Belvo’s Challenge!

**Answering developers questions:**

Hi Carol! My name is John, I am a developer interested in Belvo’s services.

I have some doubts about Belvo’s platform and functionalities, could you help me please?

Of course John! How can I help you today?

1. Could you explain to me how do I handle the 2FA for link creation at Belvo? What if the token expired?

Excellent question Jhon! The general flow for MFA (2FA) works like this:

First you need to make a **POST** request to the institution to retrieve some data or to create a link. After that Belvo sends a 428 Token Required response, detailing which MFA method is needed to complete the request.

You prompt the user to input the required authentication token.You make a **PATCH** request with the link ID, session ID, and user-provided authentication token to the Resume endpoint. Belvo sends a 201 Success message. If the token expires don’t worry you can always make a call to refresh it and get a new one.

1. That was very clarifying, what’s the difference between a link created with access\_mode = single vs access\_mode = recurrent?

The single access mode can be used if you as a developer want to request through a call the data from a financial institute or an app. We recommend using recurrent access mode, it’s setup by default as in that way Belvo calls financial institute data every 6 hours, 12 hours or 24 hours. If you don’t change the configuration, it will be refreshed every 7 days. That way you don’t need to use the **POST** command to receive the information update, you only need to use **GET** and simplify the process.

1. So I guess I will prefer recurrent mode from now on. How do I make sure my end application can open Belvo’s widget?

You need to implement our script (src="<https://cdn.belvo.io/belvo-widget-1-stable.js>") with the parameter async and add “div id = Belvo” anywhere inside your application’s <body> tag so you can run Belvo’s widget.

1. Interesting, and how could I access a links transactions from 5 months ago using a GET request?

Well John, when you create a recurrent link, Belvo automatically retrieves key information about the **LinkId**. Once we have the information, we'll send you a webhook event indicating that you can make a **GET** request for that information. Also if you have a transaction ID you can create a **GET** to request that as well.

1. I have one last problem, I am facing an issue so I sended you a logs file. In some cases my team can’t check financial data from some final users, why is this happening?

Checking your logs file I noticed you are receiving the Error 400 with the message “login error” or “missing credentials”. There’s some reasons this error message can appear:

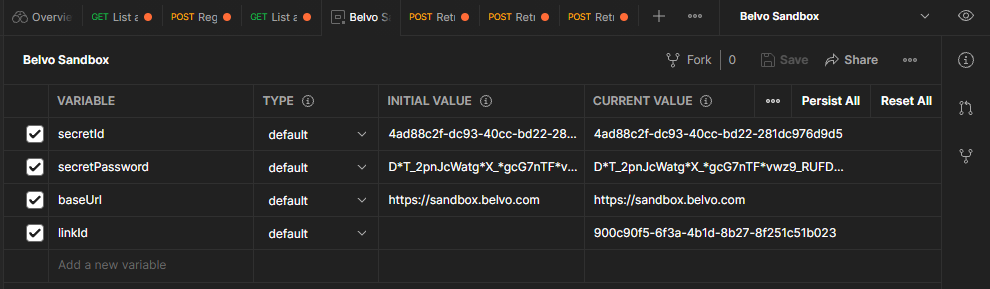
* The credentials that your user provides are incorrect or missing.
* The MFA token your user provides is not supported by Belvo.
* There is an issue with the institution that prevents login.
* The user’s account is either locked or the user does not have permission to access the internet banking.

To solve this, we can apply some troubleshooting such as:

* Asking your user to provide their correct credentials or MFA token (you can use Belvo widgets for that as well).
* Ask your user to confirm with their bank their account is active and it is not blocked.
* If there is an issue with the institution, try logging in at a later time.

