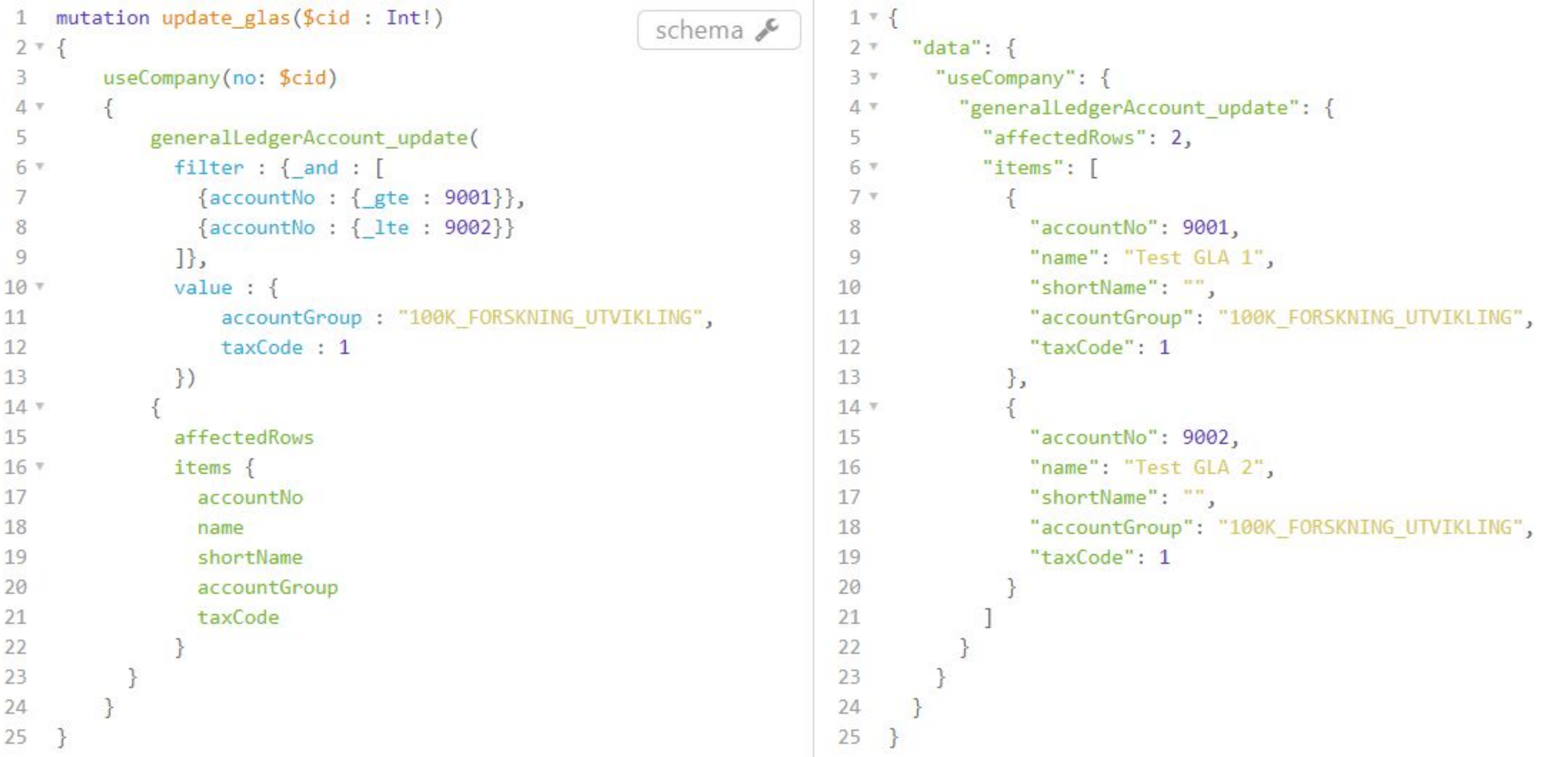
Inserts



```
mutation create_glas($cid : Int!)
{
  useCompany(no: $cid)
  {
    generalLedgerAccount_create(values:
     [
      {
       accountNo : 9001,
       name: "Test GLA 1"
      },
      {
       accountNo : 9002,
       name: "Test GLA 2"
      }
    ])
   {
    affectedRows
    items {
      accountNo
      name
     }
   }
  }
}
```

```
{
 "data": {
  "useCompany": {
   "generalLedgerAccount_create": {
    "affectedRows": 2,
    "items": [
     {
      "accountNo": 9001,
      "name": "Test GLA 1"
     },
     {
      "accountNo": 9002,
      "name": "Test GLA 2"
     }
    ]
   }
  }
 }
}
```

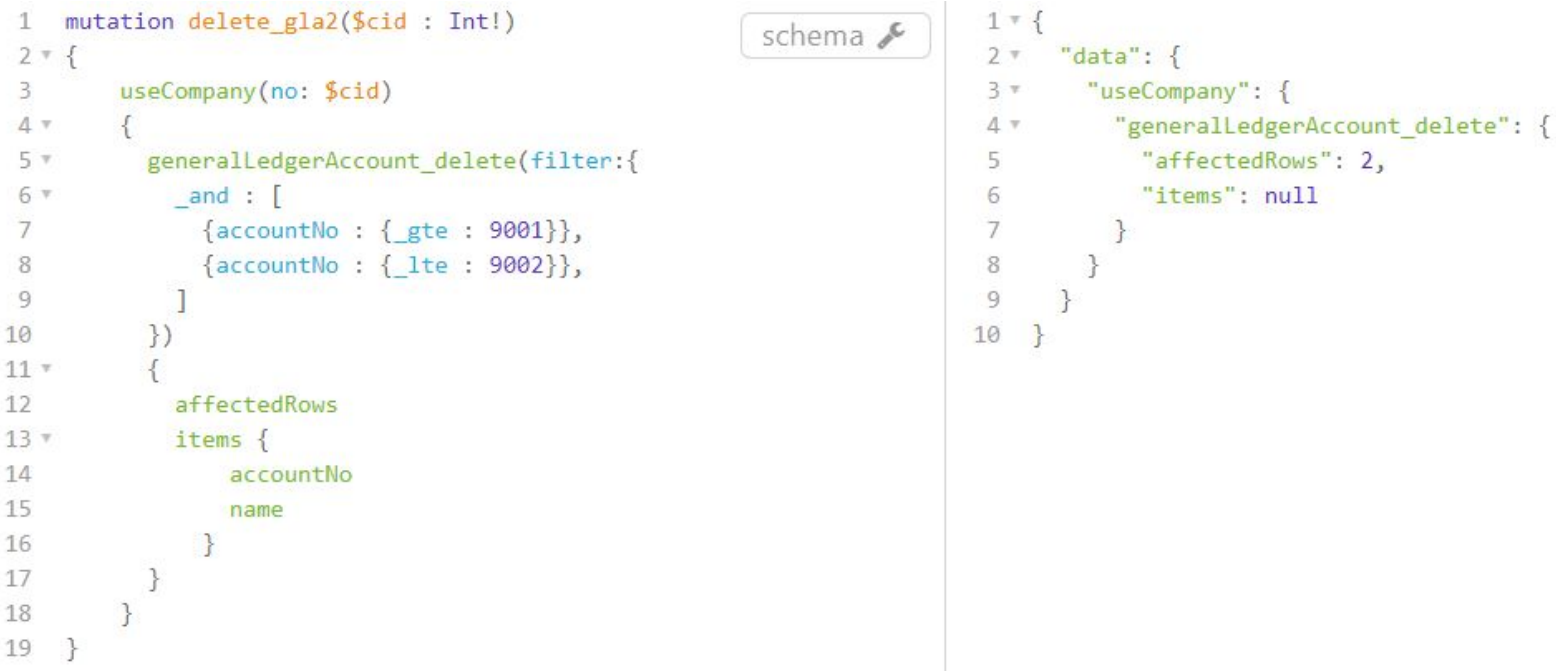
Updates



```
mutation update_glas($cid : Int!)
{
    useCompany(no: $cid)
    {
        generalLedgerAccount_update(
          filter : {_and : [
            {accountNo : {_gte : 9001}},
            {accountNo : {_lte : 9002}}
          ]},
          value : {
              accountGroup : "100K_FORSKNING_UTVIKLING",
              taxCode : 1
          })
        {
          affectedRows
          items {
            accountNo
            name
            shortName
            accountGroup
            taxCode
          }
      }
    }
}
```

```
{
  "data": {
    "useCompany": {
      "generalLedgerAccount_update": {
        "affectedRows": 2,
        "items": [
          {
            "accountNo": 9001,
            "name": "Test GLA 1",
            "shortName": "",
            "accountGroup": "100K_FORSKNING_UTVIKLING",
            "taxCode": 1
          },
          {
            "accountNo": 9002,
            "name": "Test GLA 2",
            "shortName": "",
            "accountGroup": "100K_FORSKNING_UTVIKLING",
            "taxCode": 1
          }
        ]
      }
    }
  }
}
```

