01Financials

Developer documentation

Documentation	
Release	1.3.0
Author	michael@exthand.com
Date	2020-11-19

© 2020, EXTHAND bvba/sprl. All rights reserved.
No part of this document may be reproduced in any form by print, photo-print, microfilm or any other means or stored in a retrieval system without the prior written permission from EXTHAND bvba/sprl.
EXTHAND bvba/sprl shall not in any circumstances be liable, whether in contract or tort, for any damages or any consequential or indirect loss or damage, howsoever arising and of whatsoever nature, resulting from the use of this document or derived documents.
This document may be subject to change without notice. EXTHAND bvba/sprl shall not in any circumstances be liable for any damages or any consequential or indirect loss or damage, arising as a result of such a change.

EXTHAND bvba/sprl
Av. Arnaud FraiteurLaan 15-23 A29
B1050 BRUSSELS
BELGIUM
VAT BE0701961581.

Contents

01. INTRODUCTION & DEFINITIONS 4
Introduction 4
URLs 4
Demo 4
Production 5
Definitions 5
02. Overall process & relations 6
Overall process 7
Customer pre-creation 7
Step 1 7
Step 2 8
Step 3 8
Step 4 9
Step 5 9
Step 6 10
Step 7 10
Flow continuity 11
Data structure 11
Manager – installations – connections 12
One installation by Accountant 12
One installation by customer 13
Accounting case 14
“Accounting integrator” object diagram 14
Comments 14
Direct relation & Installation access case 15
Direct relation object diagram 15
03. How to start 17
Important hints! 17
Landing page 18

Integrator dashboard 19
Integrator – create Installation 20
Integrator– Request new Connection 21
PSU – accept access request 22
PSU – manage Accounts 23
PSU – Add account 23
PSU – Share account 25
04. Using the API 26
API – authentication 26
API – order of calls 28
API – transactions list structure 29
API – CODA/CAMT53 31

01.

INTRODUCTION & DEFINITIONS

Introduction

In the demo website, the accounts and their data added are fake accounts. So, no SCA will be done with the banks.

URLs

Demo

https://test.digiteal.eu
https://demo-portal.01financials.eu

https://demo-api.01financials.eu

Production

https://app.digiteal.eu
https://portal.01financials.eu
https://api.01financials.eu

Definitions

User	One user of the application
Company	The company of the User
Integrator	The Company directly client of 01Financials which can create installations and use the API. Ex: Accounting integrator
Manager	A tier company in relation with the integrator and which is in relation with the PSU (bank account holder). Ex: the accountant. The integrator can be the manager in case the manager is not required (direct relation between integrator and PSU)
PSU	A Company that is the bank account holder. Ex: the client of the Manager or Integrator
TPP	Third Party Provider. A company, certified by national bank authority, that is allowed to access bank account with the consent of the PSU
Installation	A concept/object that unifies the integrator and the manager to which the PSU will attach bank account. Note that the integrator can also be the manager of an installation
Access Request	An access request created sent to the PSU by the manager or by the integrator on behalf of the manager
Connection	A connection between Installation and the PSU's bank account
PsuConnection	Synonym of Connection
InstallationPsuConnection	Synonym of Connection
Transaction	A bank account transaction (=movement)
Balance	A typed bank account situation for a date
AIS	Account Information Service: service to retrieve transactions and balances of a bank account
PIS	Payment Initiation Service: service to initiate payments

02.

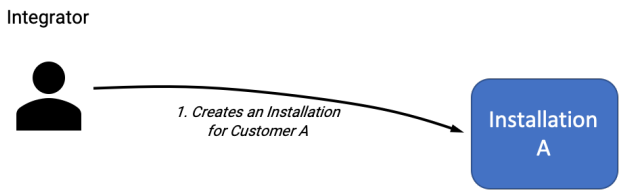
Overall process & relations

Overall process

Customer pre-creation

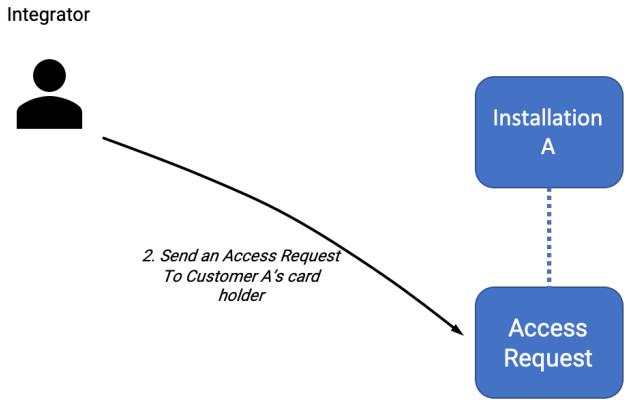
Through the API of Digiteal, you MUST pre-create customer prior to create an object for him. So, if you don't create an installation in your on name, the manager must exists in Digiteal customers' list and if you create a connection for a PSU, he must exists too. If 01Financials system doesn't found the email you give at creation in Digiteal's customers, the call will fail.

Step 1



The integrator creates an *Installation* to hold the accounts of his customer A.
All the functionalities available to the integrator (apart onboarding) can be done through the web I/F of 01Financials or programmatically through our REST API.

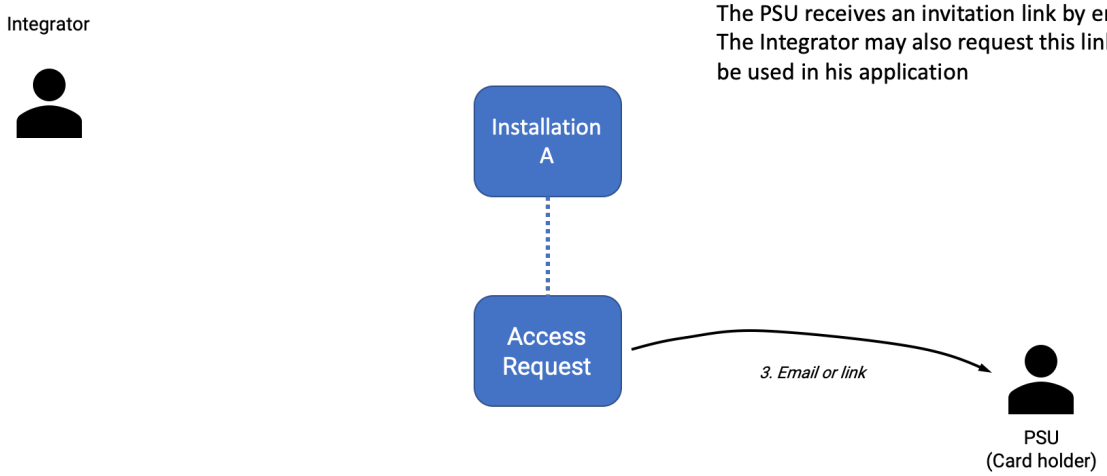
Step 2



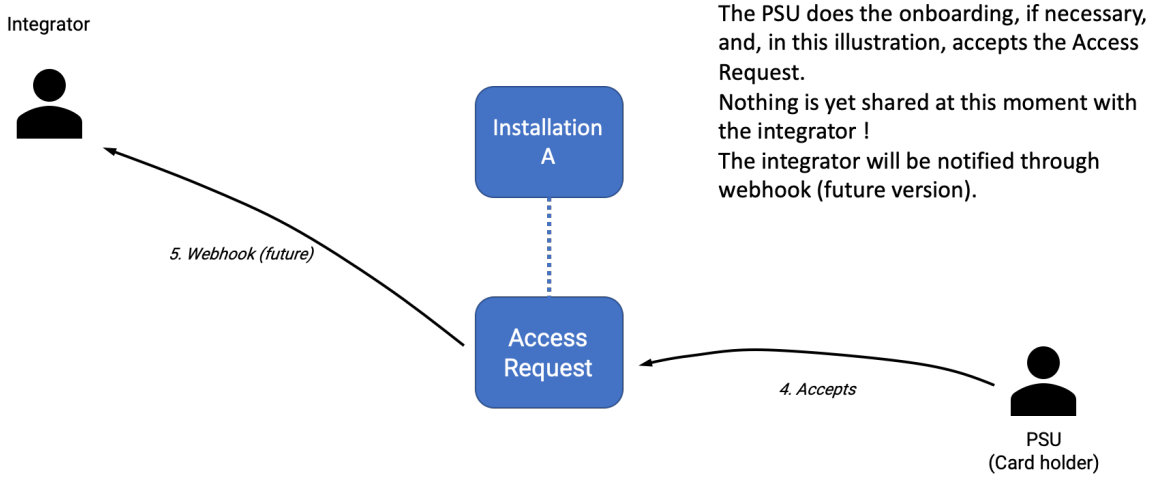
The integrator asks the card holder to allow access to his bank account(s) by creating an *Access Request*.