**Creating and preparing your Postman & Belvo’s Dashboard:**

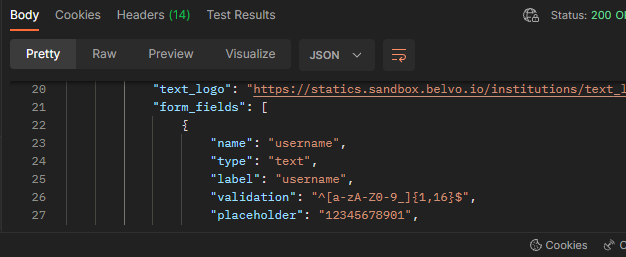
1. You will enter Belvo’s dashboard and sign up at ‘https://dashboard.belvo.com/sandbox/’
2. Once you are in the environment you have 3 package options: Sandbox (for tests and unlimited links) Developers (25 links and real data) and Production (unlimited links and real data).For tests it’s better to use Sandbox.
3. Next you need to generate your **secretPassword** and **secretId** at *“Open Finance”* then access *“API Keys”*. Click the button at the top right corner *“Generate API Keys”.* Now you have your credentials!
4. It’s time to go ahead and enter Postman to set up Belvo’s environment. Going to Belvo’s Postman workspace you can press *“Fork* to fork the API collection and add it to your workspace. Also, to always be notified about new updates, activate the *“Watch”* button on Belvo’s API. Then use *“Merge changes”* to pull the changes into your fork.
5. Inside the Postman you should create a new environment, clicking at the top right corner and selecting option *“ No Environment” > “ New Environment” .* You can name it **BelvoSandbox,** for example, and after naming you should add variables. This is where you’re going to need your credentials (**secretId** and **secretPassword**) that you previously generated and **baseUrl** for the APIs calls.
6. Please when inserting the **secretId**, **secretPassword** and **baseUrl** as variables type them exactly as I just did above. After that, complete the *initial value* section with the IDs that were generated at the dashboard. At the initial value section for **baseUrl** just insert “<https://sandbox.belvo.com>” and then click *“Save”.*

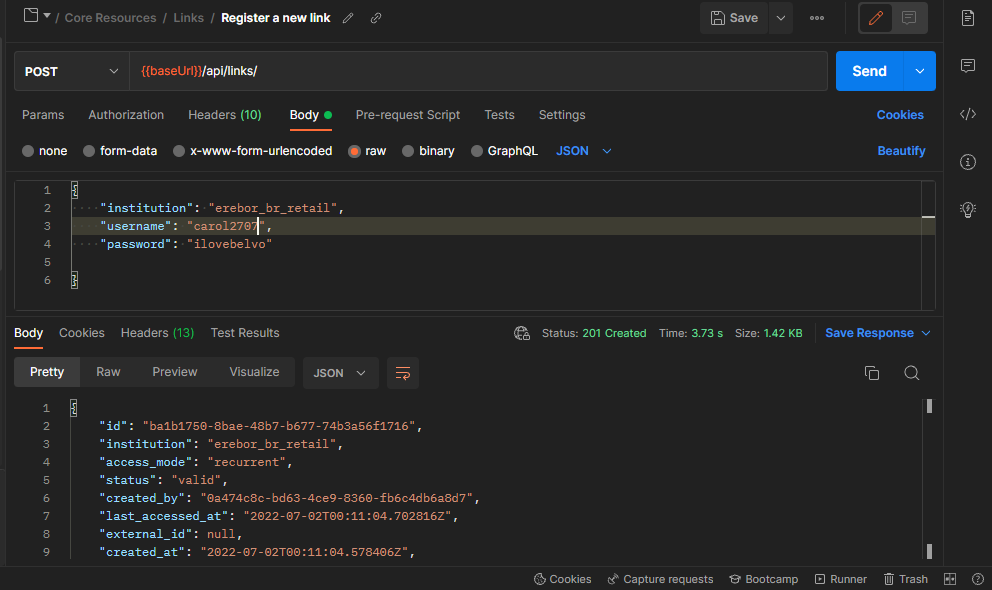


1. We can test if it’s working by going into the *“Institutions”* folder, and then selecting *“* ***Get*** *list all Institutions*” and click *“Send”* The result should be a list of all the institutions that are registered and the **status: 200 OK**