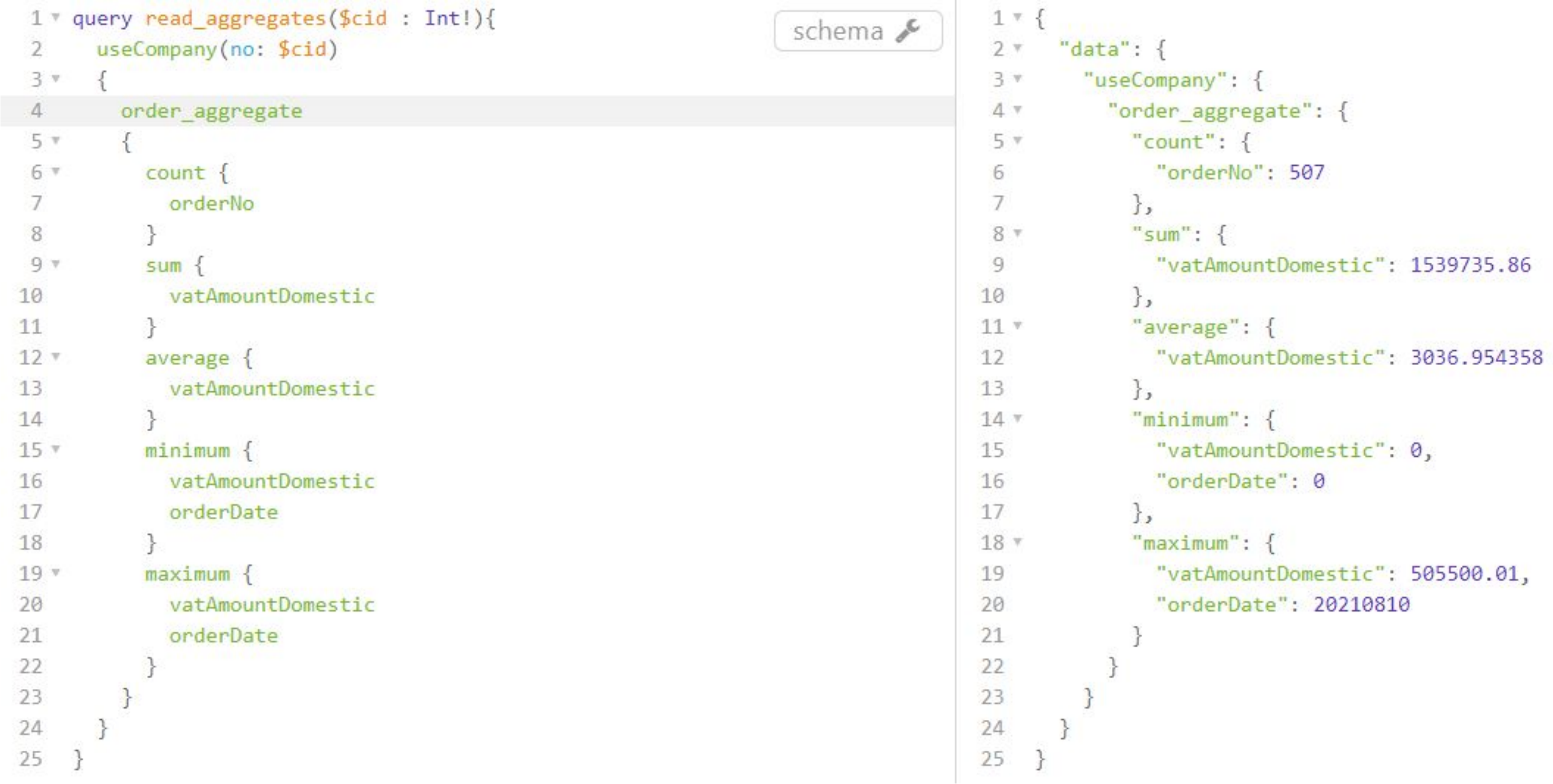
Deletes



```
mutation delete_gla2($cid : Int!)
{
    useCompany(no: $cid)
    {
      generalLedgerAccount_delete(filter:{
        _and : [
          {accountNo : {_gte : 9001}},
          {accountNo : {_lte : 9002}},
        ]
      })
      {
        affectedRows
        items {
            accountNo
            name
          }
      }
    }
}
```

```
{
  "data": {
    "useCompany": {
      "generalLedgerAccount_delete": {
        "affectedRows": 2,
        "items": null
      }
    }
  }
}
```