Step 3

The PSU receives an invitation link by email. The Integrator may also request this link to be used in his application

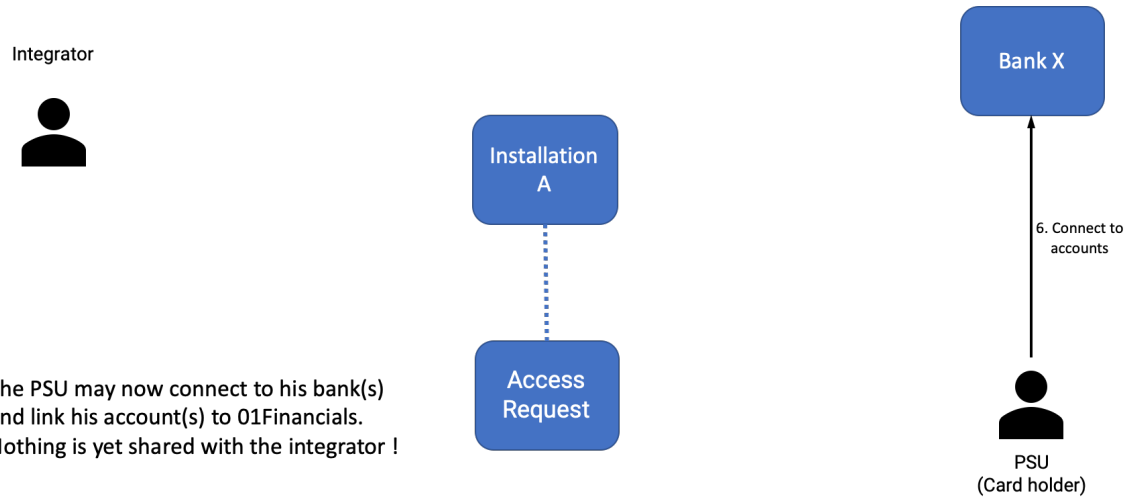


Step 4



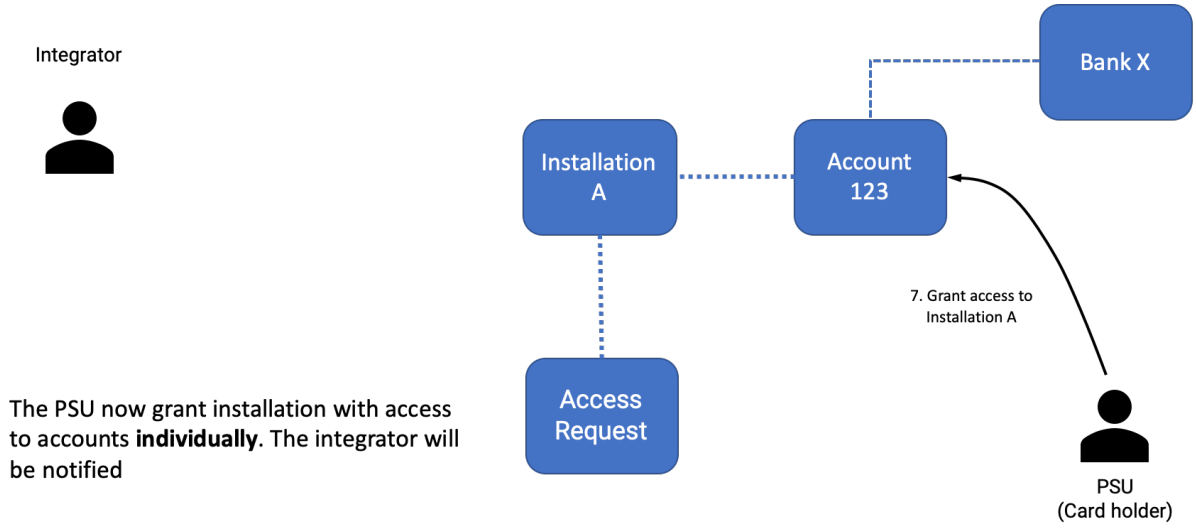
The PSU does the onboarding, if necessary, and, in this illustration, accepts the Access Request.
Nothing is yet shared at this moment with the integrator !
The integrator will be notified through webhook (future version).

Step 5



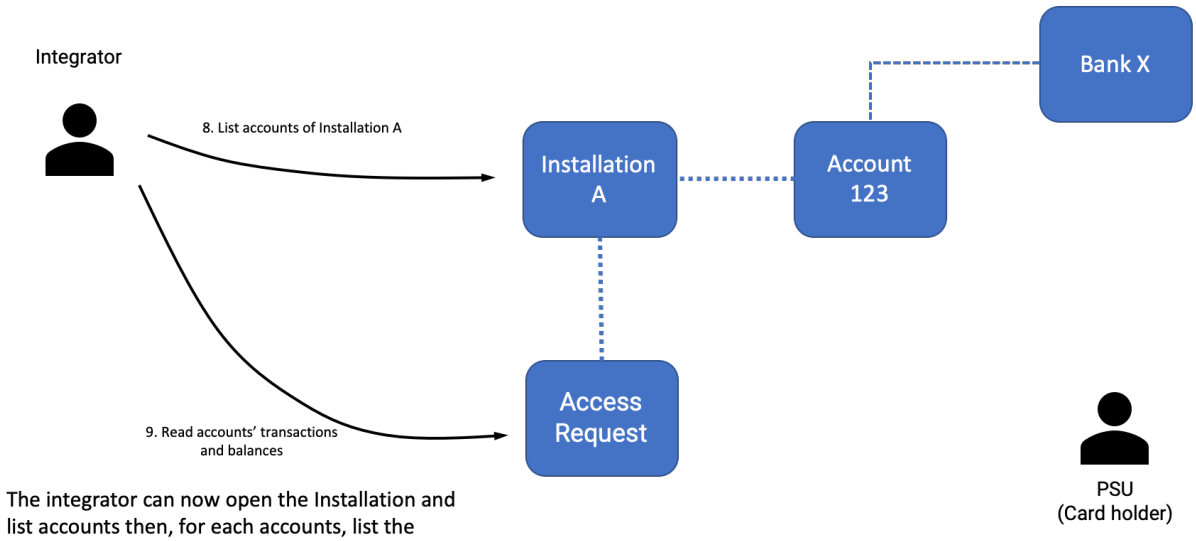
The PSU may now connect to his bank(s) and link his account(s) to 01Financials.
Nothing is yet shared with the integrator !

Step 6



The PSU now grant installation with access to accounts **individually**. The integrator will be notified

Step 7



The integrator can now open the Installation and list accounts then, for each accounts, list the transactions and/or balances.
Various format of transactions are available.
Details of accounts, transactions and balances are only accessible through the API!

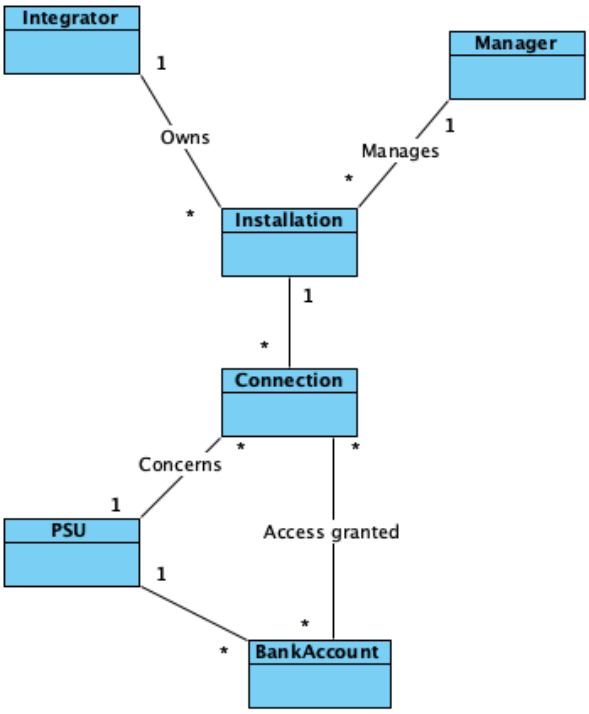
Flow continuity

When you call the API of Digiteal to pre-register a customer, he receives an email at the email address you gave with a unique link to onboard. At the end of onboarding, if you have given a returnURL in the pre-creation call, he will be redirected to your site just. The email sent to the customer is mandatory.
In the API call to create an installation, POST /api/Integrator/Installations, you may

- Set a flag (skipNotification) to prevent an email to be sent to the customer
- Give an URL to lead the customer back to your site (button displayed in the web interface)

You will also get a link attribute containing the URL of the created object you can use to redirect your customer to.

