Guide: Moving from VNI authentication to Visma Connect Authentication for Visma.net ERP API

Introduction

Since the Visma.net ERP API was released authentication has been handled by an implementation of the OAuth 2.0 protocol called VNI. This has served us well, but it has some limitations that makes it somewhat cumbersome to work with both for you as developers and internally in Visma. Since the release of the Visma.net ERP API, the importance of APIs and number of integrations has grown rapidly and the amount of traffic we have received on the API has doubled year over year. More importantly, the requirements to securing the APIs are constantly increasing, and protecting our customers' data is always a top priority for us.

In the last few years we have developed our Visma Connect product into a state of the art implementation of the OAuth 2.0 protocol, including surrounding tools like the Visma Developer Portal and Visma App Store. Using Visma Connect as the authentication provider for the Visma.net ERP API is therefore an important step to keep your integrations and our customers' data secure.

Why should you switch?

There are multiple reasons why you should upgrade your applications to use the Visma Connect authentication provider.

- Visma Connect is more secure than VNI.
- Visma Connect supports the Client Credentials-flow, this makes it possible to create machine-to-machine integrations.
- Handling of your application is done self-service in the Visma Developer Portal, this includes creating new applications, managing credentials (ClientId and Secret), managing token lifetimes, scopes and much more.
- The new next generation APIs for Visma.net ERP, like the Sales Order Service uses Visma Connect, therefore when you switch to Visma Connect for your application you can reuse the same authentication.

Steps to switch to the Visma Connect authentication provider

Switching your application from using VNI to Visma Connect as the authentication-provider involves multiple steps, this guide aims to guide you through this process. The steps can be divided into two main parts;

1. Set up your application in the Visma Developer Portal.

2. Rewrite the authentication in your application to utilize Visma Connect.

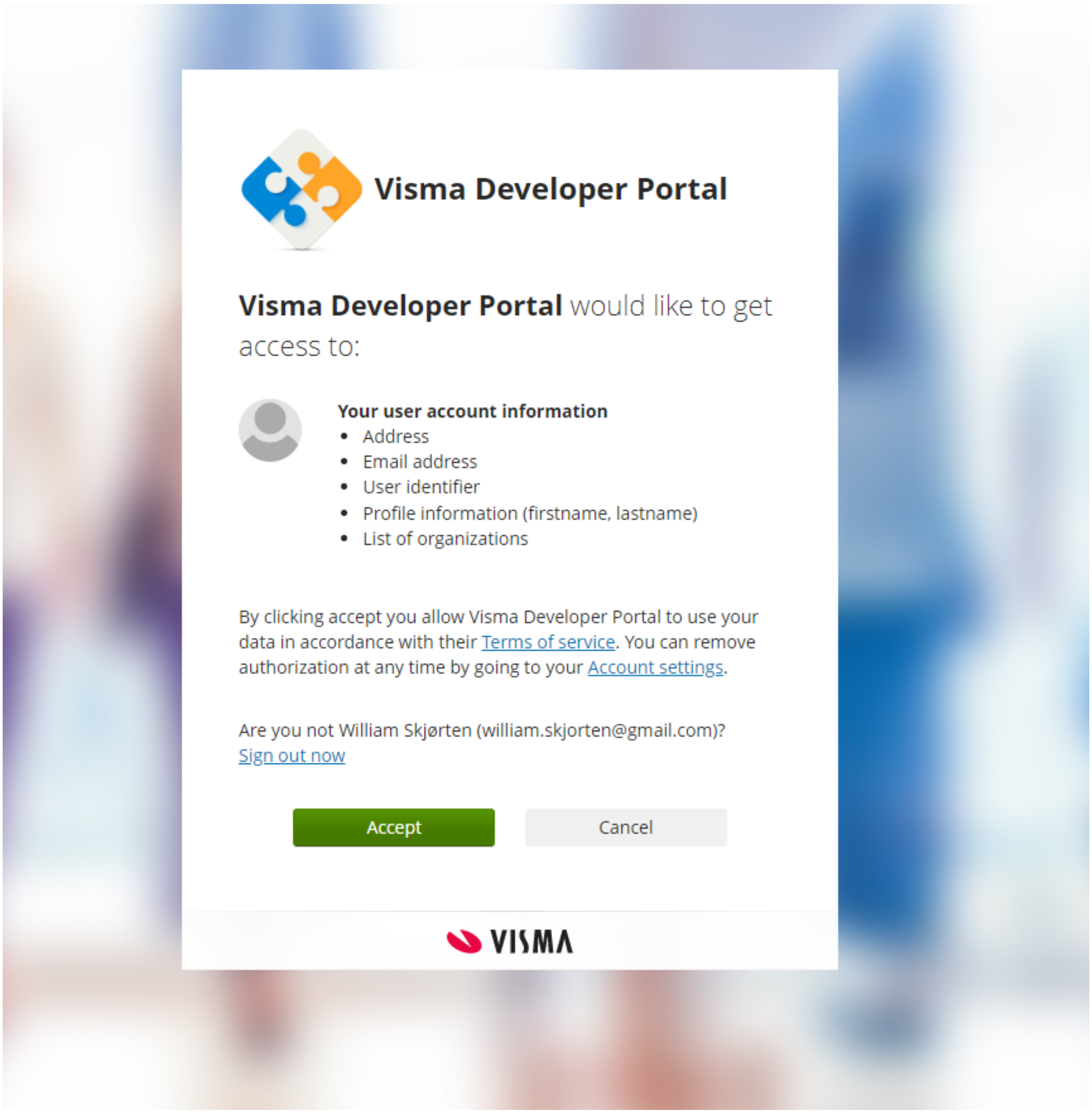
Both these parts involve choices you have to make, we will describe those choices in this guide.

Set up your application in Visma Developer Portal

An important part of the OAuth 2.0 protocol is that all applications (called clients in OAuth 2.0) that communicate with the APIs need to be registered with the Authentication Server (Visma Connect) before it can initialize the process. Visma Developer Portal is a self service tool that allows you as a developer to register and maintain your application.

Steps to register application

1. Log in to Visma Developer Portal at https://oauth.developers.visma.com/
2. Your user needs to be added to one or more teams in Visma Developer Portal to get access, if you are not part of a team you will be directed to the onboarding-page to onboard your company with Developer Portal.
   If you are part of a team you can skip to step 5.
3. To register your company with Developer Portal you must first accept that Developer Portal can access information from your user profile.



Visma Developer Portal

Visma Developer Portal would like to get access to:

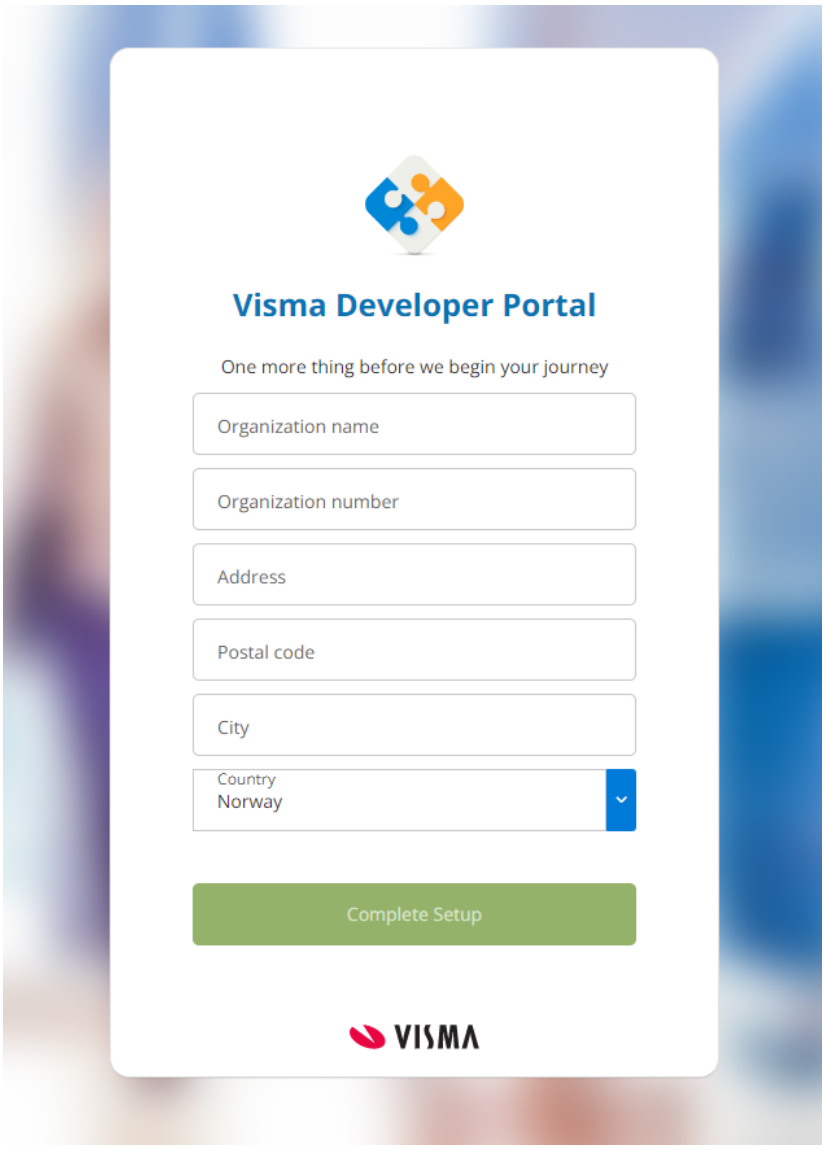
Your user account information

- Address
- Email address
- User identifier
- Profile information (firstname, lastname)
- List of organizations

By clicking accept you allow Visma Developer Portal to use your data in accordance with their Terms of service. You can remove authorization at any time by going to your Account settings.