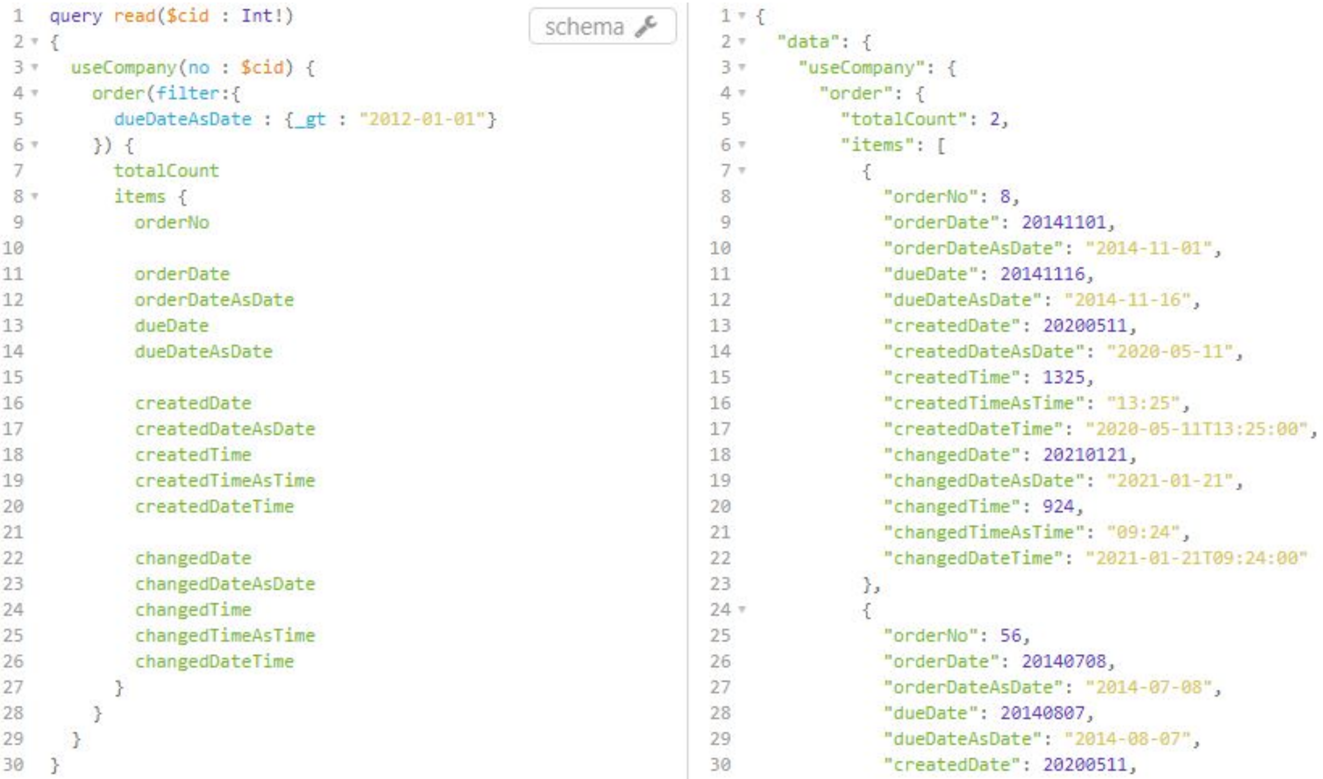
Aggregates



```
query read_aggregates($cid : Int!){
  useCompany(no: $cid)
  {
    order_aggregate
    {
      count {
        orderNo
      }
      sum {
        vatAmountDomestic
      }
      average {
        vatAmountDomestic
      }
      minimum {
        vatAmountDomestic
        orderDate
      }
      maximum {
        vatAmountDomestic
        orderDate
      }
    }
  }
}
```

```
{
  "data": {
    "useCompany": {
      "order_aggregate": {
        "count": {
          "orderNo": 507
        },
        "sum": {
          "vatAmountDomestic": 1539735.86
        },
        "average": {
          "vatAmountDomestic": 3036.954358
        },
        "minimum": {
          "vatAmountDomestic": 0,
          "orderDate": 0
        },
        "maximum": {
          "vatAmountDomestic": 505500.01,
          "orderDate": 20210810
        }
      }
    }
  }
}
```

Date and time fields



```
query read($cid : Int!)
{
  useCompany(no : $cid) {
    order(filter:{
      dueDateAsDate : {_gt : "2012-01-01"}
    }) {
      totalCount
      items {
        orderNo

        orderDate
        orderDateAsDate
        dueDate
        dueDateAsDate

        createdDate
        createdDateAsDate
        createdTime
        createdTimeAsTime
        createdDateTime

        changedDate
        changedDateAsDate
        changedTime
        changedTimeAsTime
        changedDateTime
      }
    }
  }
}
```

```
{
  "data": {
    "useCompany": {
      "order": {
        "totalCount": 2,
        "items": [
          {
            "orderNo": 8,
            "orderDate": 20141101,
            "orderDateAsDate": "2014-11-01",
            "dueDate": 20141116,
            "dueDateAsDate": "2014-11-16",
            "createdDate": 20200511,
            "createdDateAsDate": "2020-05-11",
            "createdTime": 1325,
            "createdTimeAsTime": "13:25",
            "createdDateTime": "2020-05-11T13:25:00",
            "changedDate": 20210121,
            "changedDateAsDate": "2021-01-21",
            "changedTime": 924,
            "changedTimeAsTime": "09:24",
            "changedDateTime": "2021-01-21T09:24:00"
          },
          {
            "orderNo": 56,
            "orderDate": 20140708,
            "orderDateAsDate": "2014-07-08",
            "dueDate": 20140807,
            "dueDateAsDate": "2014-08-07",
            "createdDate": 20200511,
```

GraphQL vs VBS

Comparison to VBS

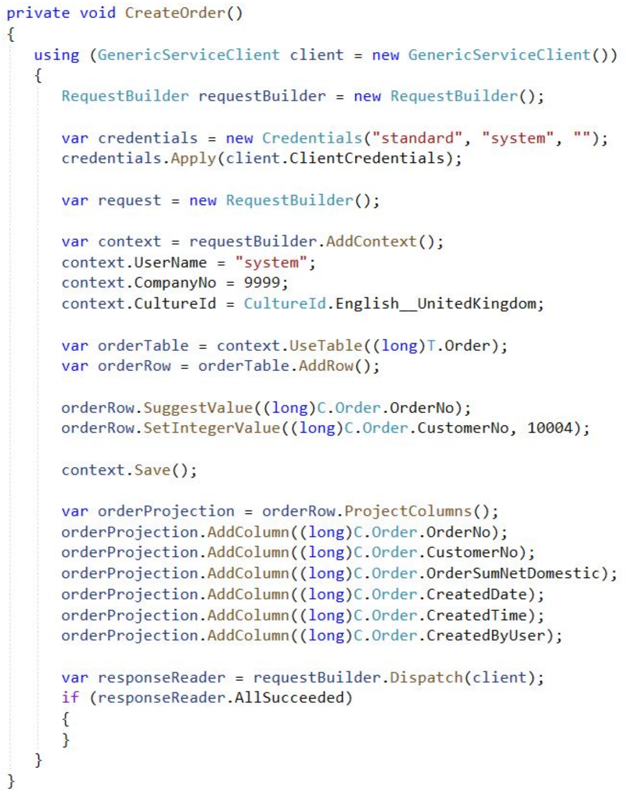
VBS

- Procedural
- Focused on how to do things
- Steep API learning curve

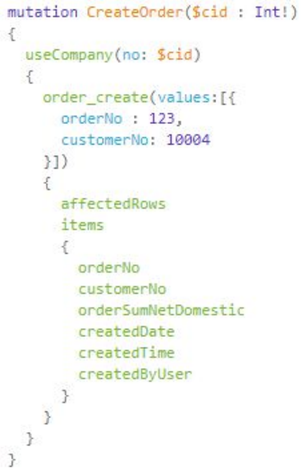
GraphQL

- Declarative
- Data-centric
- Focused on what to do
- IDEs with syntax autocomplete and schema exploration
  - Insomnia
  - GraphiQL