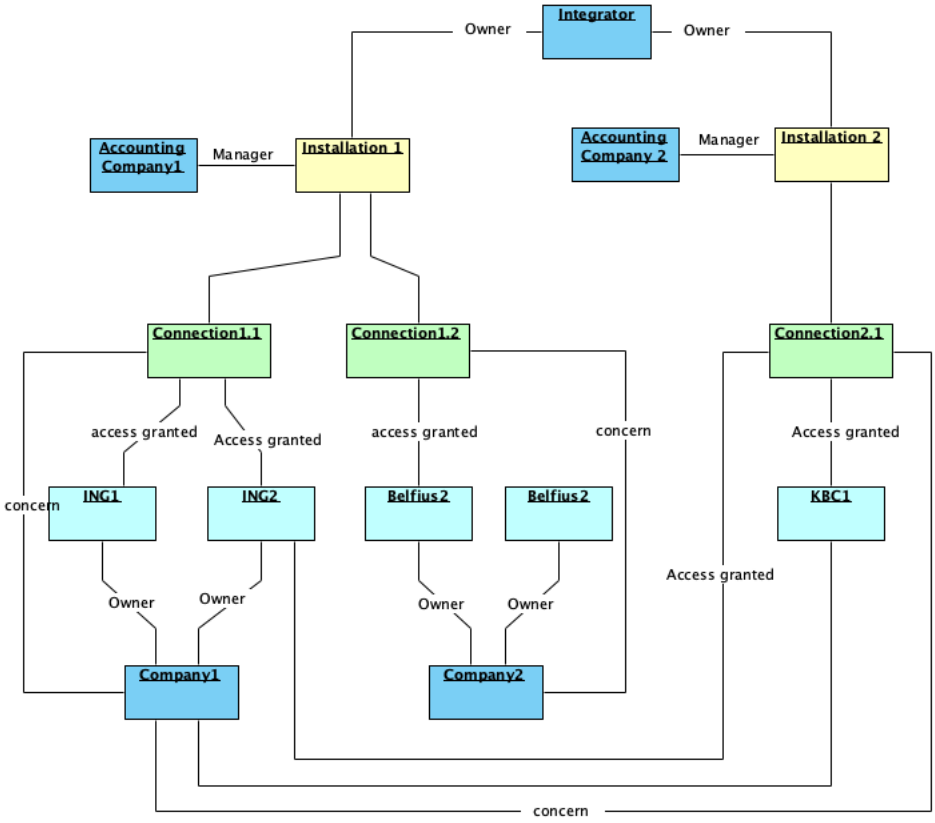
Data structure



Manager – installations – connections

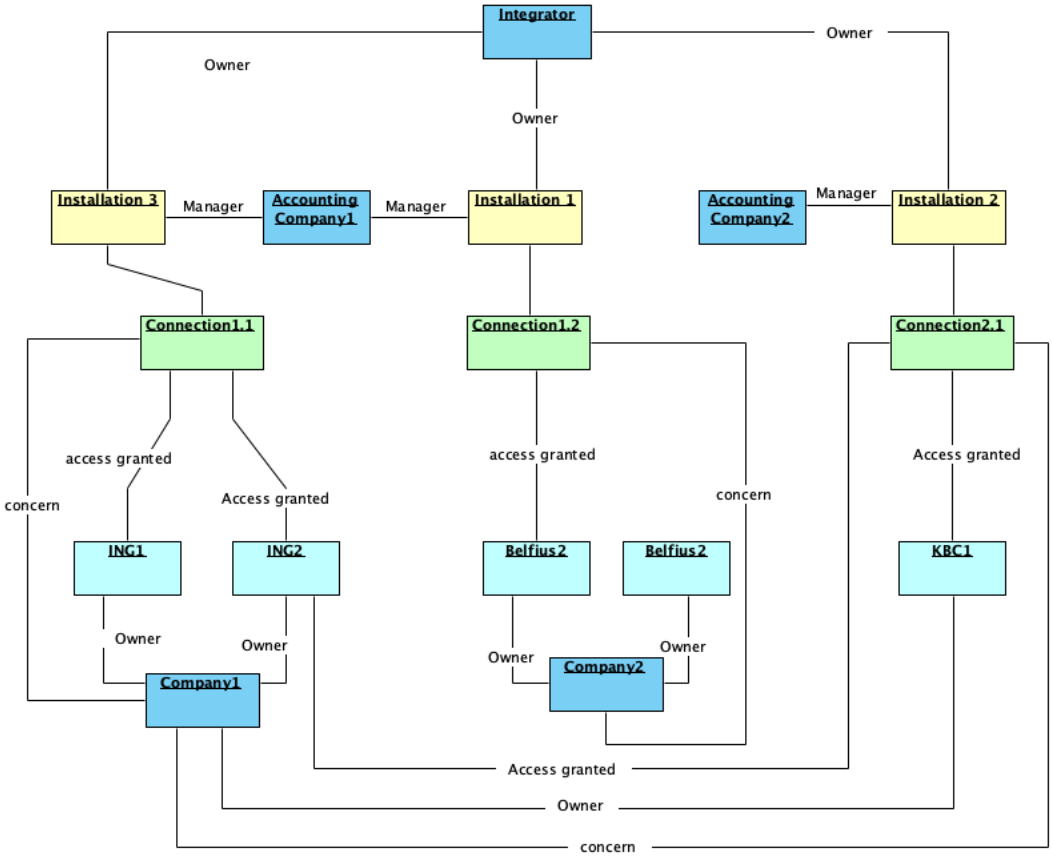
API Access rights are:
Connected as integrator, list all “Owner” Installations → Connections → “granted” Accounts → Transactions and Balances.
Connected as Manager, list all “Manager” Installations → Connections → “granted” Accounts → Transactions and Balances.
Connected as Installation, list all connected Connection → “granted” Accounts → Transactions and Balances.
The connection entity is visible within the API but not on the web I/F since it’s a technical link entity (*-* relationship). The account holder(=PSU) sees directly the list of Installation he is connected with.
There are no means to block a connection from the side of manager or integrator as for the installations.
One account can be linked by the PSU to multiple installation (like ING2 in diagram).
Account may have no relation with installation (like Belfius2 in diagram).

One installation by Accountant



In this case, the integrator has only one installation by manager (Accounting Company1 & 2). The different final client companies are linked to the same installation through their connections.