**Creating a link & Testing our Main Endpoints using Postman:**

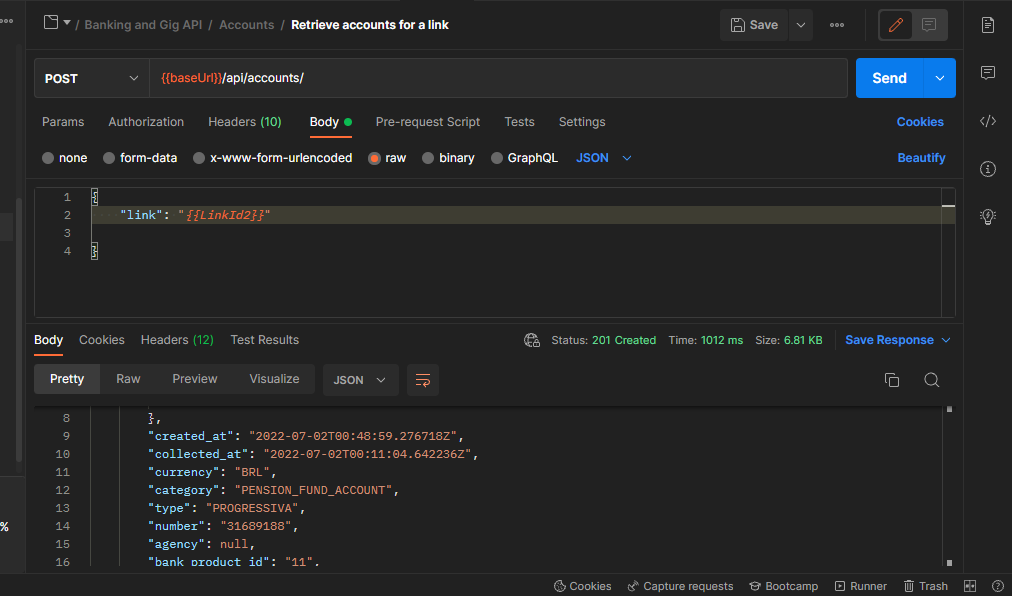
1. Now that we are up and running let’s start testing the calls on Belvo’s API. First, in the *“Institutions”* folder, at the *“Params”* tab we use a new “*Key”* to create a filter, in this case we’’l be filtering the institution by country. Using **country\_code** and typing **BR** at the *value* field**,** then clicking *“Send”* one more time, it will display all the brazilians institutes with **status: 200 OK**
2. For this example we will use the Erebor Bank. We can see all the data from that institution, such as the resources it uses, what are the necessary credentials to login and more. In this case, we’ll focus on the credentials.It only needs username and password and that information is inside of **forms\_fields**.