GraphQL vs VBS



```
private void CreateOrder()
{
    using (GenericServiceClient client = new GenericServiceClient())
    {
        RequestBuilder requestBuilder = new RequestBuilder();

        var credentials = new Credentials("standard", "system", "");
        credentials.Apply(client.ClientCredentials);

        var request = new RequestBuilder();

        var context = requestBuilder.AddContext();
        context.UserName = "system";
        context.CompanyNo = 9999;
        context.CultureId = CultureId.English__UnitedKingdom;

        var orderTable = context.UseTable((long)T.Order);
        var orderRow = orderTable.AddRow();

        orderRow.SuggestValue((long)C.Order.OrderNo);
        orderRow.SetIntegerValue((long)C.Order.CustomerNo, 10004);

        context.Save();

        var orderProjection = orderRow.ProjectColumns();
        orderProjection.AddColumn((long)C.Order.OrderNo);
        orderProjection.AddColumn((long)C.Order.CustomerNo);
        orderProjection.AddColumn((long)C.Order.OrderSumNetDomestic);
        orderProjection.AddColumn((long)C.Order.CreatedDate);
        orderProjection.AddColumn((long)C.Order.CreatedTime);
        orderProjection.AddColumn((long)C.Order.CreatedByUser);

        var responseReader = requestBuilder.Dispatch(client);
        if (responseReader.AllSucceeded)
        {
        }
    }
}
```



```
mutation CreateOrder($cid : Int!)
{
  useCompany(no: $cid)
  {
    order_create(values:[{
      orderNo : 123,
      customerNo: 10004
    }])
    {
      affectedRows
      items
      {
        orderNo
        customerNo
        orderSumNetDomestic
        createdDate
        createdTime
        createdByUser
      }
    }
  }
}
```

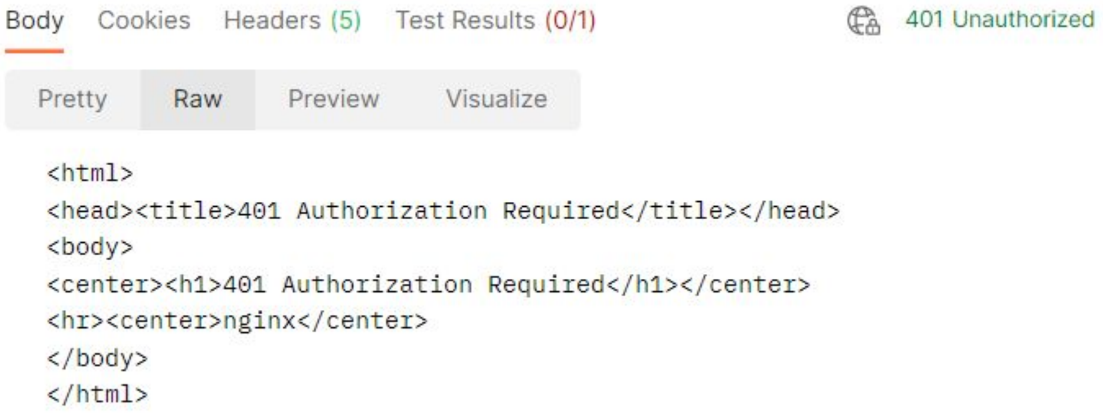
Error handling

Error handling

HTTP status codes

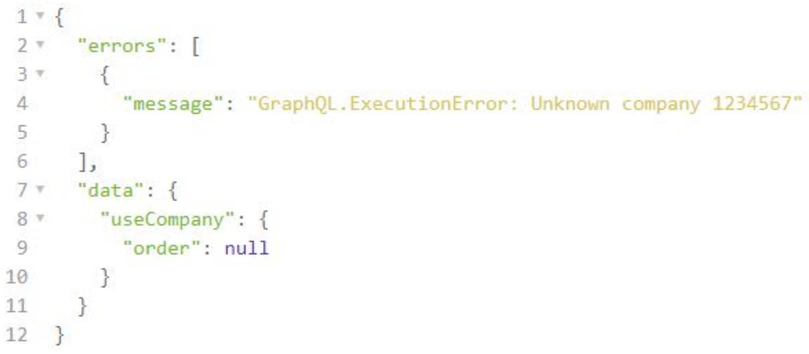
- 401 (Unauthorized)
  - No token or expired token
- 200 (OK)
  - All other requests
  - The `errors` object in the reponsone contains information about the errors

Error examples (401)

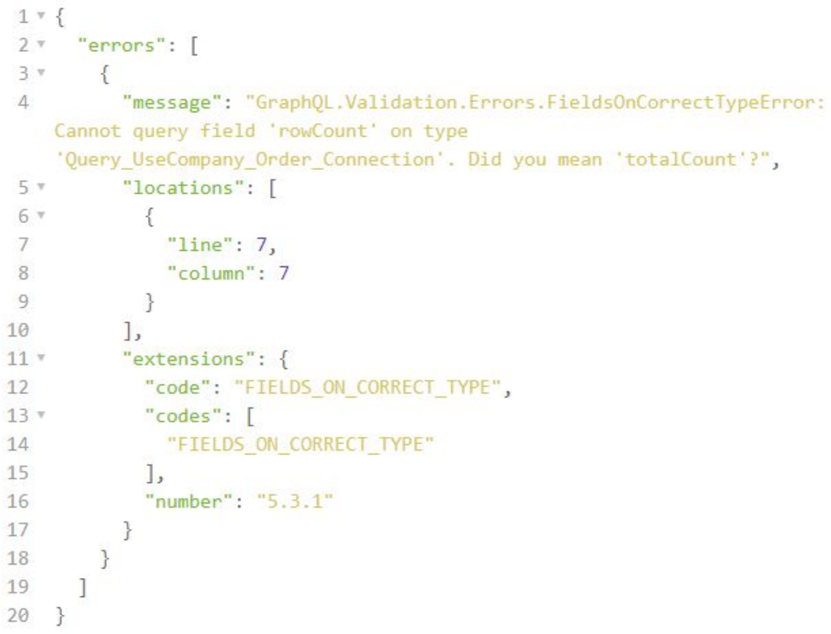


```
<html>
<head><title>401 Authorization Required</title></head>
<body>
<center><h1>401 Authorization Required</h1></center>
<hr><center>nginx</center>
</body>
</html>
```