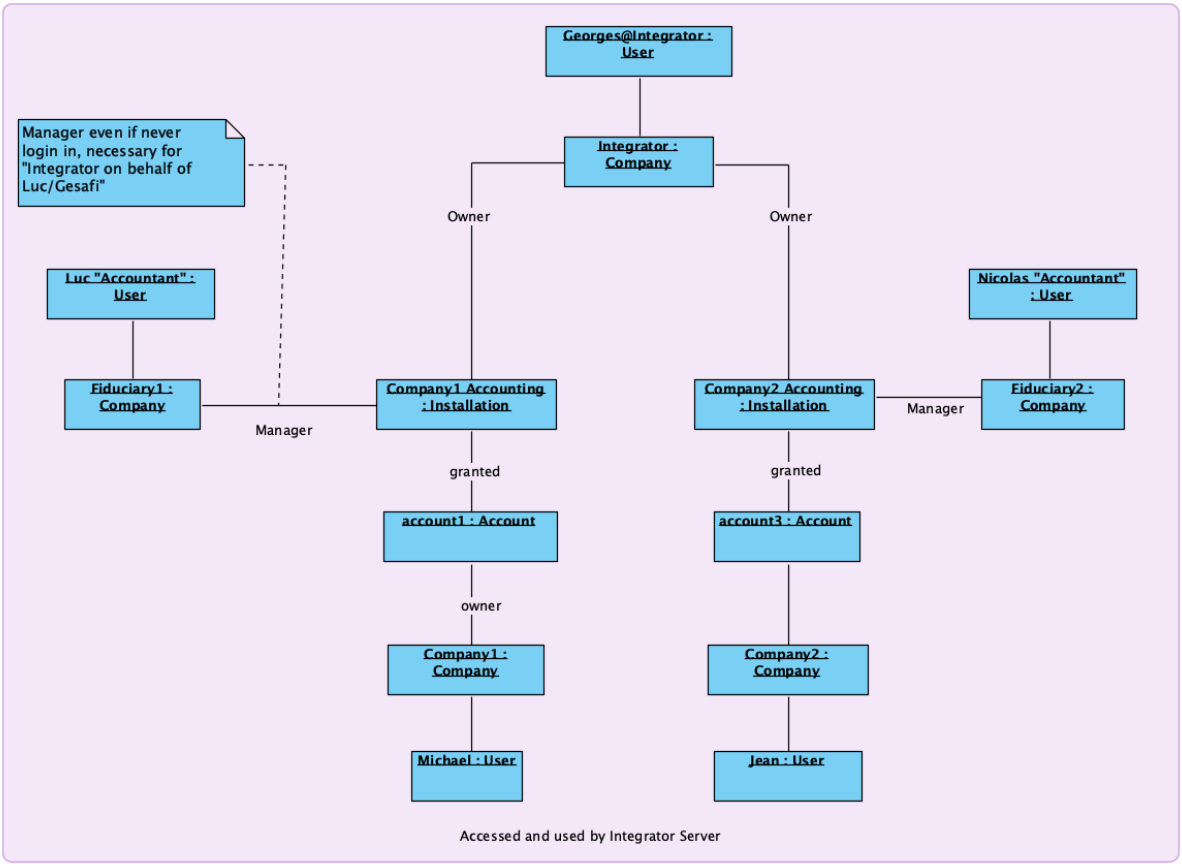
One installation by customer



In this case, the integrator has one installation by final client companies (Company1 & 2).

Accounting case

“Accounting integrator” object diagram



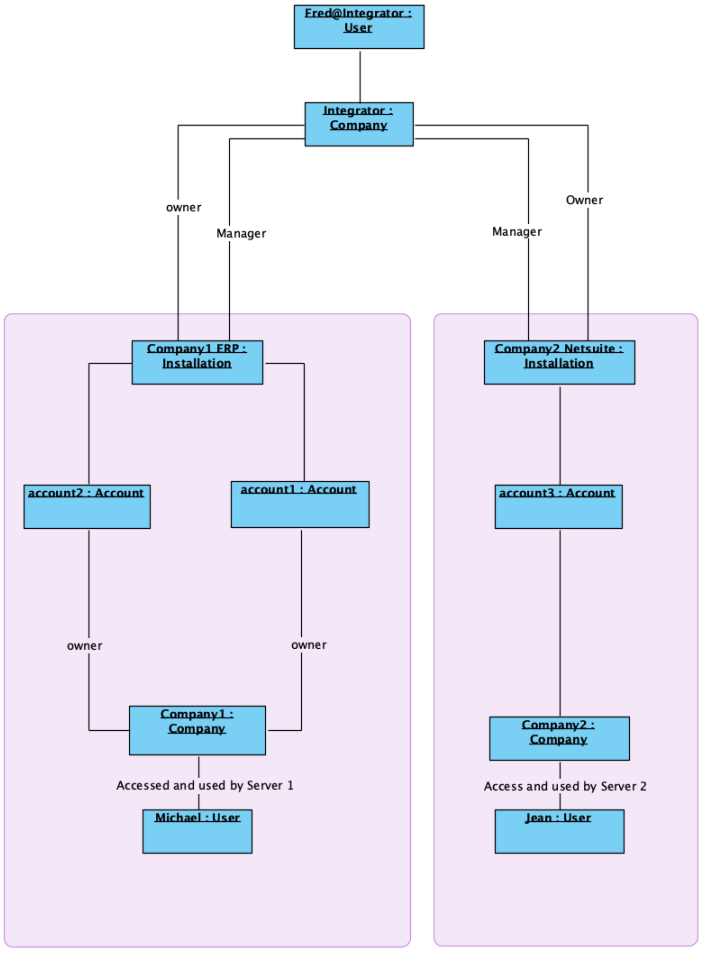
Comments

The purpose of this object diagram is to help you understand the concepts of 01Financials and their relations.
In an accounting business example, the accounting app integrator will be using the manager as a trust person in the relation with the PSU. When access request and installation will show that the integrator act on behalf of the accountant. Of course, in production, consent will be asked to the accountant (intermediary) to allow the Accounting app (integrator) to act on behalf of him!
An installation may be connected to multiple accounts of multiple PSUs.
The integrator server will access, through the API, all the installations and all the accounts beneath (rose rectangle).

If your business doesn't have intermediary, you don't have to set the manager of Installations to someone else, you'll be the owner and the manager of them.

Direct relation & Installation access case

Direct relation object diagram



In this case, the 01Financials integrator is directly in relation with the PSU, no intermediaries. The integrator is the Onwer and the Manager of the Installation.
Integrator has one instance of its app by customer. Each instance should have access to the accounts linked to one installation (represented by rose rectangle).

For this, integrator app instance will have to login on the OAuth server (machine to machine) with the installation credentials first and then authenticate on the API server to access the installation.
By this authentication,

1. Installations are completely isolated from each other and also
2. accounts are also well isolated from being accessed by an unauthorized integrator and
3. the credentials of integrator is not widespread over all app instance,
4. on one instance,
    a. there is only the Installation information related to integrator's customer
    b. there are only the OAuth credentials of the installation

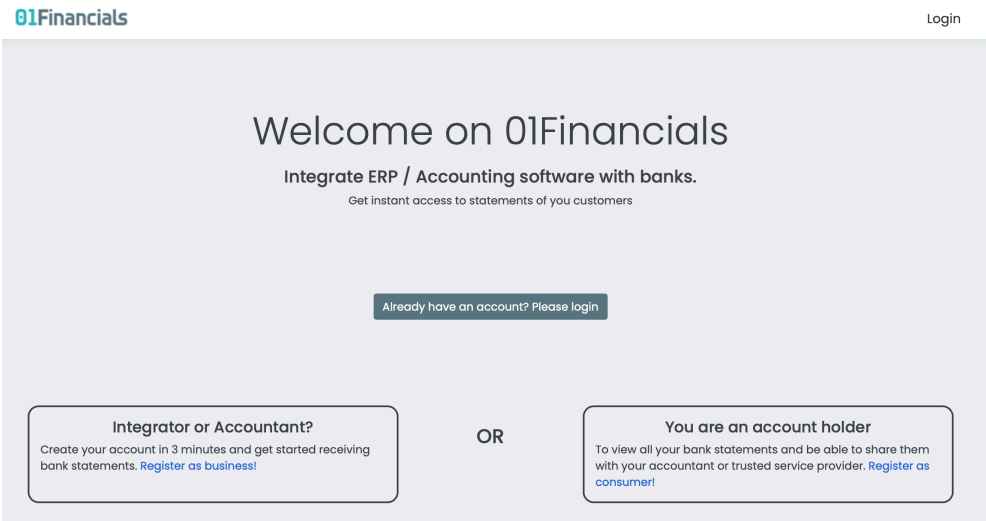
03.