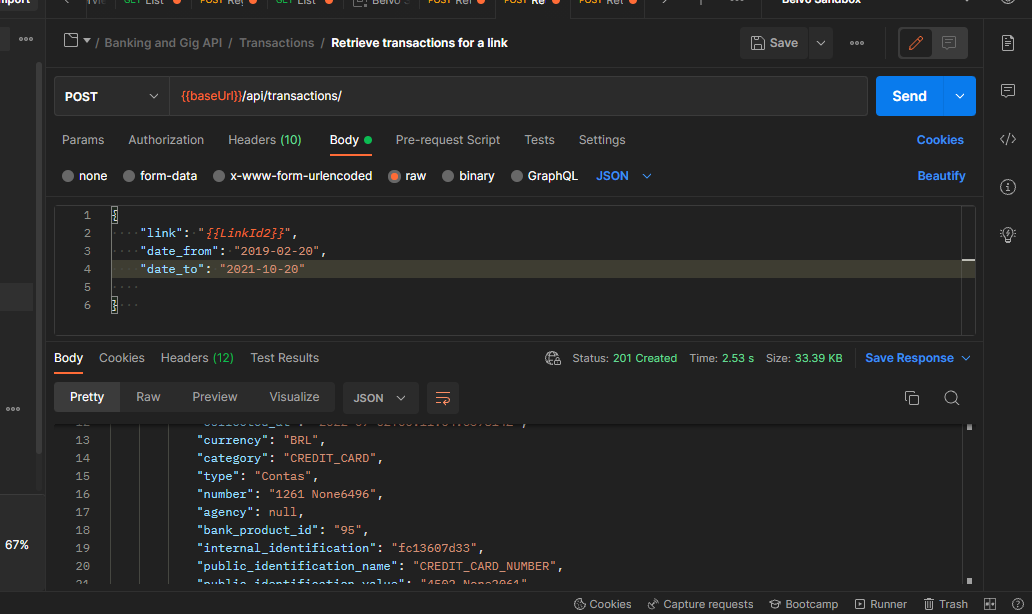


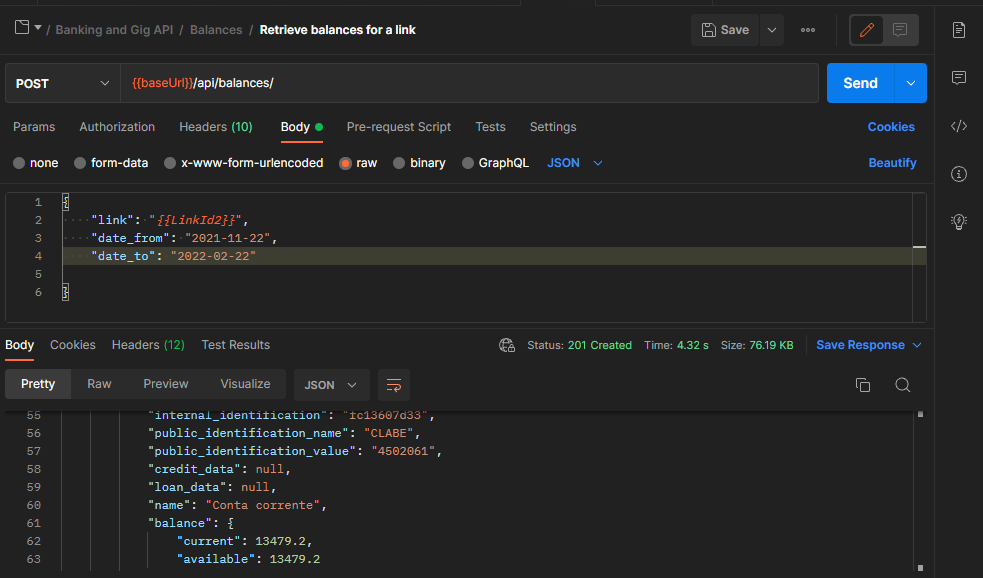
1. After that we should register a new link, for that we can navigate to *Links* > ***Post*** *Register a new link*  > *Body.* In the institution field paste “**erebor\_br\_retail”** and choose the password and username that you wish to register the new link as. After that, press “*Send*” and check if the status displayed is **status:200 OK**, which means that you have successfully registered your link.
2. Your ID was successfully created and next you must create a new variable **LinkId2** so you can **POST** and retrieve the endpoints that you need.

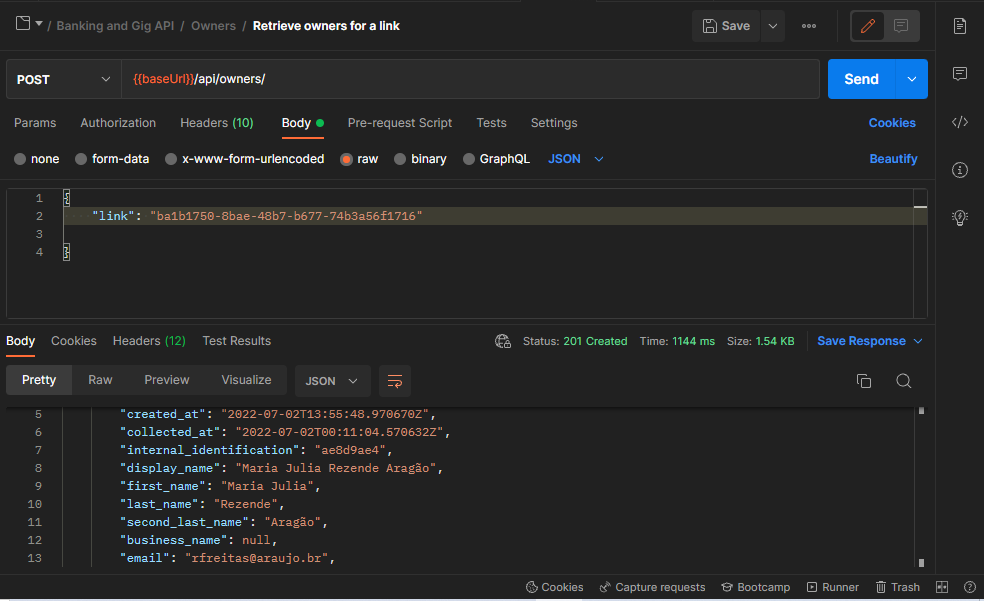
**Now we can access and retrieve the Account, Transaction, Balances and Owners data:**