Error examples (200)



```
{
  "errors": [
    {
      "message": "GraphQL.ExecutionError: Unknown company 1234567"
    }
  ],
  "data": {
    "useCompany": {
      "order": null
    }
  }
}
```



```
{
  "errors": [
    {
      "message": "GraphQL.Validation.Errors.FieldsOnCorrectTypeError: Cannot query field 'rowCount' on type 'Query_UseCompany_Order_Connection'. Did you mean 'totalCount'?",
      "locations": [
        {
          "line": 7,
          "column": 7
        }
      ],
      "extensions": {
        "code": "FIELDS_ON_CORRECT_TYPE",
        "codes": [
          "FIELDS_ON_CORRECT_TYPE"
        ],
        "number": "5.3.1"
      }
    }
  ]
}
```

Application setup

Setup flow

1. Logon to Visma Developer Portal
   https://developer.visma.com/

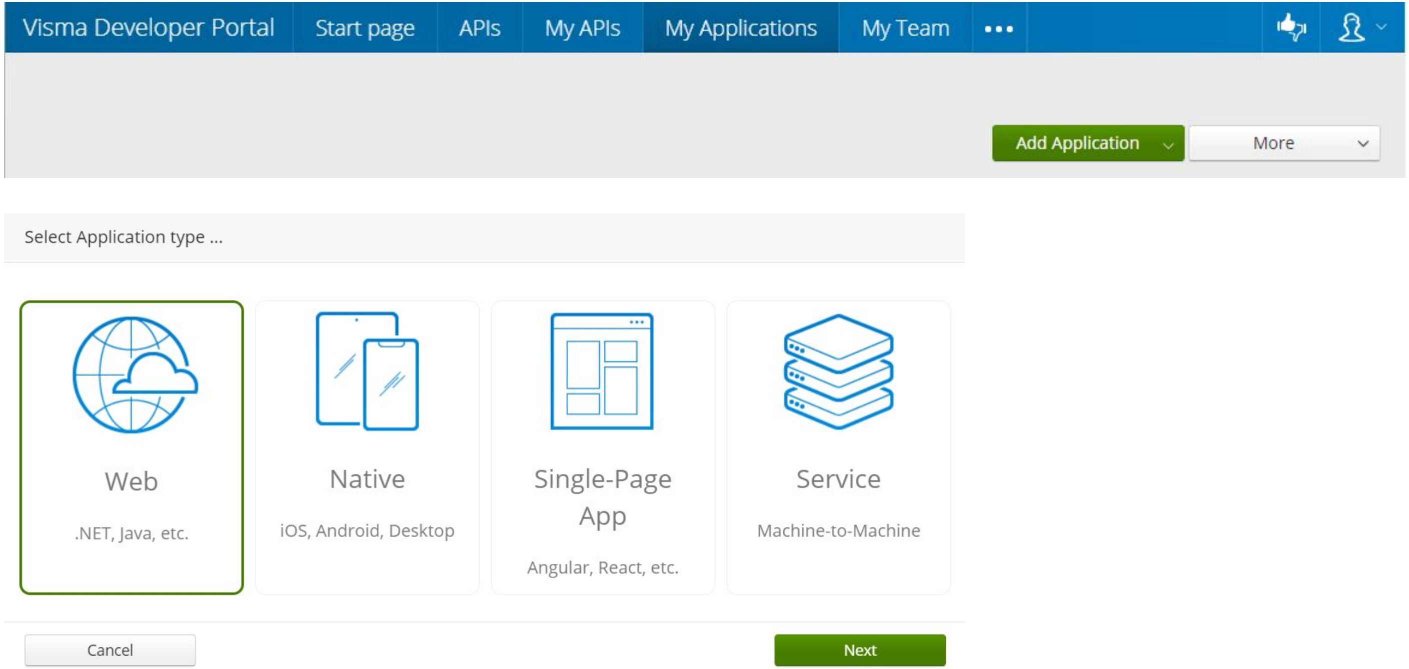
2. Create a new application

3. Wait to be approved

4. Add an integration with Visma Business Cloud Graphql API

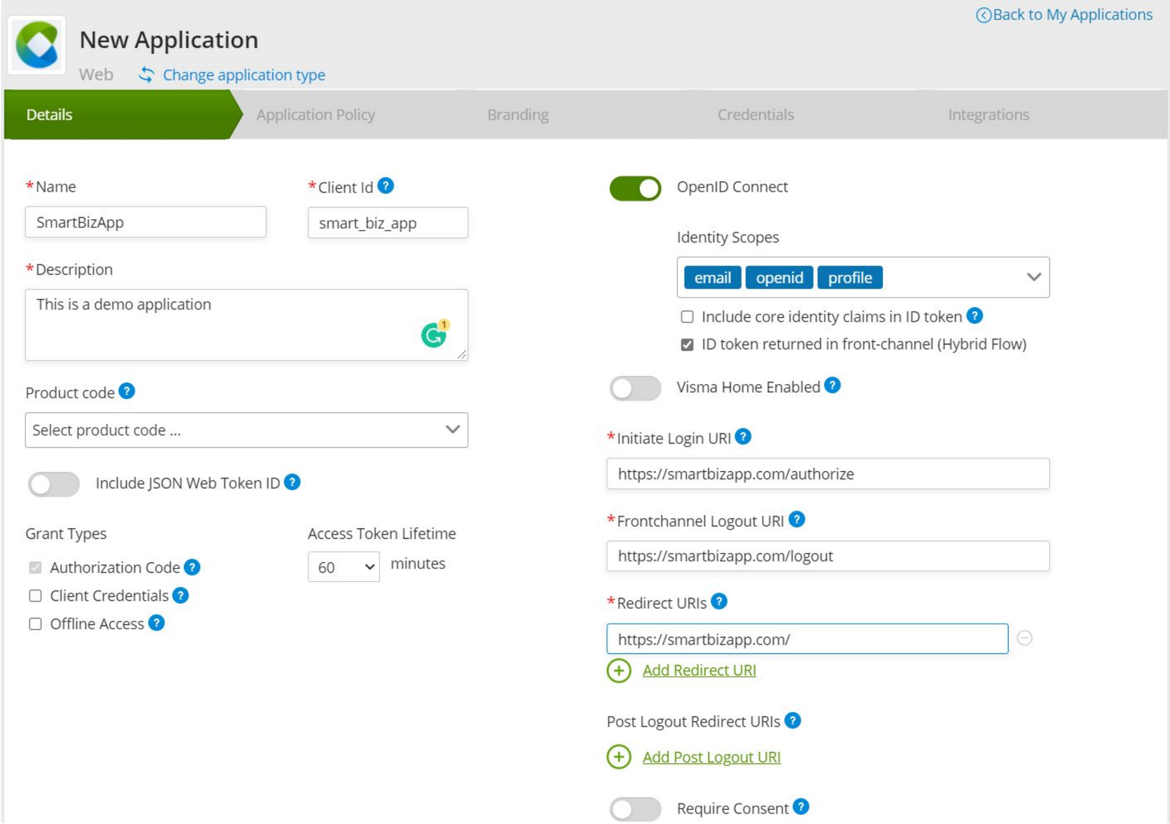
5. Wait to be approved

Setup flow - select application type



Select Application type ...

Setup flow - fill in application registration form



New Application

Web — Change application type

*Name: SmartBizApp

*Client Id: smart_biz_app

*Description: This is a demo application

Product code: Select product code ...