How to start

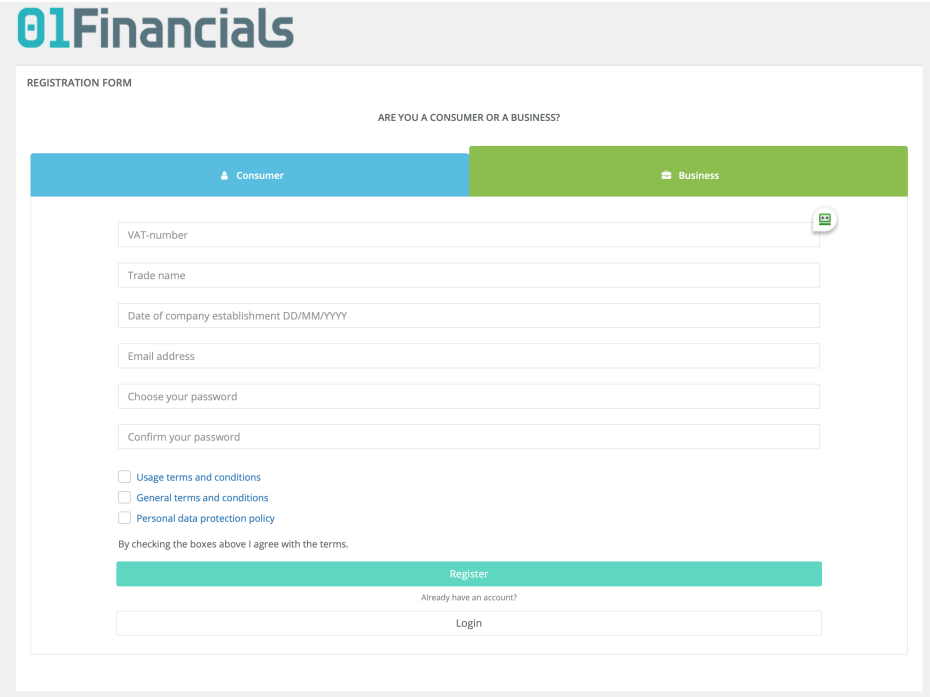
Important hints!

Email address is unique for the User.
You will then need one email address for the Integrator and one for the PSU. You'll need one more if you want to use the Manager.
If you use or have Gmail account, you may use only one address and add "+xxxx". I.e. if my Gmail base address is michael@exthand.com, I could use michael+integrator@exthand.com for the Integrator account, Michael+manager@exthand.com for the Manager account and Michael+psu@exthand.com for the PSU account.
You have to onboard in Digiteal (https://test.digiteal.eu for test and https://app.digiteal.eu for production).
The demo site uses Digiteal's test plateform.

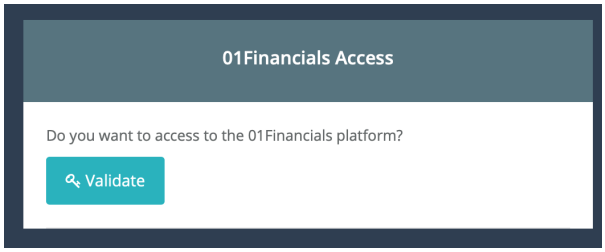
Landing page



Click on Register as business buttons if you don't have an integrator account or on login if you already have one. You will be redirected to Digiteal's onboarding page

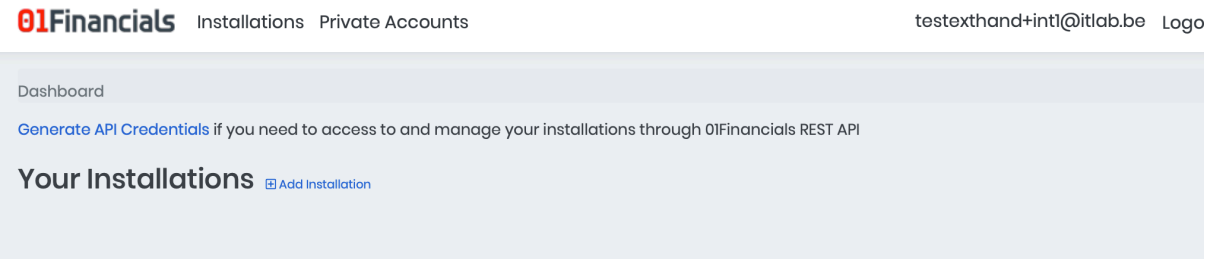


Do the login and accept the access



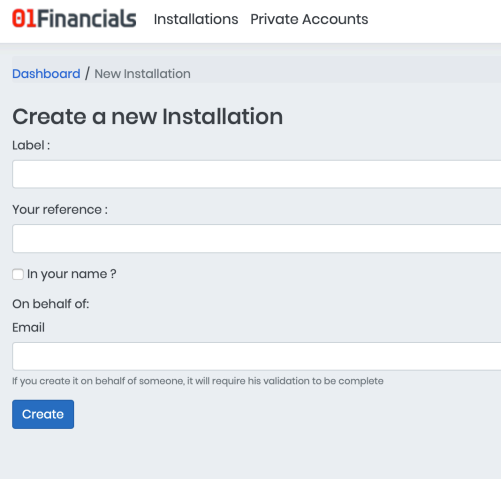
Integrator dashboard

You are now on the Integrator dashboard.

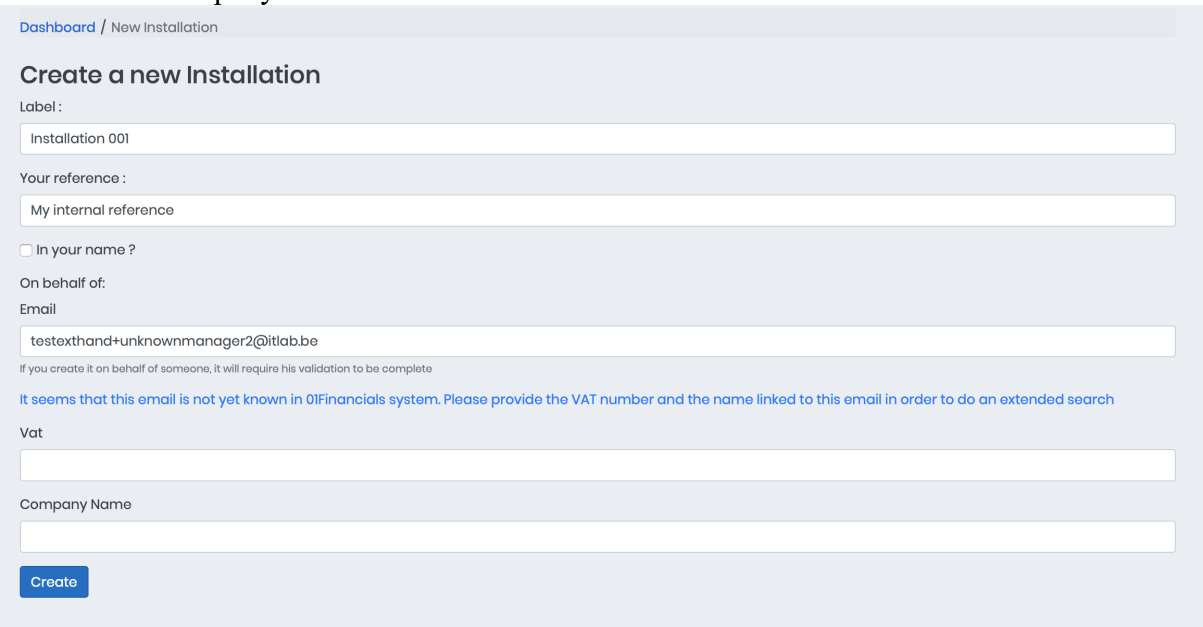


Add installation leads you to the add installation screen
Generate API Credentials leads you to the page to generate the credentials required to access 01Financials API and access all the Installations > connections > accounts > statements and balances.