Are you not William Skjørten (william.skjorten@gmail.com)? Sign out now

4. Next you have to register the required information about your company. The organization-number needs to be unique so if you get an error message that this is already registered, most likely someone else already registered your company and needs to add you as a member of the team. For more information about teams in Visma Developer Portal, you can read the guide Managing your team in Developer Portal.



Visma Developer Portal

One more thing before we begin your journey

Organization name

Organization number

Address

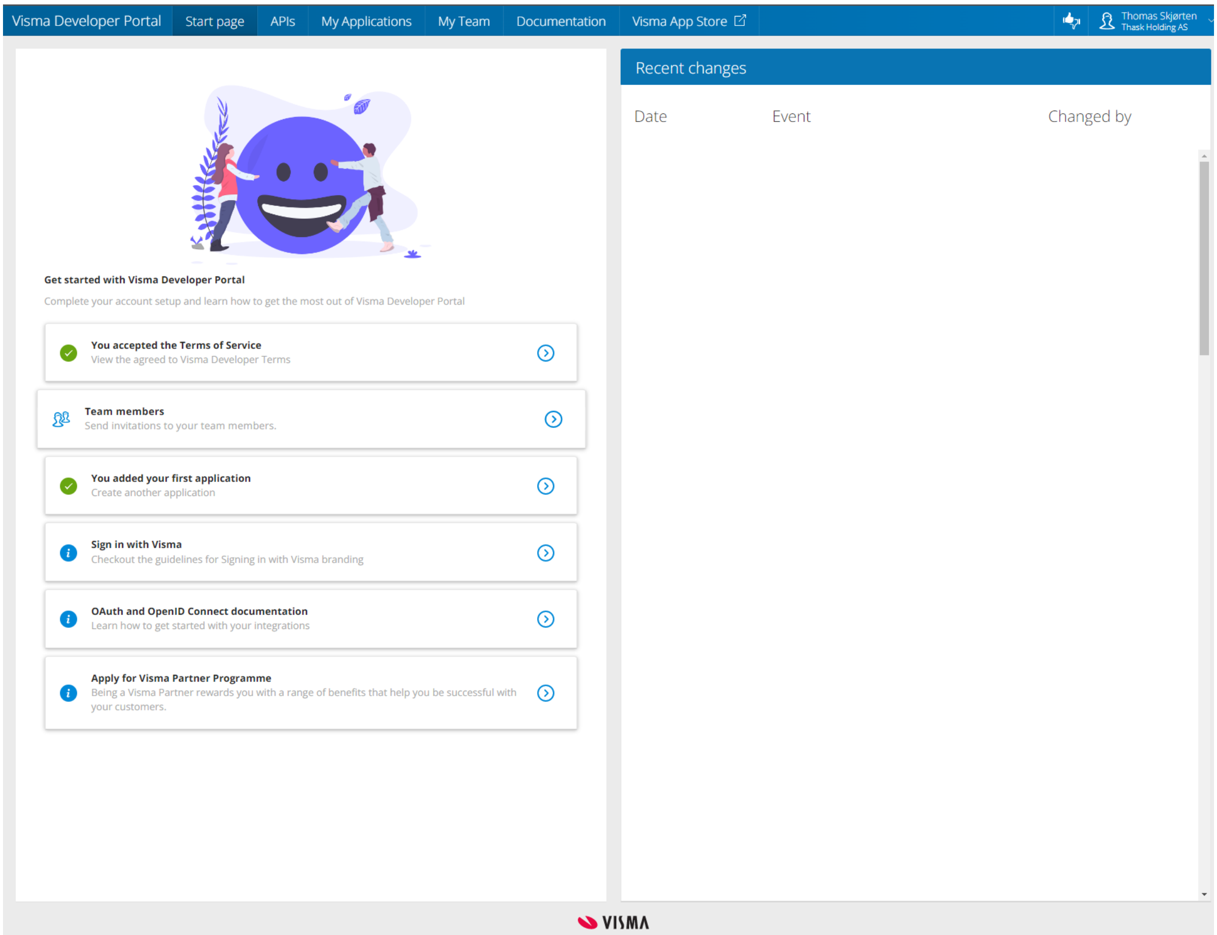
Postal code

City

Country
Norway

5. On the Developer Portal Start Page you will find links with useful information on how to get started with and use Developer Portal.
   The Developer Portal has 5 main components visible in the top main menu.

   APIs: Here you can find a list of all the Visma APIs that are available to use from within your application.
   My Applications: This is where you register and maintain your applications.
   My Team: This is where you manage your team, by adding team-members and managing roles.
   Documentation: Here you find documentation for the Visma Connect Authentication provider.



Get started with Visma Developer Portal

Complete your account setup and learn how to get the most out of Visma Developer Portal

You accepted the Terms of Service
View the agreed to Visma Developer Terms

Team members
Send invitations to your team members.

You added your first application
Create another application

Sign in with Visma
Checkout the guidelines for Signing in with Visma branding

OAuth and OpenID Connect documentation
Learn how to get started with your integrations

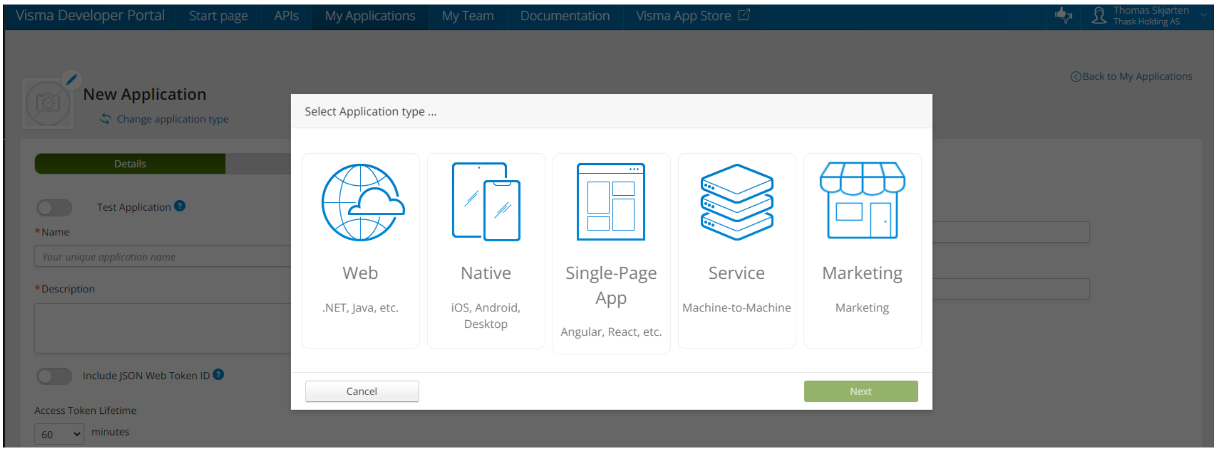
Apply for Visma Partner Programme
Being a Visma Partner rewards you with a range of benefits that help you be successful with your customers.

Recent changes

Date	Event	Changed by

6. To set up your application go to the **My applications**-section and press the **Add application** button.
7. The first thing you need to choose is what type of application you are creating. This is an important choice because it affects which OAuth 2.0 Flows are available to your application.

   Web: Standard web-application hosted by a web server or similar.
   Native: An application that runs natively on a device like a phone, or computer.
   Single-Page App (SPA): A web application hosted by the user web-browser.
   Service: A backend service, machine-to-machine integration without a user interface.
   Marketing: Special type of application used only to market applications on the Visma App Store.



Test Application

*Name

Your unique application name

*Description

Include JSON Web Token ID

Access Token Lifetime

60 minutes

Select Application type ...

Web
.NET, Java, etc.

Native
iOS, Android, Desktop

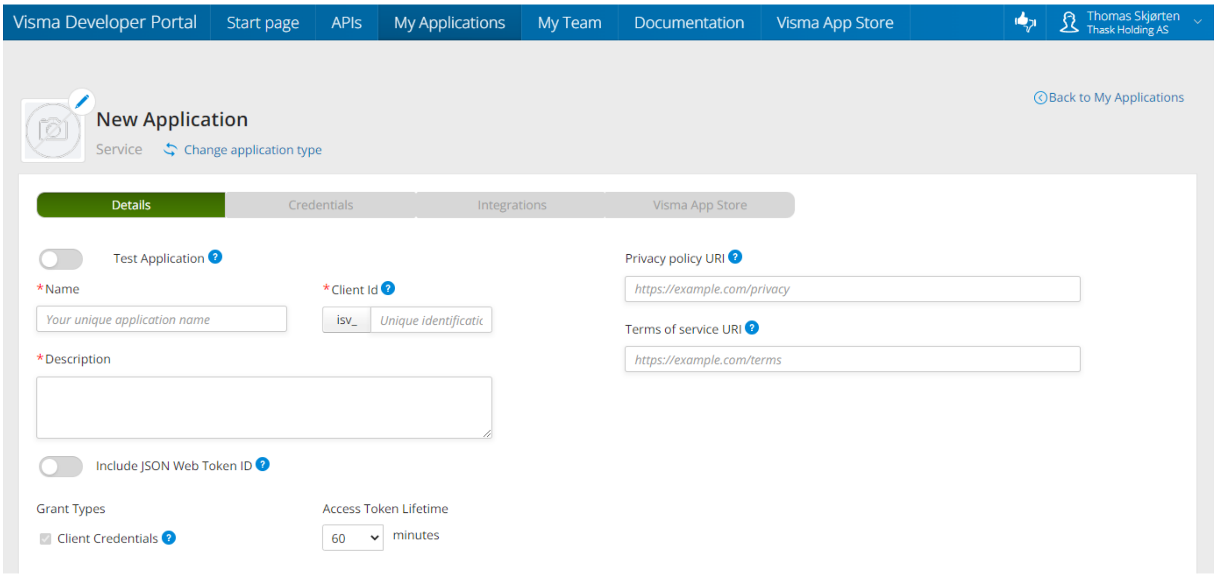
Single-Page App
Angular, React, etc.