1. You can retrieve the *Account* for a particular link, that will show you all the accounts that a person could have with the previous bank example (pension fund, loan account, credit card, etc)



2.To access all the *transactions* you need to provide only the “**link**”, “**date\_from**” and “**date\_to**” tags, then press “*Send*”, and all the transactions made during that period will be displayed. Transaction information such as the description, type, category, and when that transaction took place will be included. But you can also use other filters like “**accounts**”: Insert the credit card id for example and the result displayed will be all the credit card transactions.

3. To retrieve the *Balances* you can simply use the tags “**date\_from**” and “**date\_to**” and it will display any period that you wish to check, so you can see the current balance, and the available balance, for example.

4. And last but not least important, we can see all the data from the *Owners*, contact details, name, identity documents and more.

**How the Widget Belvo’s work & How to create a link using backend and frontend**

Belvo's widget is a plug and play solution so the front end can create a link for each user. That widget is then added to your app, I’l next explain how it works.

To create a token for your end user you need to make a call using your backend server and your **SecretId** and **PasswordId** that you created in Belvo’s dashboard, and our system will respond with an access and refresh token.That access token must be added in the widget.

As we strive to make things easier for you, we already provide the coding so you can simply add that Widget to your application.

When you run your application the system will prompt the end user to put his credentials, Belvo’s will create the link ID, and that Link ID will be shared with your server.That way, when you need to access that client’s financial data you only need to provide the link ID and Belvo will retrieve for you all the information needed.

The access token can be used for 60 minutes and will be invalidated if the user logs in successfully at the Widget, but if the token expires there is no need to worry, as you can always refresh it and send a new one for your end user.

**Showing in practice:**

1.In this example we are going to use RunKit and Belvo’s SDKs, to create the access token and the calls, and later on we’ll be using CodePen for the widget.

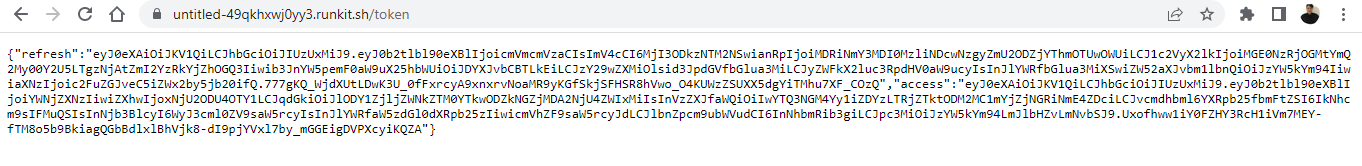
Some points that can’t be missing from your code:

1. Make the calls from Belvo’s SDKs to javascript.
2. Implement your sandbox Keys = **secretId, secretPassword and baseUrl** then instantiate the class Belvo with your credentials.
3. You can create a path using **GET** token and calling the API from Belvo so you can generate the token. Use the command ***client.connect*** for every call that you make.
4. Inside the promise add the informationfrom the widgetToken. Also, it is possible to customize the widget if you wish to, you will only have to add create options.