○ Include JSON Web Token ID

Grant Types

☑ Authorization Code

□ Client Credentials

□ Offline Access

Access Token Lifetime: 60 minutes

● OpenID Connect

Identity Scopes: email, openid, profile

□ Include core identity claims in ID token

☑ ID token returned in front-channel (Hybrid Flow)

○ Visma Home Enabled

*Initiate Login URI: https://smartbizapp.com/authorize

*Frontchannel Logout URI: https://smartbizapp.com/logout

*Redirect URIs: https://smartbizapp.com/

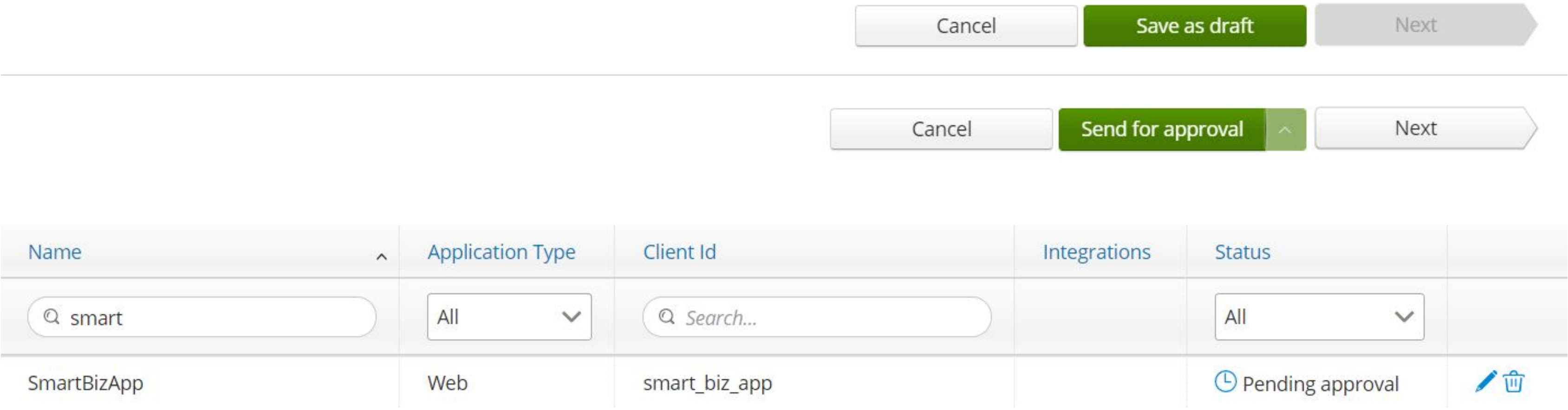
Add Redirect URI

Post Logout Redirect URIs

Add Post Logout URI

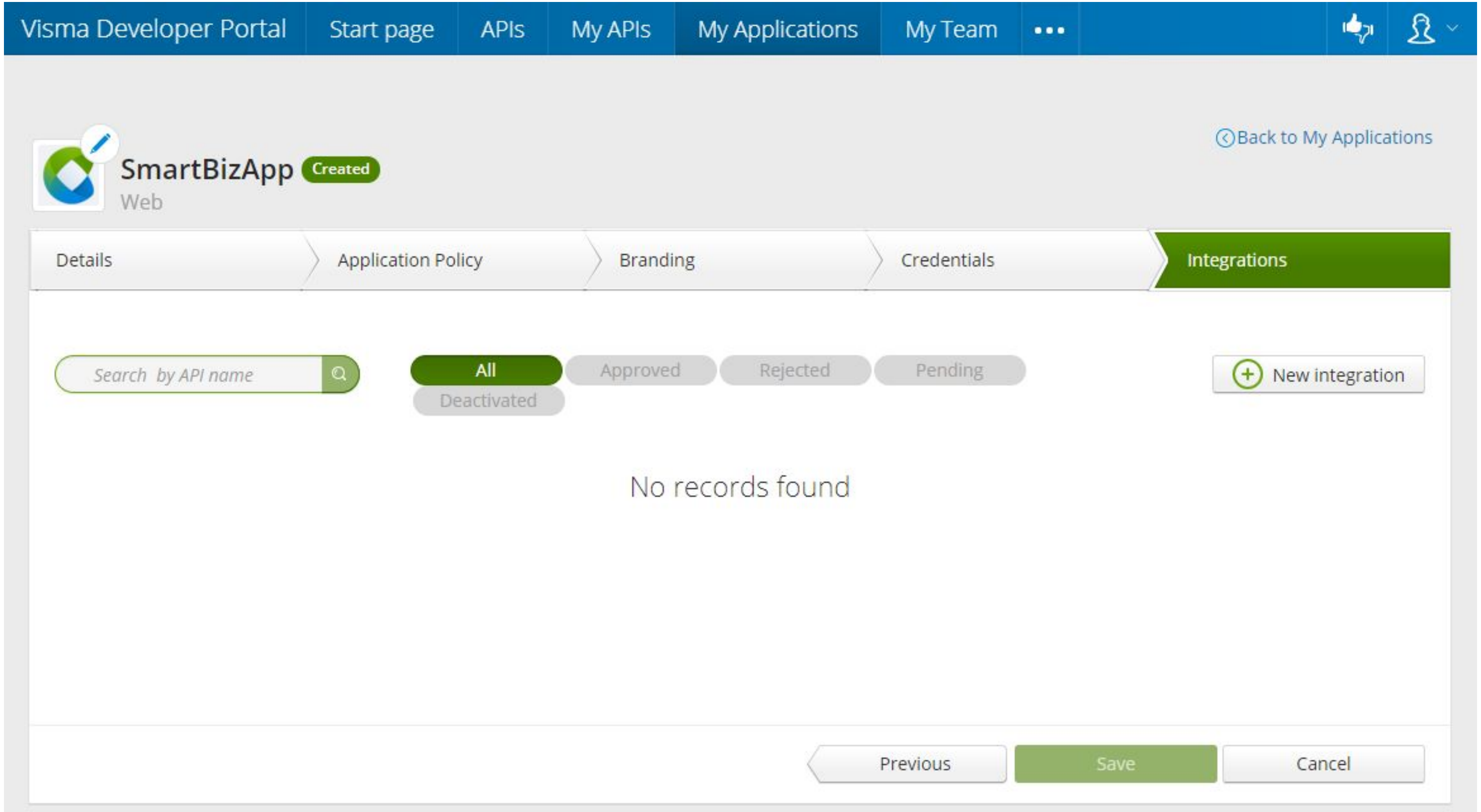
○ Require Consent

Setup flow - finish application registration



Name	Application Type	Client Id	Integrations	Status	
smart	All	Search...		All	
SmartBizApp	Web	smart_biz_app		Pending approval	