Service
Machine-to-Machine

Marketing
Marketing

8. When you have chosen your application type a guide will open where you can supply the needed information to set up the integration. The fields required will be different per application type, so in this guide we will focus on two different types, **Service**-applications that are back-end applications without user interfaces and **Web** applications which is an application that presents a web interface to the user.

   To create a **Service**-application go to step 9.
   To create a **Web**-application go to step 10.
9. For service-type applications the only supported grant-type is the Client Credentials grant-type. This means that there is no user that is involved in accessing the API, and the token does not represent a specific user. This setup of the application is fairly simple with few values needed.



New Application
Service

Test Application
*Name: Your unique application name
*Client Id: isv_ Unique identificatic
*Description
Include JSON Web Token ID
Grant Types: ☑ Client Credentials
Access Token Lifetime: 60 minutes
Privacy policy URI: https://example.com/privacy
Terms of service URI: https://example.com/terms

Required information:

Name: Each application needs to have a name, choose a name that describes the application.
ClientId: This is the unique id of the application, this always starts with *isv_*. The Id needs to be unique.
Description: A description of your application.

Optional information:
Application icon: Your application should have an icon to easily identify it if you are publishing it on App Store or Visma Home. Change the icon by pressing the pencil-icon on the application icon.
Test Application: Set this to true if this is a test-application.
Access Token Lifetime: How long will the access token received live before becoming invalid (default 60 minutes, max 60 minutes)
Privacy policy URI: Link to a page that describes the privacy policy for this application, used on App Store.
Terms of service URI: Link to a page that describes the terms of service for this application, used on App Store.

To save the application, press **Save as draft** and then **Create**.

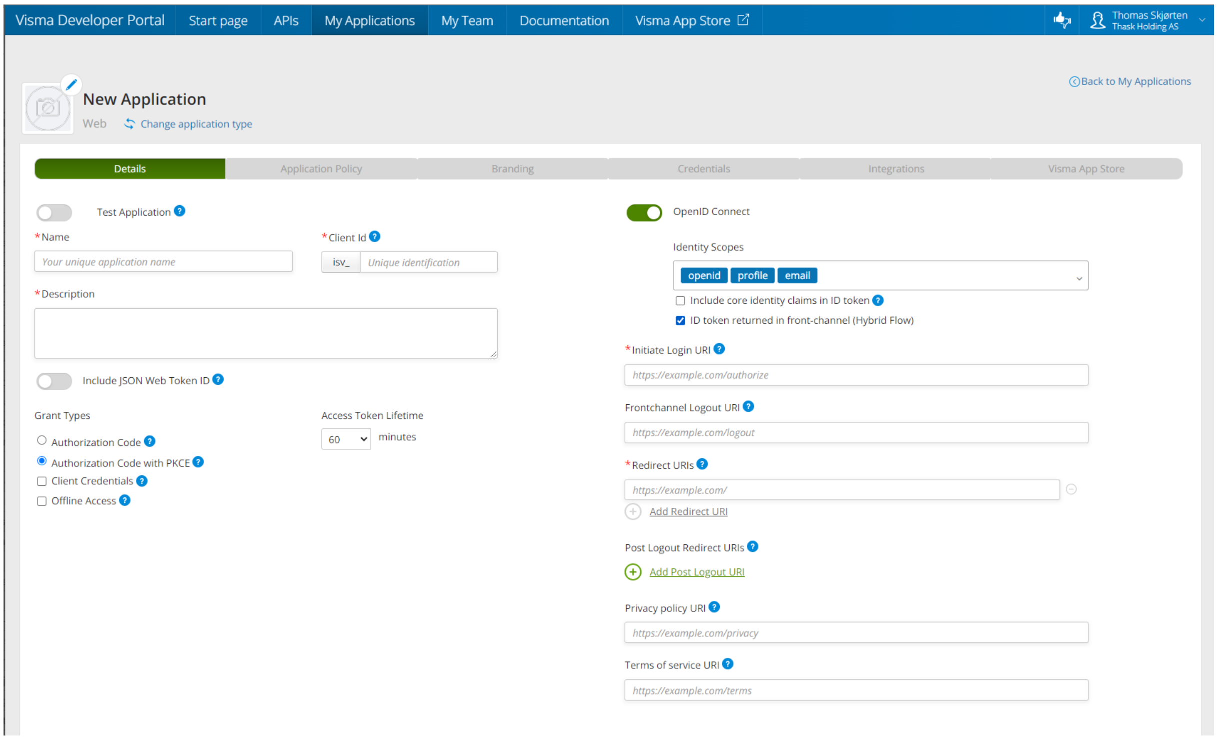
After this setup, continue to step 13, to get the application credentials.

10. For Web-type applications where the application is able to present a log-in screen to the user and create a token that impersonates that specific user, two grant-types are supported. Both are versions of the Authorization Code grant-type which shows a login-screen to the user and asks for the user's consent to access the data.

    Since the original specification of the OAuth 2.0 protocol it has been discovered that the Authorization Code grant can be compromised by an attach called the the authorization code interception attack, therefore an addition called the Proof Key for Code Exchange (PKCE) by OAuth Public Clients was added to the protocol in 2015.
    It is highly recommended to use the **Authorization Code with PKCE** grant for these types of clients.

    For web-type applications, it might also be relevant to use the Client credentials grant type in addition to Authorization code for parts of the application that does not require a token that impersonates a user. Enable this grant by activating the Client Credentials grant type.

    Offline Access
    When using the Authorization Code grant type, the user is required to log-in via the log-in screen to obtain a new token. However when the token expires, it might be relevant for applications to renew the token without the user having to log in again. For this we have the offline access mechanism, if offline access is enabled you will receive a refresh-token together with your access-token, you can use the refresh-token to request a new access-token when it expires.



New Application
Web
Change application type

Test Application
*Name: Your unique application name
*Client Id: isv_ Unique identification
*Description
Include JSON Web Token ID

Grant Types
○ Authorization Code
● Authorization Code with PKCE
□ Client Credentials
□ Offline Access

Access Token Lifetime: 60 minutes

OpenID Connect
Identity Scopes: openid profile email
□ Include core identity claims in ID token
☑ ID token returned in front-channel (Hybrid Flow)
*Initiate Login URI: https://example.com/authorize
Frontchannel Logout URI: https://example.com/logout
*Redirect URIs: https://example.com/
Add Redirect URI
Post Logout Redirect URIs
Add Post Logout URI
Privacy policy URI: https://example.com/privacy
Terms of service URI: https://example.com/terms

Required information:

Name: Each application needs to have a name, choose a name that describes the application.
ClientId: This is the unique id of the application, this always starts with *isv_*. The Id needs to be

unique.
Description: A description of your application.

<u>Optional information:</u>
Application icon: Your application should have an icon to easily identify it if you are publishing it on App Store or Visma Home. Change the icon by pressing the pencil-icon on the application icon.
Test Application: Set this to true if this is a test-application.
Access Token Lifetime: How long will the access token received live before becoming invalid (default 60 minutes, max 60 minutes).
Client Credentials: Allow the Client Credentials grant for your application.
Offline Access: Allow offline access for your application.

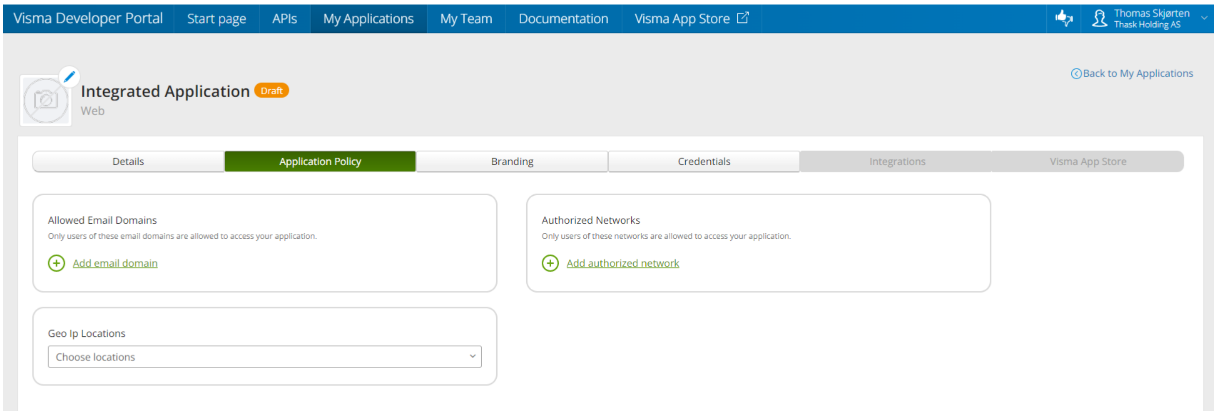
OpenID Connect: OpenID Connect (OIDC) is a thin layer on top of OAuth 2.0 that enables you to get information about the logged in user, like the username, email or other profile info. This can be enabled or disabeled for your application.