Integrator – create Installation

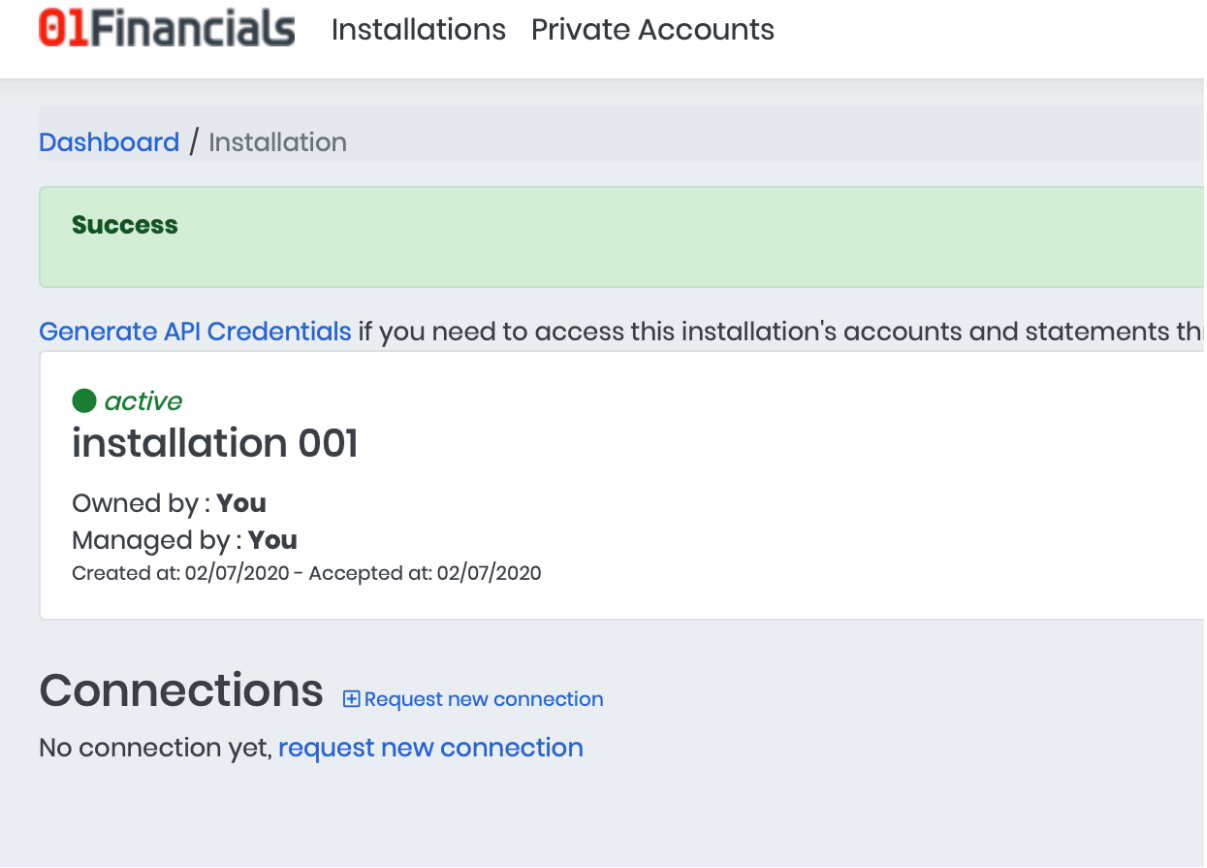


The label is visible by everyone.
Your reference is only visible by the owner (=creator) of the installation.
In your own name: means that you will be the only user that will use this Installation and you won't invite the PSU on behalf of someone else (like the accountant).
If you check it, you won't be able to enter an email.
If you want to add someone as the manager of the installation, you have to enter the email of the manager.
In case the system doesn't know this email, he will ask you to enter the VAT number and the name of the company.



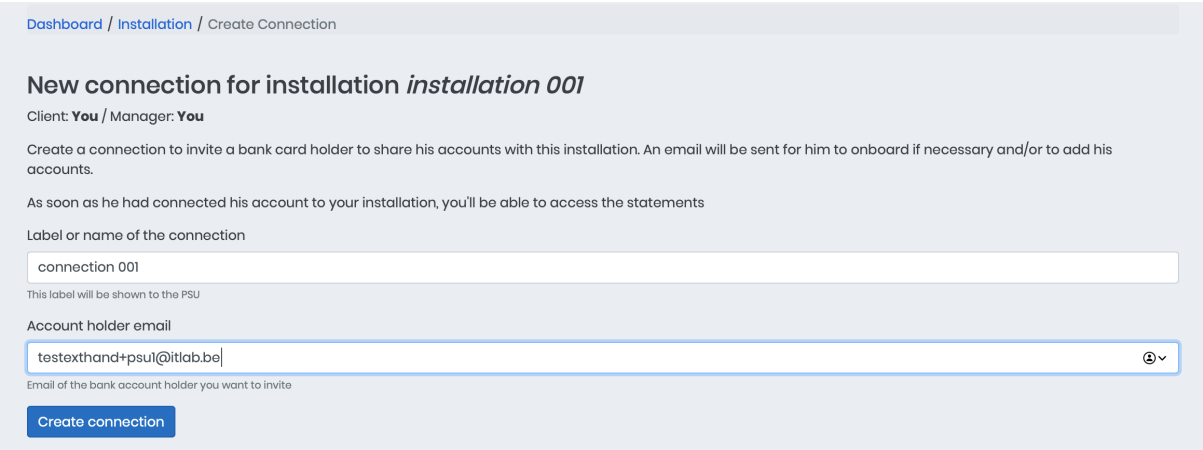
The target manager will receive an email to onboard in Digiteal and an email to validate the installation after he has onboarded in Digiteal and logged in 01Financials.

After creation, you see the detail of the Installation

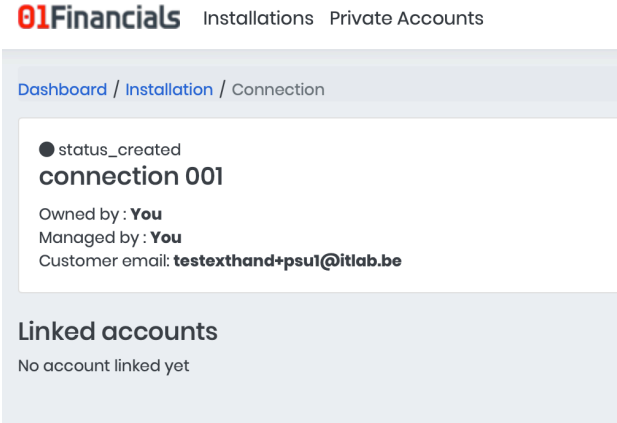


Integrator– Request new Connection

Entering in the detail of an installation, you can see the list of existing connections with PSU. You can create a new connection request that will send an email to onboard in Digiteal if the PSU is not known and an email to invite the PSU to accept the Connection and add accounts.



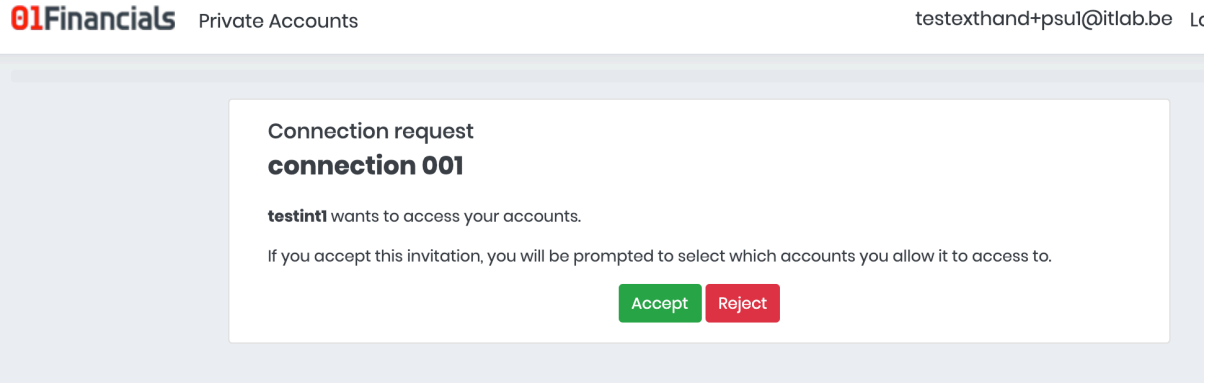
After creation, you see the detail of the Connection.