Then when you are done click on *“endpoint”*

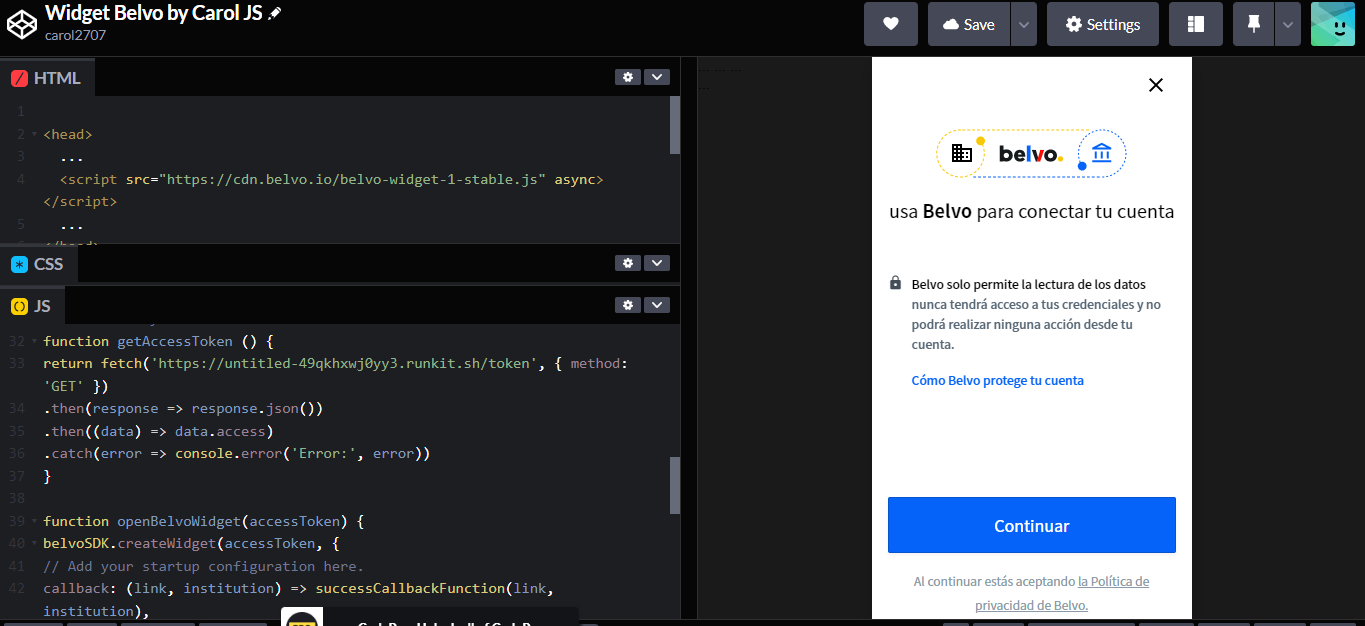


2.For the token to appear you need to change the URL and add “*\token”*so the refresh and access token will be displayed

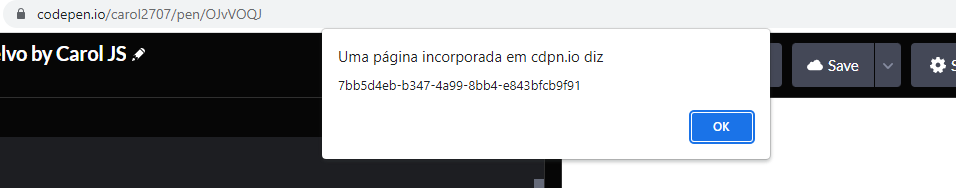


3.Now we need to upload Belvo’s lib script in your application’s <head> tag (src="<https://cdn.belvo.io/belvo-widget-1-stable.js>”) with the parameter async. To functionally load the widget add (<div id="belvo"></div>) anywhere inside the <body>.

Use functions to call the endpoint and bring access information. Once the token is received it will open the widget.



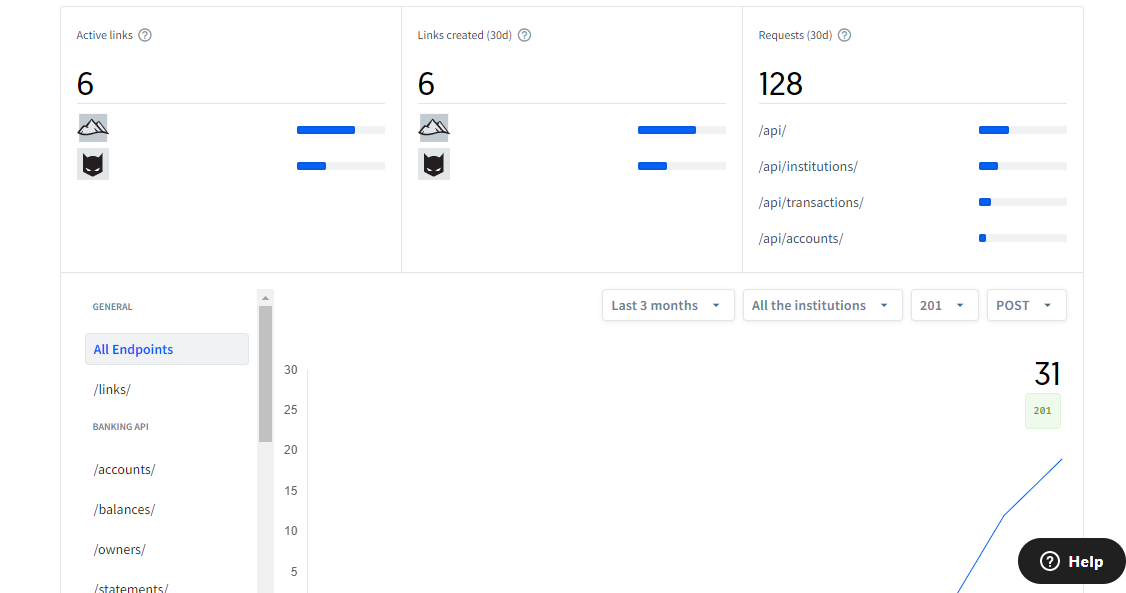
4. You will then receive the **linkId** from Belvo, and will be able to start making the call to retrieve information from that end user.