Initiate Login URI (Required when OIDC is enabled): An URI to the address of your application that initiates the Sign-in with Visma Connect process.
Frontchannel Logout URI (Required when OIDC is enabled): An URI to the address of your application that will be called by Visma Connect to log out the user.
Redirect URIs: Part of the flow for the Authorization Code grant type is to redirect the user's browser to an URI in your application that can receive the authorization code and run the required token-call to exchange this for an access-token. The valid redirect-URIs need to be known by the application before they can be used.
You can register multiple valid Redirect URIs.
Post Logout Redirect URIs: You can optionally define one or more URIs that the user can be redirected to after logout has been completed.

Privacy policy URI: Link to a page that describes the privacy policy for this application, used on App Store.
Terms of service URI: Link to a page that describes the terms of service for this application, used on App Store.

To save the application, press **Save as draft** and then **Create**.

11. For interactive-type applications like Web, you can set additional policies for controlling who can log in to the application. This is done on the **Application Policy**-tab.



Integrated Application Draft
Web

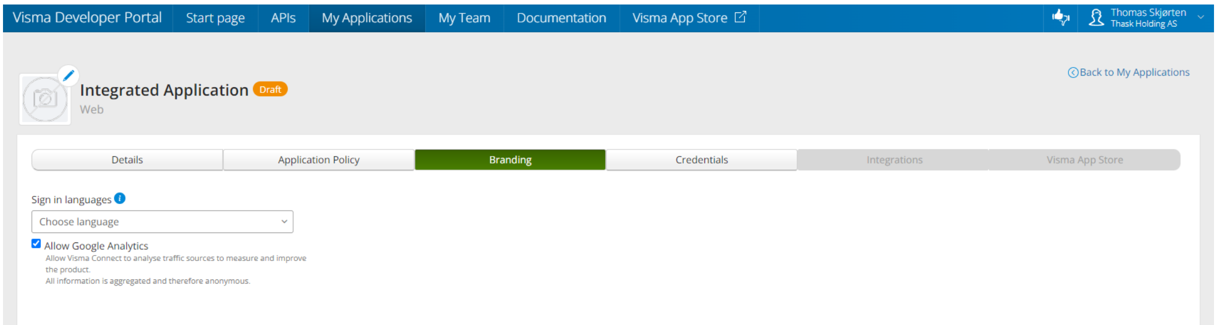
Allowed Email Domains
Only users of these email domains are allowed to access your application.

Authorized Networks
Only users of these networks are allowed to access your application.

Geo Ip Locations

Here you can choose to allow only logins for users with a specific email-domain, users that are part of a particular network (range of ip addresses) or users that are located in specific geographical locations (based on their IP-address).

12. For interactive-type applications like Web, you can also set some branding-settings under the **Branding**-tab.



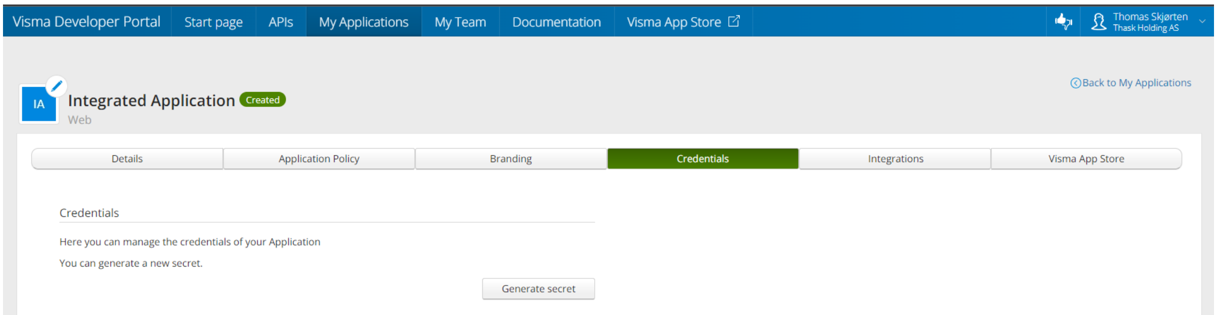
Sign in languages

Choose language

☑ Allow Google Analytics
Allow Visma Connect to analyse traffic sources to measure and improve the product.
All information is aggregated and therefore anonymous.

Here you can choose which languages should be supported in the sign-in page.
You can also choose to disable Google Analytics for the application, but we highly recommend to allow this as it gives valuable feedback to Visma Connect to allow for further improvement of the product.

13. Next you need to get your application's credentials. The credentials are two parts, first the ClientID which you created in the **Details**-tab, next a Client secret which is used as part of the authentication process.
The Client secret is confidential and private to your application, and should never be shared with anyone.

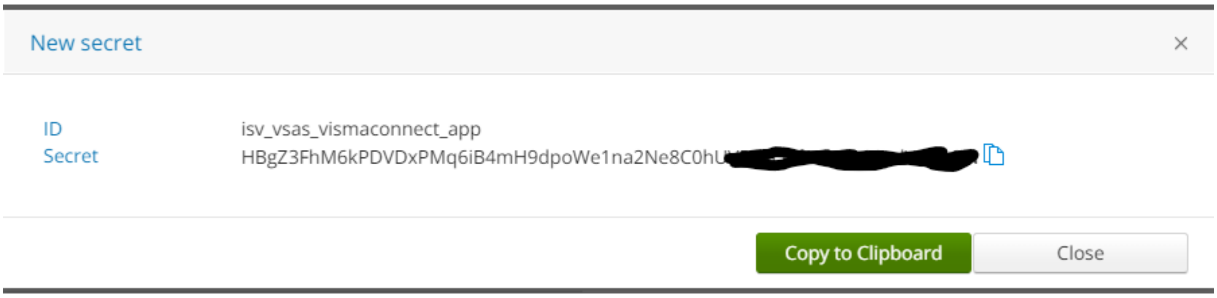


Credentials

Here you can manage the credentials of your Application

You can generate a new secret.

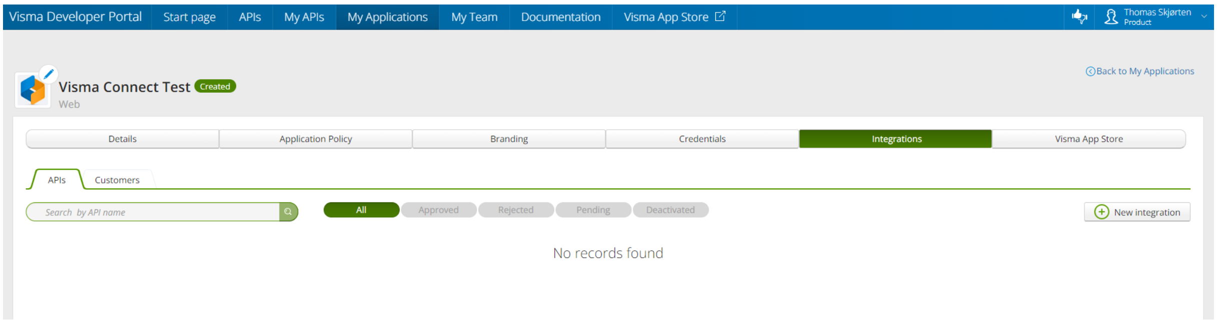
You can generate one or more secrets by pressing the **Generate secret** button.



ID	isv_vsas_vismaconnect_app
Secret	HBgZ3FhM6kPDVDxPMq6iB4mH9dpoWe1na2Ne8C0h[redacted]

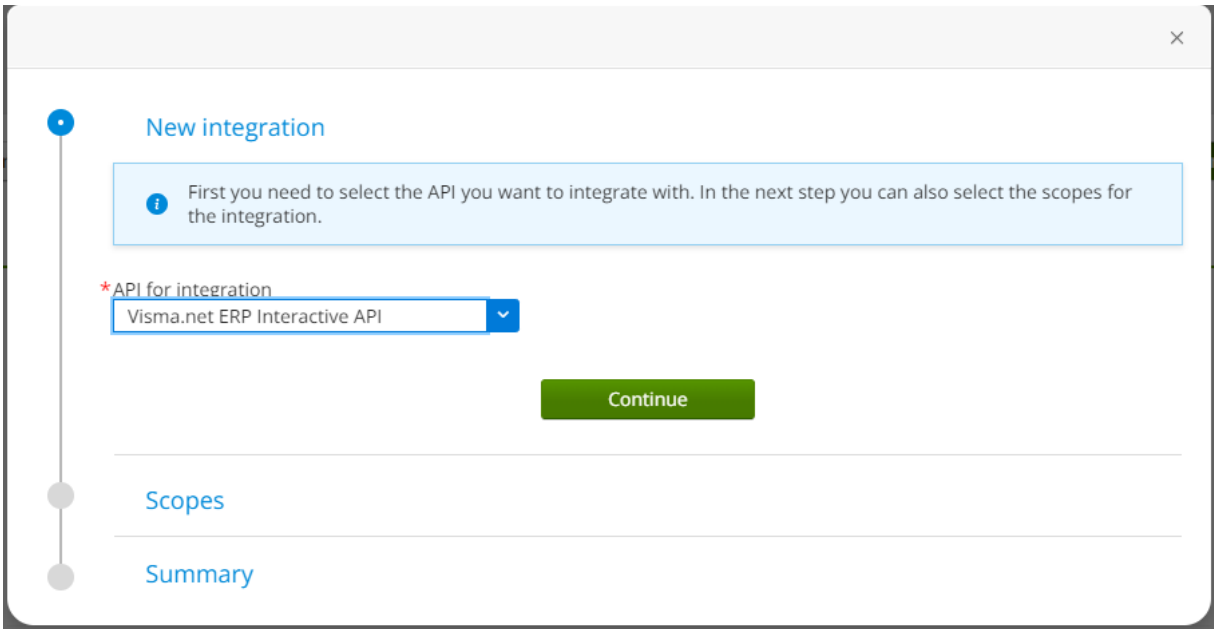
The Client ID and secret are shown in a new window, store them in a safe place and continue.
This is the only time the secret will be visible, you will not be able to retrieve it.