This one has not yet been accepted by the PSU

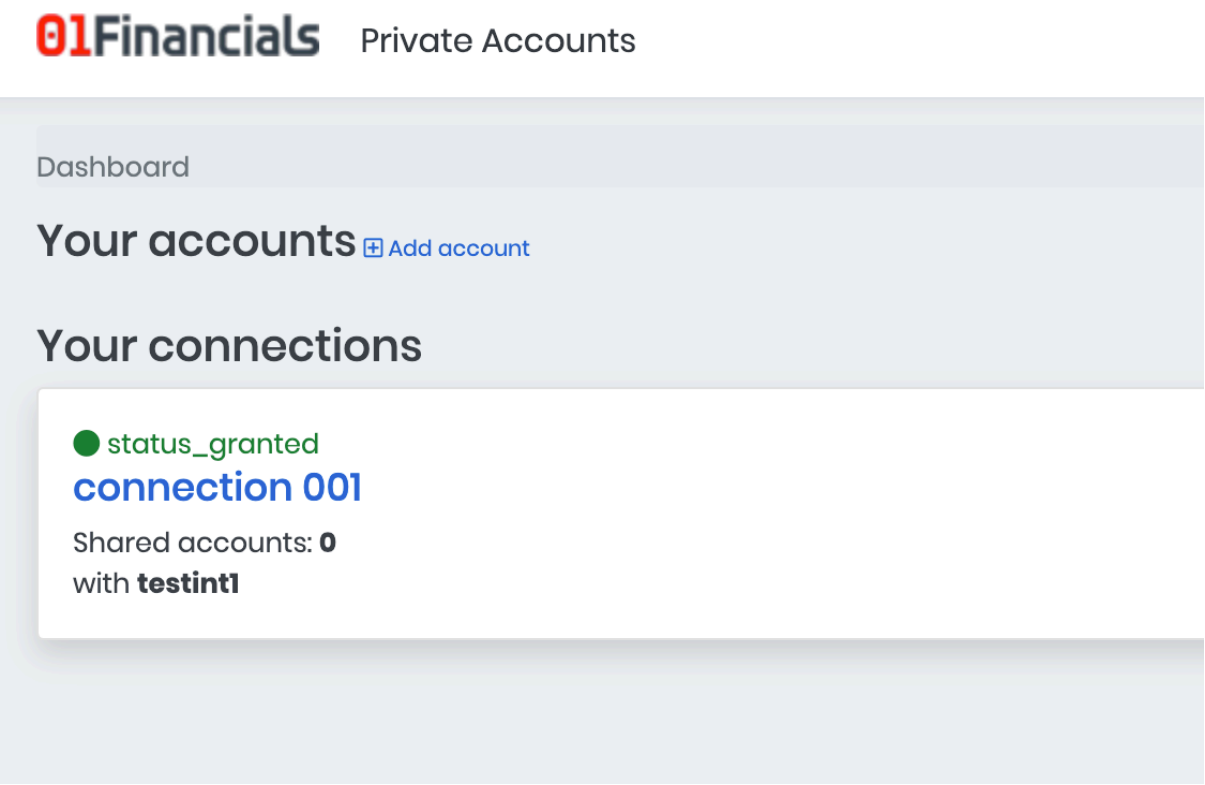
PSU – accept access request

After the PSU has onboarded and logged in 01Financials:



PSU – manage Accounts

PSU dashboard:

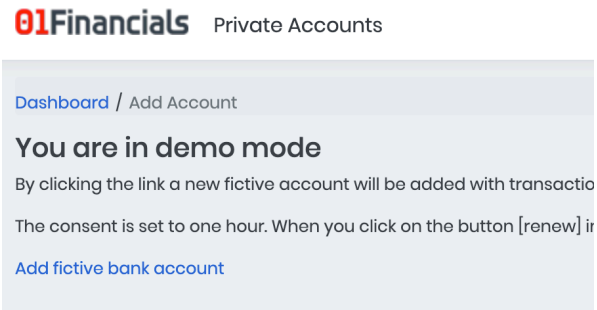


There is one connection that shares 0 account.
No accounts have been created in 01Financials for this user.

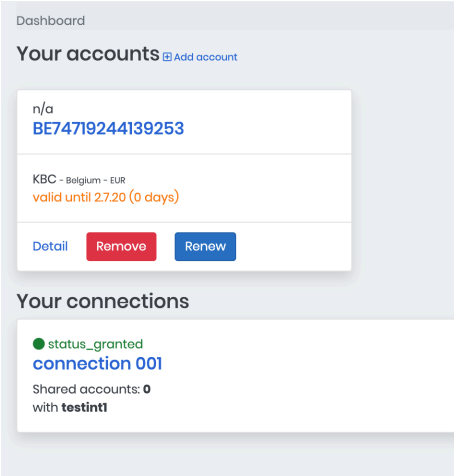
PSU – Add account

On the dashboard, the button [Add account] allows the PSU to link its bank accounts to 01Financials. After selecting the country and the bank (or financial institution), he will be redirected to that select bank to do the SCA and confirm access. He is then redirected back to 01Financials.
This operation generates a consent given by the PSU to 01Financials to do AIS and PIS.

In this demo mode, it adds a fake bank account with fake transactions and balances.



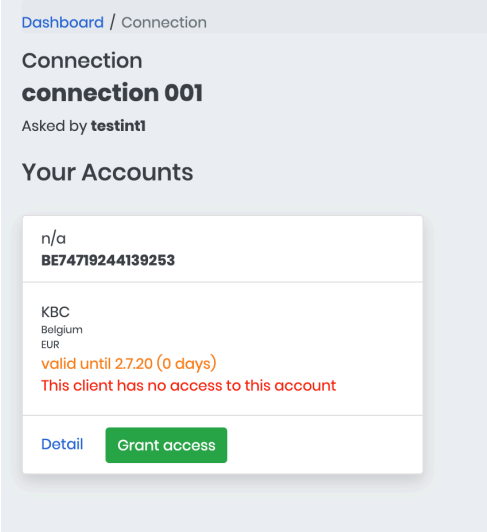
Fictive account has been added



“N/a” is the placeholder for the description of the bank account.
“BE74719244139253” is the fake IBAN
It is for the bank KBC in Belgium. It’s a euro account.
“valid until” is consent validity, the date at which the authorization given by the PSU to 01Financials will end. in real PSD2 it’s 90 days. Here, in demo mode it is set to one hour. When this consent has expired, 01Financials calls to the bank will fail. The consent must then be renewed.
In this demo, the renew button just sets the validity date to now plus one hour and adds fake transactions and balances. In production, the renew will go to the bank to renew the consent. You can also remove the account.

PSU – Share account

It shows information about the connection, in this case it has been asked the manager for the client and the list of linked accounts of the PSU. The PSU may grant access to one or multiple accounts.



The Integrator and the Manager can now see the allowed accounts details through the API and through the web.

04.

Using the API

To use the API, you must be an Integrator.
The default page of the API server is a Swagger help.
{{base}} is the server name and base path of the endpoints (see URLs).

API – authentication

01Financials' API must identify who is making the call by inspecting an OAuth token that you will supply to HTTP request. To obtain this token, you need to:

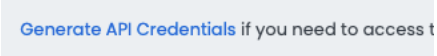
1. Get an id/secret pair for your account by going to 01Financials website
2. Use this pair to get a token from the Digiteal's OAuth service.
3. Pass the token to 01Financials calls01Financials will use this token to validate you user account and let you access your resources.

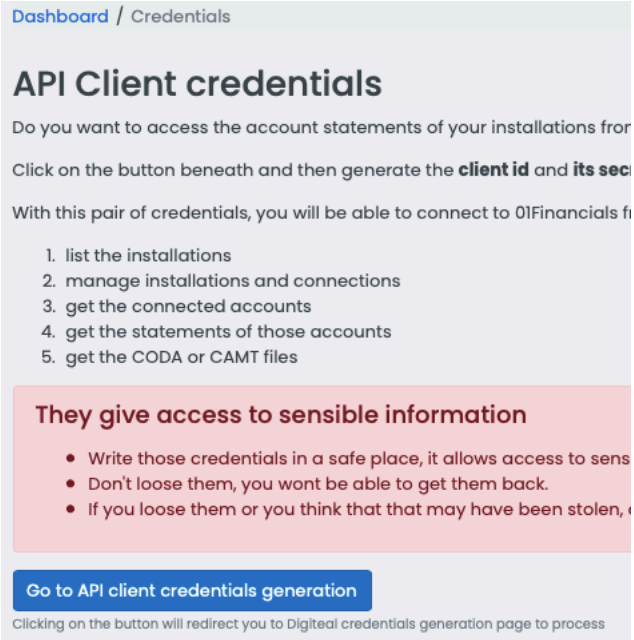
There are two kind of access, integrator access or installation access.