Setup flow - add new integration



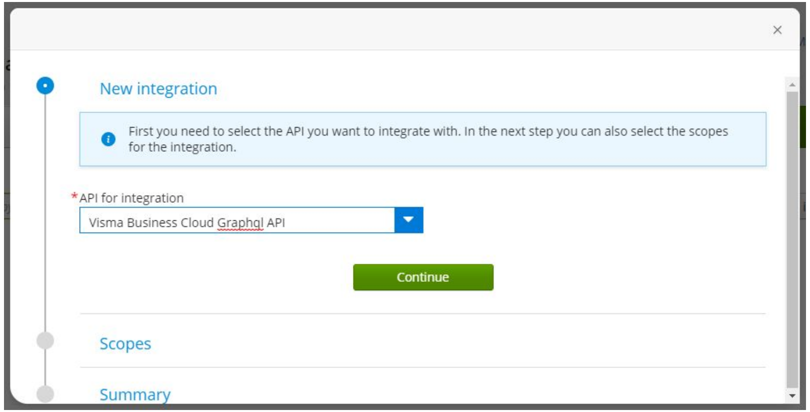
SmartBizApp Created

Web

Back to My Applications

No records found

Setup flow - select API and scopes

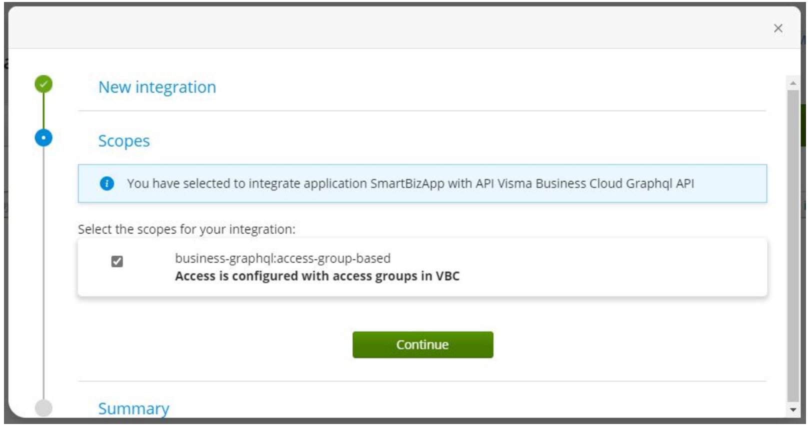


New integration

First you need to select the API you want to integrate with. In the next step you can also select the scopes for the integration.

*API for integration

Visma Business Cloud Graphql API



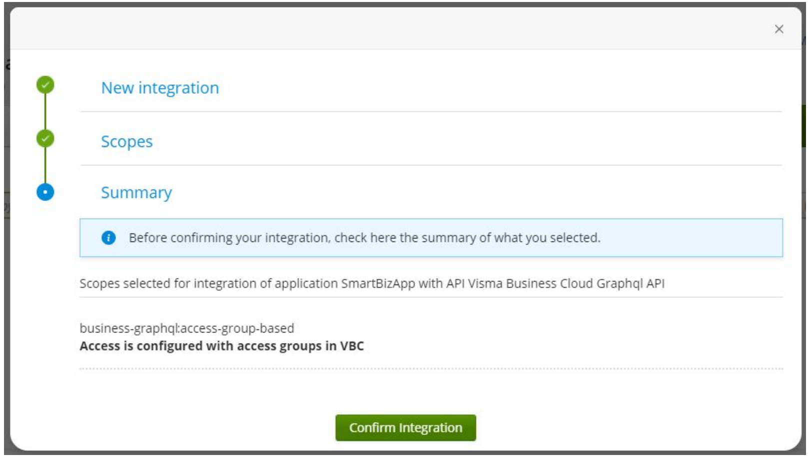
New integration

Scopes

You have selected to integrate application SmartBizApp with API Visma Business Cloud Graphql API

Select the scopes for your integration:

☑ business-graphql:access-group-based
Access is configured with access groups in VBC



New integration

Scopes

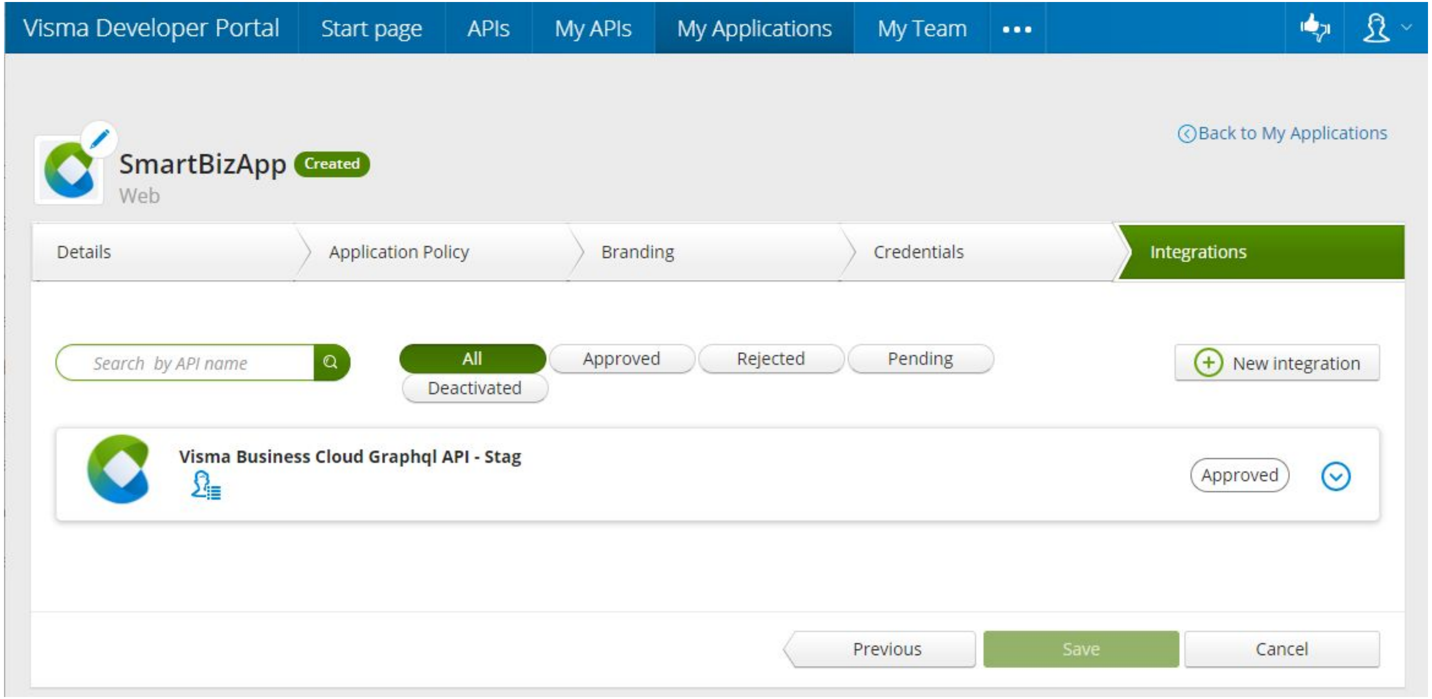
Summary

Before confirming your integration, check here the summary of what you selected.