14. The next step is to integrate your application to the required Visma APIs. This is done on the **Integrations**-tab.



No records found

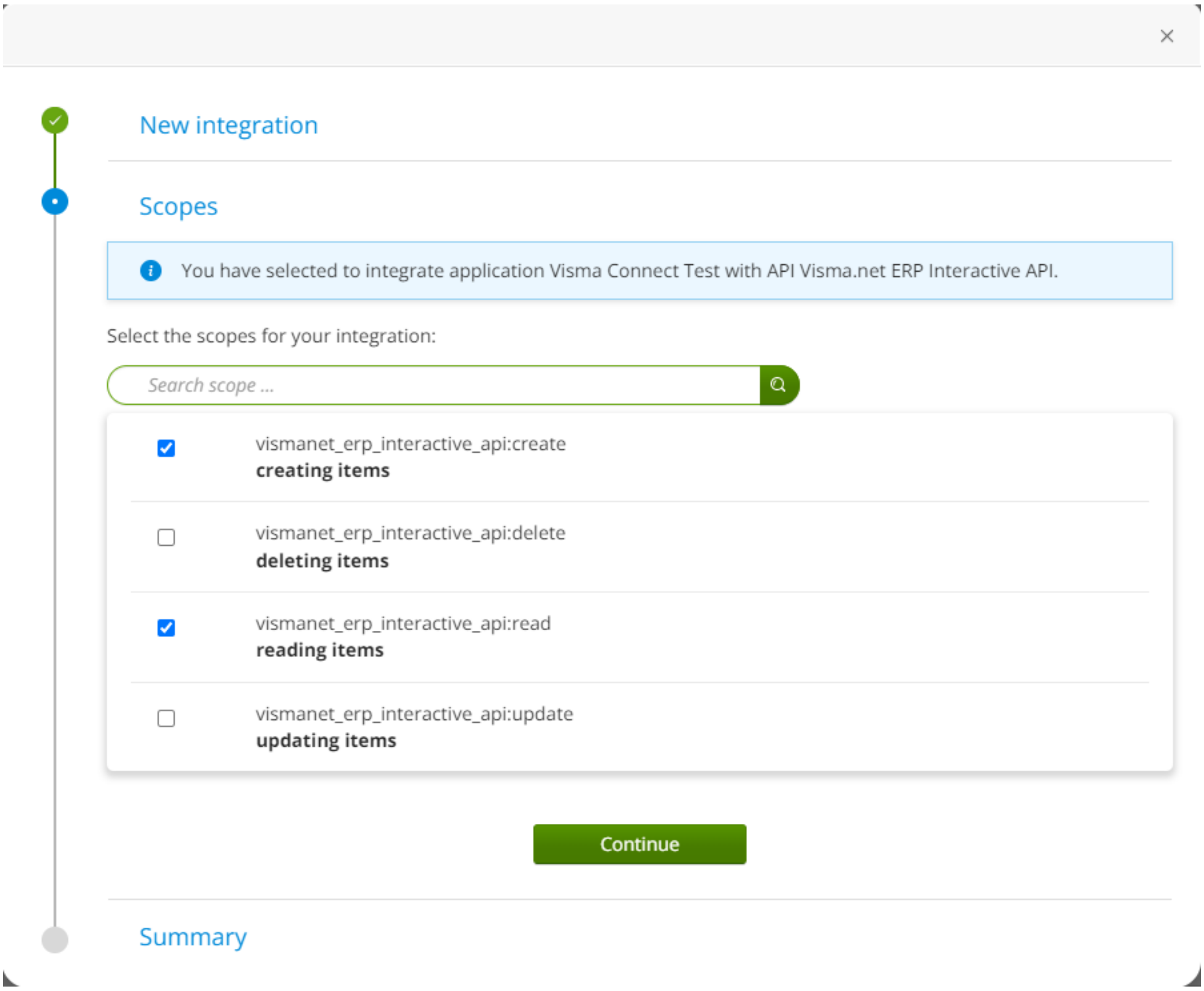
To a new integration press the **New integration** button, this will bring up a new screen that guides you through the process of adding the API-integration.



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.net ERP Interactive API

In the first step you choose the API you want to integrate to. The list of APIs is filtered to only show the APIs that are available for the grant-types that are selected for your application.



New integration

Scopes

You have selected to integrate application Visma Connect Test with API Visma.net ERP Interactive API.

Select the scopes for your integration:

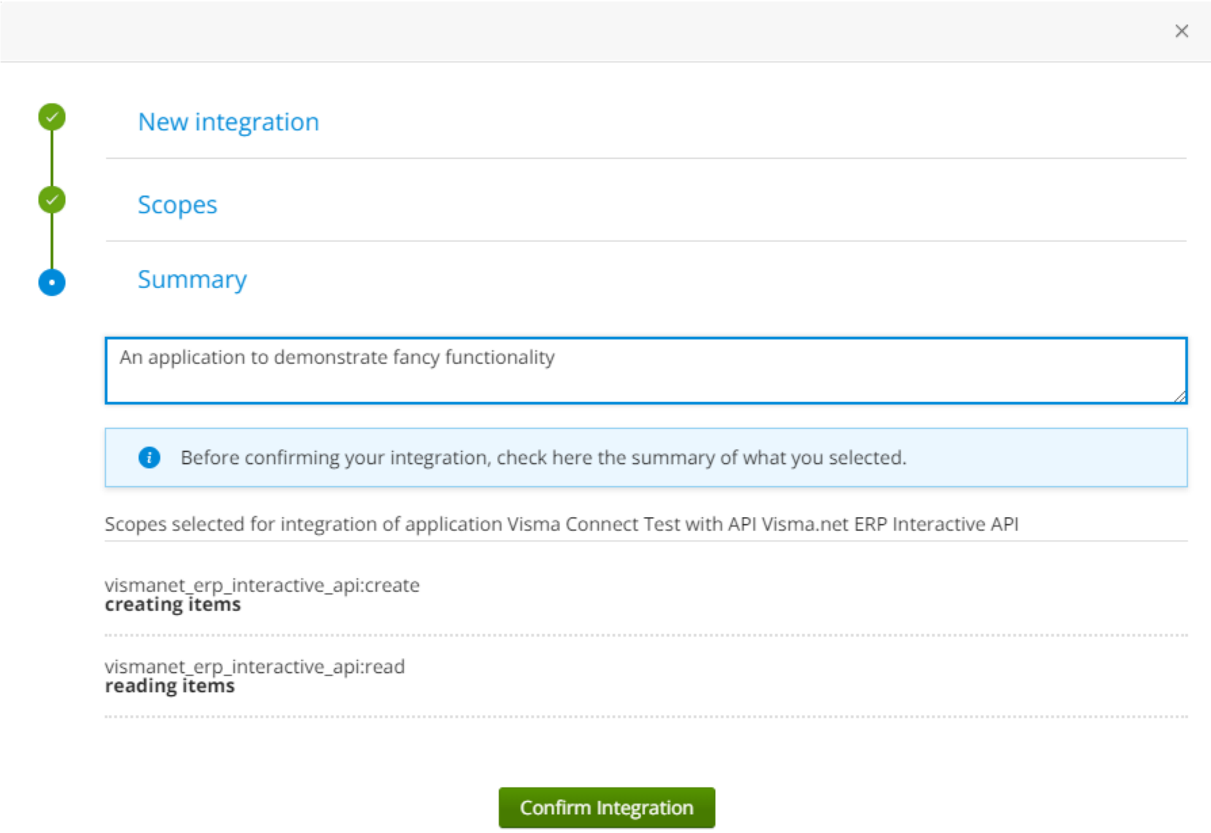
☑ vismanet_erp_interactive_api:create
creating items

□ vismanet_erp_interactive_api:delete
deleting items

☑ vismanet_erp_interactive_api:read
reading items

□ vismanet_erp_interactive_api:update
updating items

After choosing the API you need to choose the scope that your application will require to use in the API. For the Visma.net ERP API there are four scopes; Create, Read, Update and Delete.



An application to demonstrate fancy functionality

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

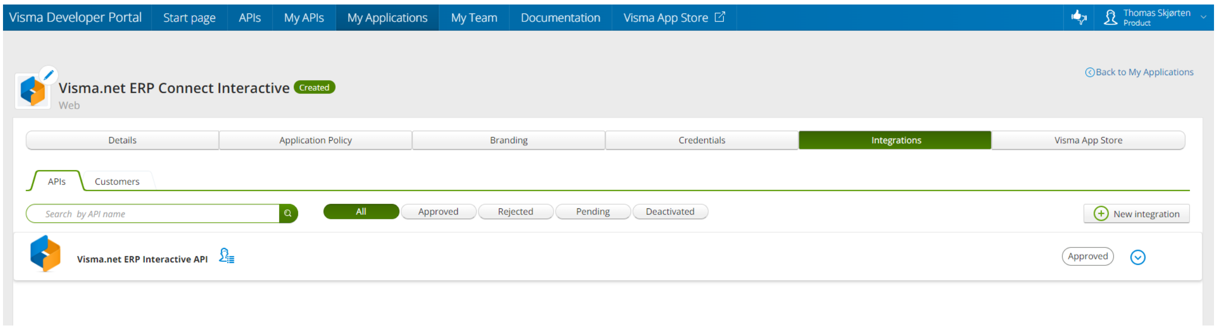
Scopes selected for integration of application Visma Connect Test with API Visma.net ERP Interactive API

vismanet_erp_interactive_api:create
creating items

vismanet_erp_interactive_api:read
reading items

In the last step you will find a summary of the information you've given, plus a field where you can give a brief description of what your application will do with the API. This information is given to the Visma-team that will approve the integration.