5. To retrieve the accounts, balances, transactions and owners simply make a call from the backend side with *client.connect*. For example: function *client.accounts.retrieve*

and put the **linkId** (you can always use the SDK). Balances and transactions require you to establish a date.  


6. And you can see your links, requests and so much more in real time at your Belvo’s Dashboard! **Success!**



**Excitement Business idea for Belvo in Brasil**

In Brazil we have a major problem when it comes to Social Security and payroll loans for retirees and pensioners. Thousands of elderly are deceived daily and fall into schemes, and how does it happen?

Legally registered Financial Consulting Companies contact this public that usually don’t have a lot of information about credit, and how the daily factor works (it is a factor calculated using the Selic rate). These representatives offer a contract with banks that usually approve more easily and so are usually the ones that offer the least amount of money, and then give wrong information to those victims, such as “ Your benefit will be blocked if you finalize the contract with other company with a better offer”, or even worse “ You should transfer 20% of what the bank gives you to me, so that the transaction can be complete and you don’t suffer any penalty”. That actually happens, and why does it happen? Because we don’t have autonomy so that the client can see for themself how much the bank is offering, in how many installments, and if it is the only bank they can try to apply for. Would it not be interesting if we could give that autonomy with Belvo's API? And implement it into an app that could work using data from INSS (retirement fund and Social Security) and linked with the banks, sharing the data to all the banks and then directly pass the proposals to the end user (in that case the retirees and pensioners) so they can make the better choice for themselves. It’s possible to give transparency to the user, by creating a new endpoint with only INSS data.

**I prepared a list with the main problems that this Financial Consulting Companies generate:**

1. Harassment of customers (incessant calls until the proposed contract is finalized, contact without prior permission, pressure to complete the transaction and psychological terror and devious argumentation tactics).

2. Dissemination of false information (if you do not use the company's link for the loan, your benefit will be canceled, if you try with other companies, your loan margin may be blocked, among others).

3. Scam attempts (refund requests, non-existent or impossible promises of payment).

4. The worst loan proposals (as it makes no difference to the agent what amount the customer will receive at the end of the transaction, they usually offer the lowest possible amount with the bank that would approve more easily, just to finish the process faster, not caring for the benefit of the customer.)

Anyone can open a credit assignment company only with the aim of harming elderly people with little access to information, in situations of despair due to debts and poor financial management.

**Here I offer a brief description of the proposed solution:**

Autonomy for the client (pensioners and retirees) to send their proposal according to their margin and daily factor. Greater transparency for the client and the possibility of having access to proposals from all banks that would like to link their account and request a loan,

giving permission to the bank to access your INSS data and credits that you want to make available to get a good loan approval.

All this would be possible through Belvo's API widget and database. In this way it would be possible to obtain the best rate and the best proposal.

I noticed that there is currently no application that handles the comparison of the daily factors by credit margin. And that sends the data to the banks and makes possible multiple proposals to different banks with requests for payroll loans. Once the app spreads, I don't see how all retired Brazilians dependant on INSS would not want to have this app downloaded onto their respective devices.

Thank you for your time.

Developed by Ana Caroline Moura Figueira