- **Integrator access** uses a token generated from your account id and secret and lets you access all your installations → connections → bank accounts → transactions and balances.
- **Installation access** uses a token generated from the id and secret of one precise installation and lets you access this installation connections → bank accounts → transactions and balances

To generate a token and access 01Financials API follow steps bellow.

1. In 01Financials portal integrator dashboard or installation detail page, request API credentials,





2. You will be redirected to Digiteal page,

3. Copy the Id,
4. Generate new secret (if you already generated secret for that specific connection it will be replaced with the new one),
5. Copy new secret and store them in secure location. It will never be shown again, but you can always generate new secret for any connection which will make the existing invalid,
6. Using generated secret, you can now generate an API access token by calling this token endpoint (note: test server domain of Digiteal):

```
curl --location --request POST
'https://test.digiteal.eu/api/v1/01financials/tokens'
--header 'Content-Type: application/json'
--data-raw '{
    "client_secret": "[secret]",
    "client_id": "[id]",
    "grant_type": "client_credentials"
}'
```

7. In response you will receive Json token data in following format:

```
{
    "access_token": "string",
    "refresh_token": "string",
    "token_type": "Bearer",
    "expires_in": int,
    "scope": "string"
}
```

8. Now you can access available 01Financials API endpoint by using received token in Authorization header:

```
Authorization: Bearer [access_token]
```

Note that the access_token has a limited validity. After expiration, you must get a new one using the refresh_token by calling the same end-point with the body data :

```
{
    "client_id": "[same id as first call]",
    "refresh_token": "[refresh token of first response]",
    "grant_type": "refresh_token"
}
```

You will receive the same response format as the initial tokens call with new access token. The refresh token stays the same.

The expires_in field represent the number of seconds to be added to 1970-01-01 00:00:00 UTC

API – order of calls

Endpoints to get the transactions as CODA or other format are still in beta. The Json format is available and stable.
The calls for those specific formats will be the same as for Json but with the format name added to the endpoint.
In the case of integrator access, the order of other endpoints could be:

- OAuth login as explained with integrator id and secret

- List installations: GET /api/Integrator/Installations
- List connections: GET /api/Integrator/Installations/{installationId}/connections
- List accounts of a connection: GET /api/Integrator/Installations/{installationId}/connections/{connectionId}/accounts
- List Transactions of an account: GET /api/Integrator/Installations/{installationId}/connections/{connectionId}/accounts/{accountId}/transactions
- List Balances of an account: GET /api/Integrator/Installations/{installationId}/connections/{connectionId}/accounts/{accountId}/balances

In the case of installation access, the previous sequence could be done as Client but only from their own server.
For their customer app instance:

- OAuth login as explained with installation id and secret
- List connections (no need for installation id): GET /api/Installation/connections
- List accounts of a connection: GET /api/Installation/connections/{connectionId}/accounts
- List Transactions of an account: GET /api/Installation/connections/{connectionId}/accounts/{accountId}/transactions
- List Balances of an account: GET /api/Installation/connections/{connectionId}/accounts/{accountId}/balances

There are also other endpoints available when connected as integrator to:

- Create an installation
- Activate installation
- (un)suspend an installation
- Delete an installation
- Create an access request

API – transactions list structure

Here is the structure of the list of transactions returned by the API.


```
{
    "transactions": [
        {
            "id": "70000000-0000-0000-0000-000000000002",
            "debtorIban": "BE68260028272031",
            "amount": 101.00,
            "counterpartReference": "x iban",
            "counterpartName": "x",
            "currency": "EUR",
            "description": "test transaction",
            "executionDate": "2020-05-19T09:33:27.3439",
```

```
            "valueDate": "2020-05-19T09:33:27.343902",
            "accountId": "40000000-0000-0000-0000-000000000001",
            "requestedAt": "2020-05-19T09:33:27.343902"

        },
        {
            "id": "70000000-0000-0000-0000-000000000001",
            "debtorIban": "BE68260028272031",
            "amount": 100.00,
            "counterpartReference": "x iban",
            "counterpartName": "x",
            "currency": "EUR",
            "description": "test transaction",
            "executionDate": "2020-05-19T09:33:27.332606",
            "valueDate": "2020-05-19T09:33:27.332706",
            "accountId": "40000000-0000-0000-0000-000000000001",
            "requestedAt": "2020-05-19T09:33:27.343902"

        }
    ],
    "page": 0,
    "size": 10,
    "dateFrom": "0001-01-01T00:00:00",
    "total": 2
}
```


Transaction.id: the id of that transaction in 01Financials system
Transaction.debtorIban: the account this transaction belongs to, the account of the PSU you are connected with
Transaction.amount: the signed amount of the transaction
Transaction.counterpartReference: the IBAN of the counterpart account
Transaction.counterpartName: the name of the counterpart
Transaction.currency: the currency of the transaction
Transaction.description: the remittance of the transaction
Transaction.executionDate: the date when the transaction has been requested
Transaction.valueDate: the date when the transaction has been done
Transaction.accountId: the 01Financials id of the account the transaction is related to
Transaction.requestAt: the date at which 01Financials has received the transaction from the bank
Page: the page number (zero based) of the current list
Size: the page size
DateFrom: the filtering date. All the transactions have been fetched from bank after that date
Total: total number of page for that query

API – CODA/CAMT53

Getting the CODA/CAMT53 works the same as getting the transactions as JSON.
Identified as integrator, you can request the CODA/CAMT53 through the REST API.

- You must be the owner or manager of the installation,
- The connection must be accepted by the PSU,
- The connection must have been granted access to the account by the PSU

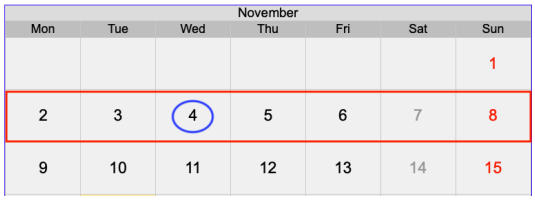
End-points:
`/api/Integrator/Installations/{installationId}/connections/{connectionId}/accounts/{accountId}/coda`
and
`/api/Integrator/Installations/{installationId}/connections/{connectionId}/accounts/{accountId}/camt53`
The path parameters are:

- Installation id (UUID string)
- Connection id (UUID string)
- Account id (UUID string)

The query string parameters are:

- PeriodType (integer): the period you want to take as a whole "CODA/CAMT53 file":
  - 0 = day
  - 1 = week
  - 2 = month
  - 3 = quarter
  - 4 = year
- periodFrom (string date, format yyyy-MM-dd): one date inside this period. Must be less or equal to today

The week starts on Monday and will not overlap the year.
The system will compute the lower and upper borders around this date corresponding to the requested period type. The date doesn't have to be on the period border.
Example:



If you give the Wednesday November 4, 2020 as periodFrom and week (periodtype=1) as periodType, the system will take, inclusively, the transaction for which the value date is between Monday November 2, 2020 and Sunday November 8, 2020.

The system doesn't memorize any information of which CODA/CAMT53 you already have requested.
You will get a JSON response containing the parameters you gave,

- transactionCount: the count of transactions fetched,
- sequenceNumber: the number identifying the period in the year
    - Day period type: the day number in the year
    - week period type: the week number in the year
    - Quarter period type: the quarter number (1..4)
    - Year period type: 1,
- requestedPeriodFrom : the date you provided to the end-point,
- periodType : the period type you provided to the end-point,
- periodTypeText : text version of the period type,
- dateFrom : the lower border of the period,
- dateTo : the upper border of the period,
- coda: an array of strings, each line contains a line of the CODA file OR
- camt: a string containing the whole CAMT53 XML
- html: the human readable version of the CODA/CAMT53 in HTML
- resultTruncated: true if the list has been truncated to the maximum number of records the server accept to return. At the time of writing it is 200

```
{
    "transactionCount": 20,
    "sequenceNumber": 4,
    "requestedPeriodFrom": "2020-10-19T00:00:00",
    "periodType": 3,
    "periodTypeText": "Quarter",
    "dateFrom": "2020-10-01T00:00:00",
    "dateTo": "2020-12-31T23:59:59.999",
    "coda": […], // OR
    "camt": "…",
    "html": "…",
    "resultTruncated": false
}
```