After you've pressed **Confirm integration**, the request for approval will be sent to Visma which will approve that your application can access the API. While this approval is taking place the integration will be visible under the Integrations-tab with status **Pending**.
Once the integration has been approved the integration will be marked as **Approved**.



It is important to note that your application can be integrated to more than one API. For example it would be normal for a Visma.net integration to use both the Visma.net ERP API and the SalesOrder Service API.

15. Once all your API-integrations have been approved, you are now ready to develop your application. The Visma Developer Portal will be the place where you can manage and make changes to your

application.

16. The last step of the process is to publish your application to the Visma App Store. Publishing the application has multiple functions, and depending on the grant types for your application it might be important for your application to work.

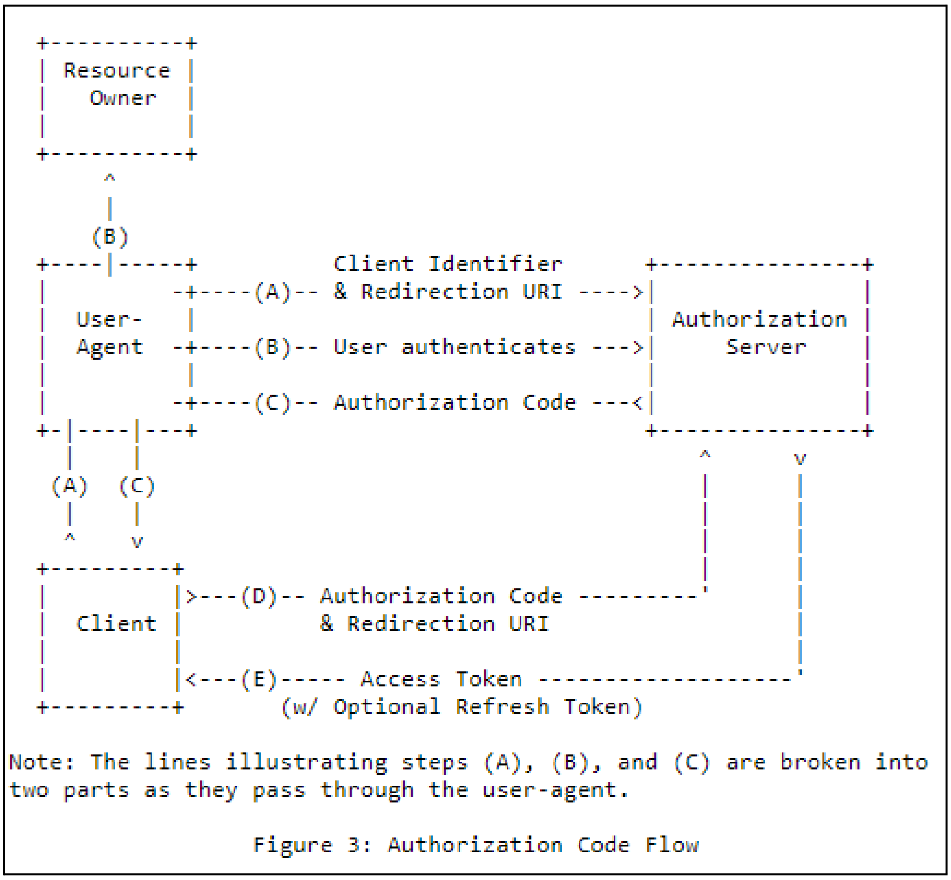
Rewrite the authentication in your application to utilize Visma Connect

Since both VNI and Visma Connect are implementations of the OAuth2.0, the implementation in your code is fairly similar. There are however some differences, and also some new opportunities by using Visma Connect.
Inside the Visma Developer Portal there is a documentation-section that describes the process of implementing the authentication flow for different types of application, we highly recommend that you make yourself familiar with this.
In this section of the guide we will briefly explain the differences in implementing the authentication flow for interactive type applications like Web, and we will also go through how you can implement the flow for non-interactive service applications that are a new possibility with Visma Connect.

Interactive applications

As discussed earlier in this guide, interactive applications are applications that have a natural place to have the user login and give their consent to the application accessing their Visma.net ERP data. These types of integration will impersonate the logged in user.

The OAuth2.0 Authorization Code grant-type has the following flow (figure from the OAuth2.0 rfc):



```
+----------+
| Resource |
|   Owner  |
|          |
+----------+
     ^
     |
    (B)
+----|-----+          Client Identifier      +---------------+
|         -+----(A)-- & Redirection URI ---->|               |
|  User-   |                                 | Authorization |
|  Agent  -+----(B)-- User authenticates --->|     Server    |
|          |                                 |               |
|         -+----(C)-- Authorization Code ---<|               |
+-|----|---+                                 +---------------+
  |    |                                         ^      v
 (A)  (C)                                        |      |
  |    |                                         |      |
  ^    v                                         |      |
+---------+                                      |      |
|         |>---(D)-- Authorization Code ---------'      |
|  Client |          & Redirection URI                  |
|         |                                             |
|         |<---(E)----- Access Token -------------------'
+---------+       (w/ Optional Refresh Token)
```

Note: The lines illustrating steps (A), (B), and (C) are broken into two parts as they pass through the user-agent.

Figure 3: Authorization Code Flow

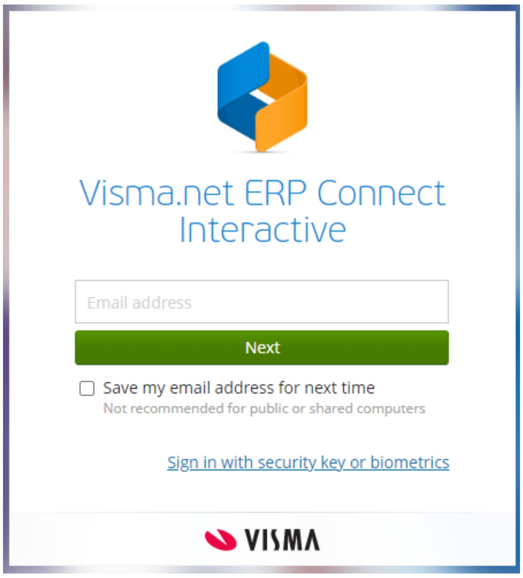
This flow is the same as the one used in VNI and consists of the following steps:

1. Your application constructs an URI to the Authentication Server (Visma Connect) authorize-endpoint, including the ClientID, the required scopes and the RedirectURI. Your application then redirects the users browser to this URI which displays the login-screen to the user and asks the user for consent to access the data.

   Differences between VNI and Visma Connect

   a. The address to the authorize-endpoint is now https://connect.visma.com/connect/authorize.
   b. In VNI we had just one scope (financialstasks), in Visma Connect we have more granular scopes (create, read, update, delete) so you have to supply the scopes needed by your application.

c. The login-screen is new, now using Visma Connect.



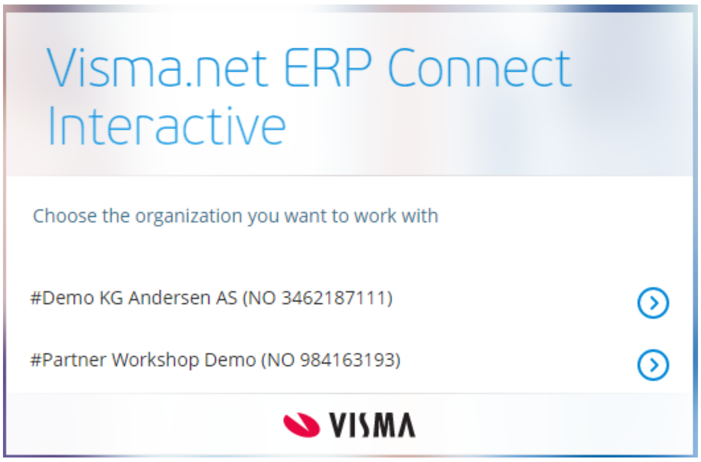
Visma.net ERP Connect Interactive

Email address

□ Save my email address for next time
Not recommended for public or shared computers

Sign in with security key or biometrics

d. You now select the tenant/company as part of the authentication-process. If the user only has access to one tenant, this will be used as default. If the user have access to multiple tenant the following screen will be show as part of the authentication process:



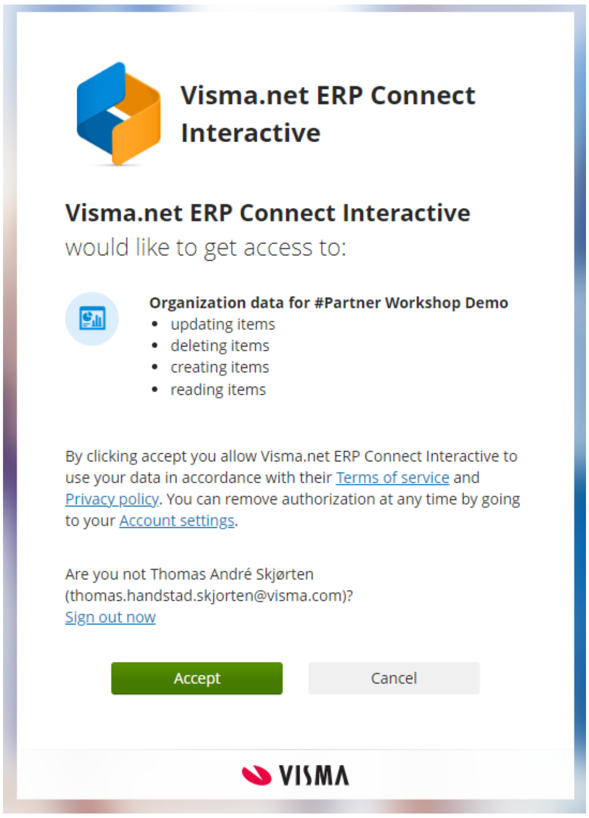
Visma.net ERP Connect Interactive

Choose the organization you want to work with

#Demo KG Andersen AS (NO 3462187111)

#Partner Workshop Demo (NO 984163193)

e. The consent-screen is new, now using Visma Connect. The screen also provides information about the scopes required by the application.