Scopes selected for integration of application SmartBizApp with API Visma Business Cloud Graphql API

business-graphql:access-group-based
Access is configured with access groups in VBC

Setup flow - integration approved



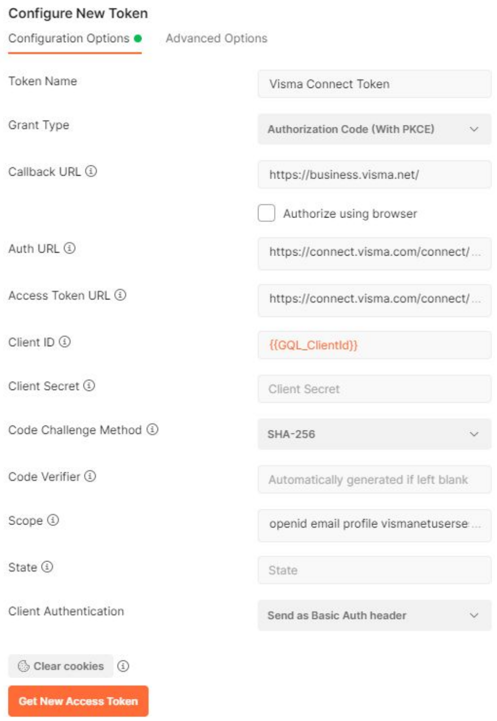
Authentication

Authentication flow

1. Setup the OAuth2 flow
2. Authenticate to Visma Connect and get the authorization token
3. Pass the token in the *Authorization* header in all queries

Authentication setup (Postman)

```
Type                   : OAuth 2.0
Add auth data to       : Request headers
Header prefix          : Bearer

Configuration Options  :
Grant Type             : Authorization Code (With PKCE)
Callback URL           : https://${YOUR_APP_DOMAIN}/
Auth URL               : https://connect.visma.com/connect/authorize
Access token URL       : https://connect.visma.com/connect/token
Client ID              : ${YOUR_CLIENT_ID}
Code Challenge Method  : SHA-256
Scope                  : openid email profile
                         business-graphql-api:access-group-based
Client authentication  : Send as basic auth header
```



Configure New Token

Configuration Options ● Advanced Options

Token Name: Visma Connect Token

Grant Type: Authorization Code (With PKCE)

Callback URL ⓘ: https://business.visma.net/

□ Authorize using browser

Auth URL ⓘ: https://connect.visma.com/connect/...

Access Token URL ⓘ: https://connect.visma.com/connect/...

Client ID ⓘ: {{GQL_ClientId}}

Client Secret ⓘ: Client Secret

Code Challenge Method ⓘ: SHA-256

Code Verifier ⓘ: Automatically generated if left blank

Scope ⓘ: openid email profile vismanetuserse...

State ⓘ: State

Client Authentication: Send as Basic Auth header

Granting access

Access is granted the same way as normal users through Visma.net Admin.

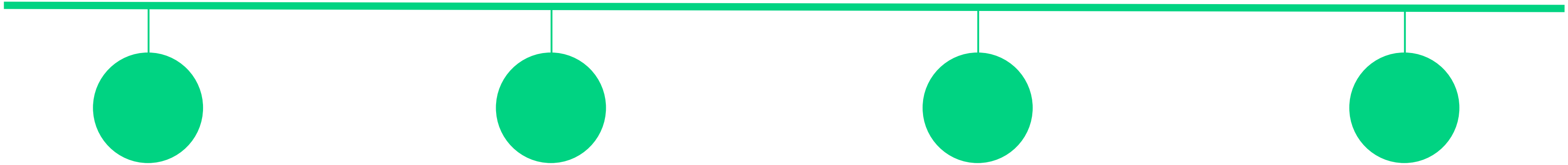
The integration users:

- Should not have the ***application access*** role
- Could in theory use the same access groups as other users
- **BEST practice**:
  - create a new access group that only grants access to the data needed by the integration
  - assign that access group to the integration user on the companies it should have access to (in Visma.net Admin).

Roadmap

Marko

Roadmap



Q4 2021
Pilot (1st wave)

- UI & Design improvements
- Workflow related functionality (Document viewer and setting approver manually)
- Layout management
- User documentation and onboarding (WalkMe) guides
- Multi-language support
- MDM integration

Q1 2022
Pilot (2nd wave)

- Stabilisation and Optimisations (Core, Database, API, Functionality)
- Functionality required by pilot customers & partners
- Scheduled jobs
- Task overview
- Bizweb integration

Q2 2022
Soft-launch

- Stabilization and Optimizations (Core, Database, API, Functionality)
- Visma Cloud Delivery Model (VCDM) approval
- Adaptation to new business models (consumption drivers and transaction counting)
- Streamline integrations

Q3-Q4 2022
Release

- P2P and O2C process optimisations
- Streamline integrations
- Data Model Extension (DME)
- AI and Machine learning initiatives: invoice automation and deviation handling, fraud detection
- Business Alerts

API roadmap

Support for running processings

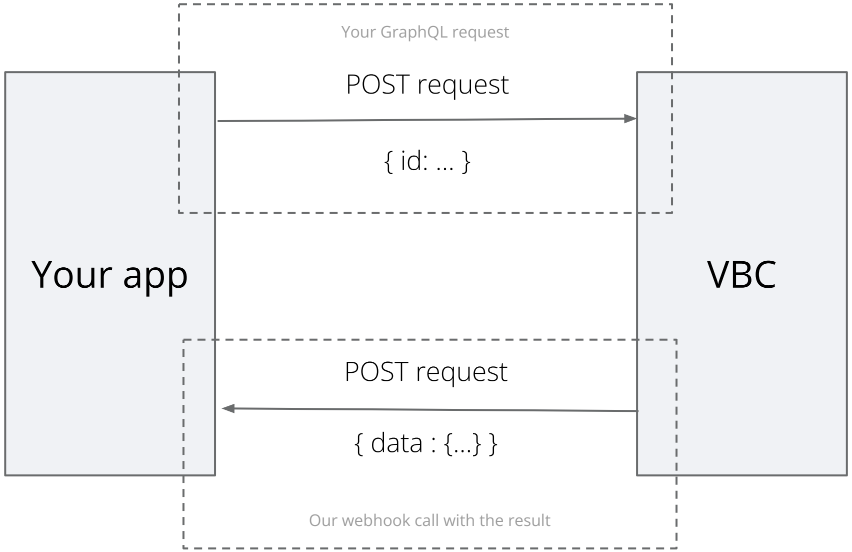
Webhooks

What are processings

- Business logic operations executed in the back-end
  - Add/send/split/cancel/finish order, add attachment, create document
  - Create batch, update batch, validate batch
  - Create voucher
  - Regenerate stock balance
- Potentially long-running operations

What are webhooks

- Application-defined HTTP callback
- Registered by your application
- Called by the system when an event occurs



How to get access

Alexandra

Demo environment

ISV demo environment has been set up and access can be provided on request

The environment will be shared:

- All ISVs under the same customer
- One company each
- Administered by Visma

Instructions: Send an email to Øyvind Årseth (oyvind.arseth@visma.com) - Company will then be created with demo data and access will be granted

Q&A

Entrepreneurial

Responsible

Dedicated

Inclusive