Visma.net ERP Connect Interactive

Visma.net ERP Connect Interactive
would like to get access to:

Organization data for #Partner Workshop Demo
- updating items
- deleting items
- creating items
- reading items

By clicking accept you allow Visma.net ERP Connect Interactive to use your data in accordance with their Terms of service and Privacy policy. You can remove authorization at any time by going to your Account settings.

Are you not Thomas André Skjørten (thomas.handstad.skjorten@visma.com)?
Sign out now

f. In Visma Connect we support the Authorization Code with PKCE flow. This means that you should calculate a random code-verifier string that you supply to the authorize-endpoint.

2. When the user has logged in and given its consent, the Authentication Server redirects the browser to the given RedirectURI including a code.

Differences between VNI and Visma Connect

a. The login and consent-screen is now different, but it serves the same purpose.

b. With VNI the user that logged in needed to have the role “Financials User” in the Visma.net Company in order to be able to authenticate. This has now been replaced by a new role called “API User” that the user has to have to be able to authenticate. The roles are given in Visma.net Admin.

3. Your application responds to the RedirectURI by making a call to the Authentication Server token-endpoint with the received code, together with the ClientID and ClientSecret.

4. The Authentication Server responds with the access token that can be used for calling the API.

Differences between VNI and Visma Connect

a. The address to the token-endpoint is now https://connect.visma.com/connect/token.

b. The received token now expires according to the setting set in Developer Portal, max 60 minutes. After that a new token has to be acquired either by running the process again og by enabling offline-access which will give you a refresh-token that can be used to obtain a new access-token.

c. The new APIs are Tenant enabled, meaning that the access token contains information about what tenant (company) the token is allowed to access. In VNI the token could be used across tenants, and you had to supply the **ipp-company-id** header to each API-call to specify which context the call should be made in. This header is no longer needed or supported.

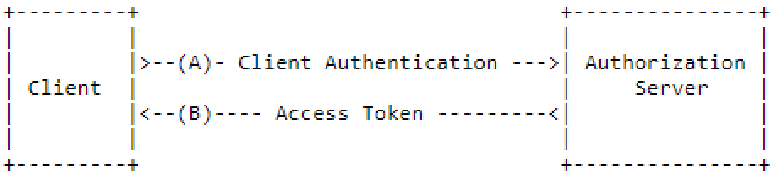
d. In Visma Connect we support the Authorization Code with PKCE flow. If this is enabled you need to provide the code-verifier string in the token-request.

We have created a demo-application that implements this functionality. The application with source code can be found at https://github.com/Visma-Software-AS-Product/PYTHON_Vnet_Connect_auth_interactive.

Non-Interactive applications / service-applications

A large majority of the application we see are using the Visma.net ERP API today are actually service-applications, meaning that they are backend-applications that does not have a natural place to present a login-screen to a user and does not need to run in the context of a given user. Therefore having the opportunity to authenticate using the Client Credentials grant type is a big difference for the VNI authentication method. We expect that a lot of the existing applications should switch to this authentication flow.

The OAuth2.0 Client Credentials grant-type has the following flow (figure from the OAuth2.0 rfc):



```
+---------+                                  +---------------+
|         |                                  |               |
|         |>--(A)- Client Authentication --->| Authorization |
| Client  |                                  |     Server    |
|         |<--(B)---- Access Token ---------<|               |
|         |                                  |               |
+---------+                                  +---------------+
```

Figure 6: Client Credentials Flow

The flow is fairly simple and involves just one call to the Authorization Server (Visma Connect):

1. Make a call to the token-endpoint including your ClientId, the ClientSecret, the scopes needed for your application and the tenantId of the Visma.net company you want to connect to.
2. Visma Connect returns the access-token you can use to access the API.

We have created a demo-application that implements this functionality. The application with source code can be found at https://github.com/Visma-Software-AS-Product/PYTHON_Vnet_Connect_auth_service.

As you can see in this flow there is no user being impersonated, and no user that is involved to give the consent for your application to access the data. This means that the consent has to be given before authenticating. This is done by the customer in the Visma App Store, see the section Giving consent to an application in App Store.

Rewrite the API-calls in your application to use the new tokens

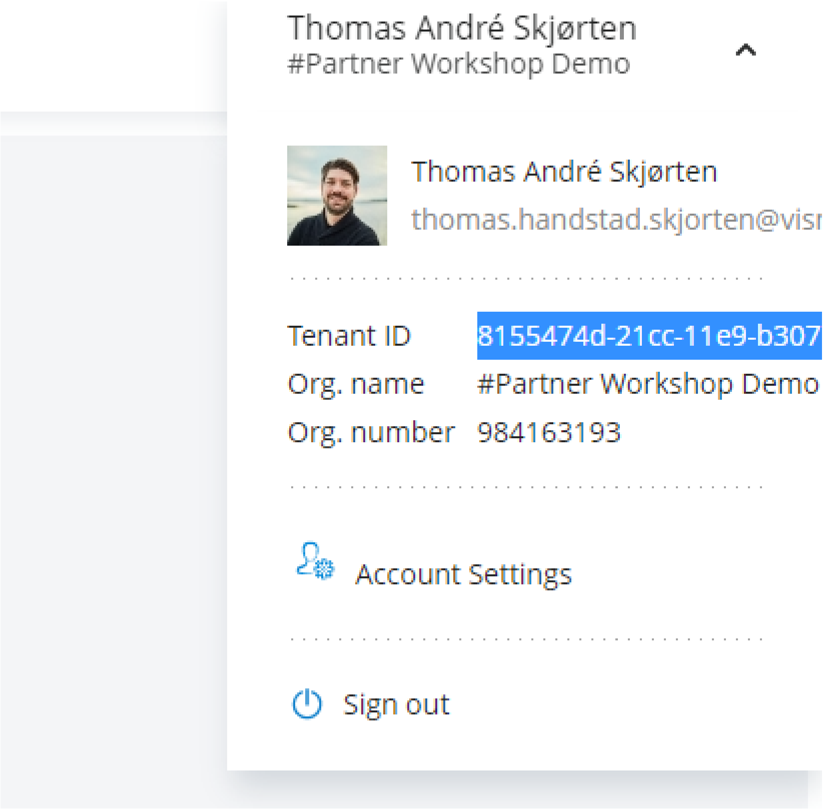
Once you have received your access-tokens you are ready to make API-calls. The procedure to call the Visma.net ERP API is largely the same, but there are some differences.

The access-token now contains the context (no more /context or ipp-company-id)

Previously you had to explicitly include the context you wanted the call to be made in, in other words for which company you were asking, this was done by including a **ipp-company-id** header in each API-call. This is no longer the case, the access-token now contains information about which context that is authorized.

For interactive applications using the authorization code grant, the user is presented with a context-selector screen when logging in and granting access. The context chosen here is the context included in the token. If the user only has access to one company, this will automatically be chosen as the context.

For non-interactive applications the context needs to be decided when creating the token, this is done by sending in the tenant-id as part of the token request. Then tenant-id is unique for each company and can be found by the customer in the Visma App Store.



Thomas André Skjørten
#Partner Workshop Demo

Thomas André Skjørten
thomas.handstad.skjorten@visr

Tenant ID 8155474d-21cc-11e9-b307
Org. name #Partner Workshop Demo
Org. number 984163193

The **ipp-application-type** header is no longer needed so this can be omitted.

Example call with new API

```
curl --request GET --url
https://integration.visma.net/API/controller/api/v1/inventory/?pageSize=10
--header 'Accept: application/json'
--header 'Authentication: Bearer [access_token]'
```

Giving consent to an application in App Store

The Visma.net ERP API gives your application access to the customers data and functionality in Visma.net ERP, therefore it is very important that the customer is always in charge of granting your application this access and have the ability to revoke the access if needed.
In interactive applications the user is presented with a consent-screen when authorizing your application access to the API, but for non-interactive applications the user is not involved in the authentication-process and thus needs to grant the access before your application can authenticate. This is done in the Visma App Store.

There are two different ways to allow a customer to find and activate your application in Visma App Store;

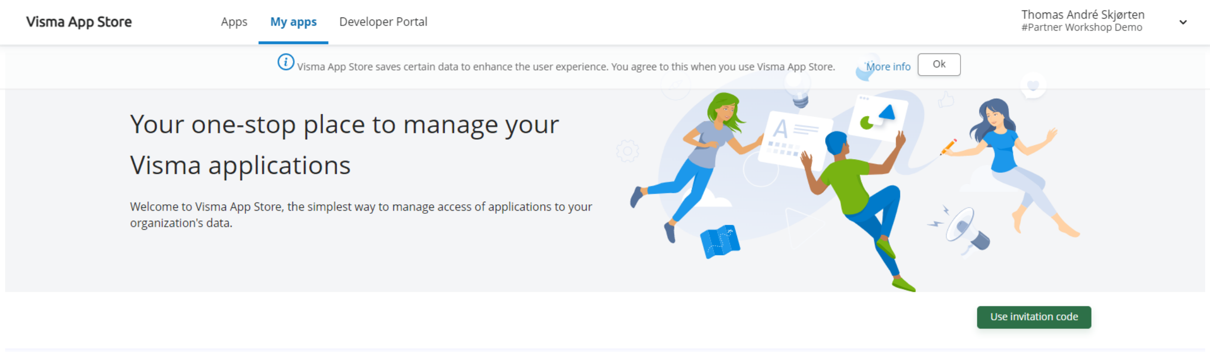
1. You can publish your application as a public application which is available to all customers in App Store.
2. You can create invitation-codes for your application, that you can share with the customer you want to activate your application.

Both of the alternatives are done in the Visma Developer Portal. To get more information about this see the section Publishing your application on Visma App Store.

As a customer, to manage which applications have access to your data in Visma App Store you have to log in to App Store with a user that has the role **Integration Administrator** for the **App Store** product of your company. These roles are set in Visma Admin.

Steps to give consent to an application

1. Log in to the App Store at https://apps.visma.com.

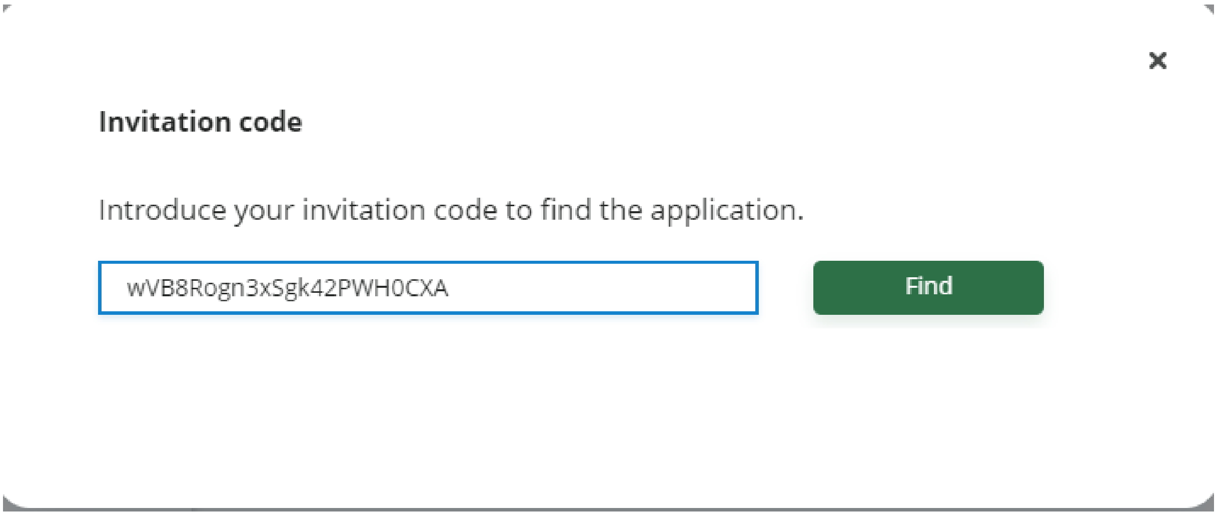


Visma App Store saves certain data to enhance the user experience. You agree to this when you use Visma App Store.

Your one-stop place to manage your Visma applications

Welcome to Visma App Store, the simplest way to manage access of applications to your organization's data.

2. Once logged in you will see the tab **My apps**. From here you manage the applications that have access to your company (if you have multiple companies you can switch in the context selector in the top-right corner).
3. For this example we use the scenario that you have been given an invitation code by the Application publisher. Press **Use Invitation code**.
4. In the pop-up, enter your invitation code and press **Find**

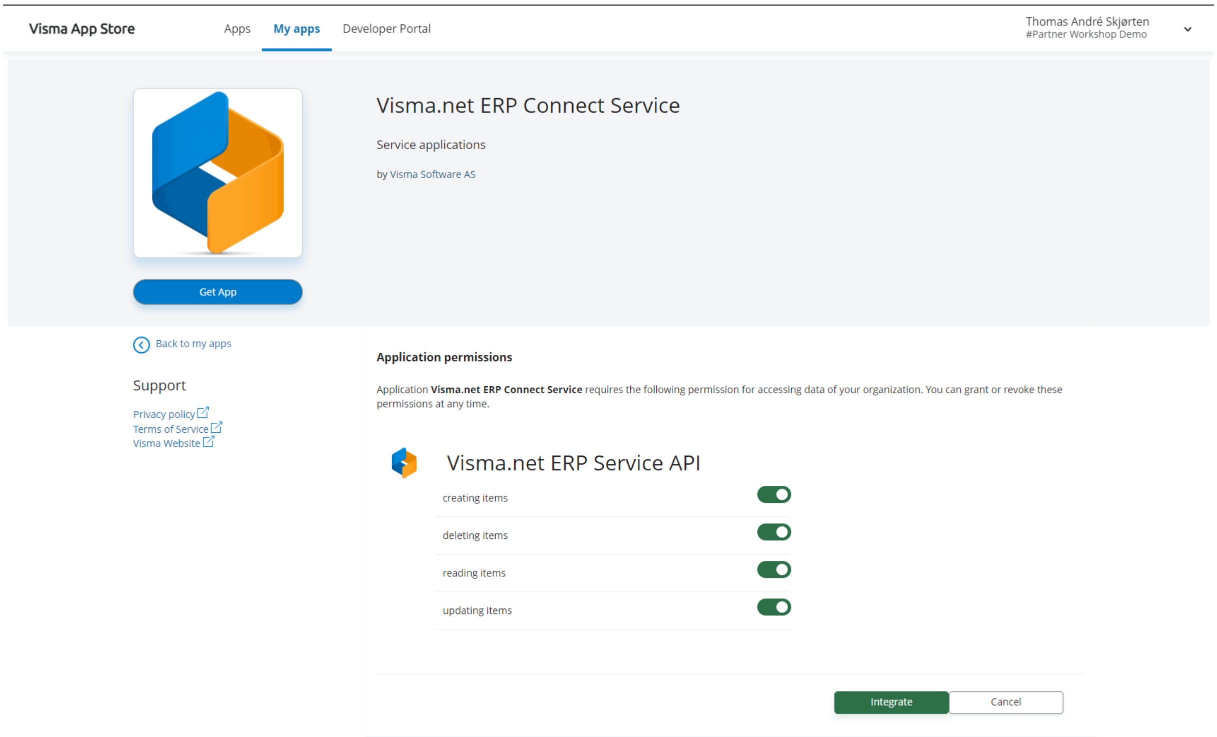


Invitation code

Introduce your invitation code to find the application.

wVB8Rogn3xSgk42PWH0CXA

5. You will be directed to the consent-page for the given application. This screen gives you some information about the application, plus a list of the permissions this application needs to perform its functionality.
If the application requires access to multiple APIs each API with the corresponding permissions will be listed.



Visma.net ERP Connect Service

Service applications

by Visma Software AS

Application permissions

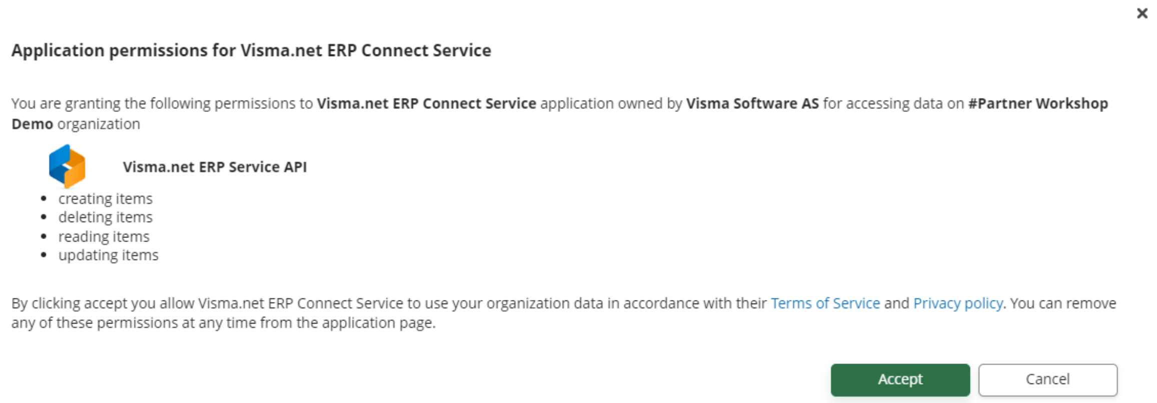
Application **Visma.net ERP Connect Service** requires the following permission for accessing data of your organization. You can grant or revoke these permissions at any time.

Visma.net ERP Service API

creating items
deleting items
reading items
updating items

6. To integrate with the application, activate each of the required permissions and press **Integrate**.

7. You will be presented with a confirmation screen.



Application permissions for Visma.net ERP Connect Service

You are granting the following permissions to **Visma.net ERP Connect Service** application owned by **Visma Software AS** for accessing data on **#Partner Workshop Demo** organization

Visma.net ERP Service API

- creating items
- deleting items
- reading items
- updating items

By clicking accept you allow Visma.net ERP Connect Service to use your organization data in accordance with their Terms of Service and Privacy policy. You can remove any of these permissions at any time from the application page.

To continue press **Accept**.

8. The application has now been granted the required access to your data, and can get access to the APIs. You can at any time change these permissions from the Visma App Store.
9. The application publisher might need your Tenant ID, which is the unique identifier of your company. This can be found by pressing the context-menu in the top-right corner.