flinks

Integration Guide

Version 2024.1

Summary

1. **Objective**

2. **Frontend vs. Backend distinction**

3. **Frontend - user interface:**
   a. *Embedded iframe - Flinks Connect*
   b. *Embedded iframe - Flinks Connect (oAuth)*
   c. *Hosted Solution - Flinks Express*

4. **Backend - access to data:**
   a. *Client Dashboard*
   b. *API requests*
   c. *Webhooks*

5. *External Data Attributes*

6. **The Integration Journey**

1. Objective

Welcome to **Flinks** ! We are excited and proud to have you as our partner.

What is this exactly?

The purpose of this guide is to provide an overview on how to integrate our products.
Starting from the no-dev solution and going all the way into the most sophisticated integration, we expect these pages to give you more clarity on the existing options, and serve as reference material throughout your onboarding process.

Is this all the support I'll receive?

Don't worry, our team is here to help clarify any questions that might arise. In order to do so, our team of Technical Account Managers will be on standby. They will be your technical champions at Flinks, and are eager to help you go live asap! Please, never hesitate to reach out and don't forget to CC your Onboarding & Implementation Specialist.

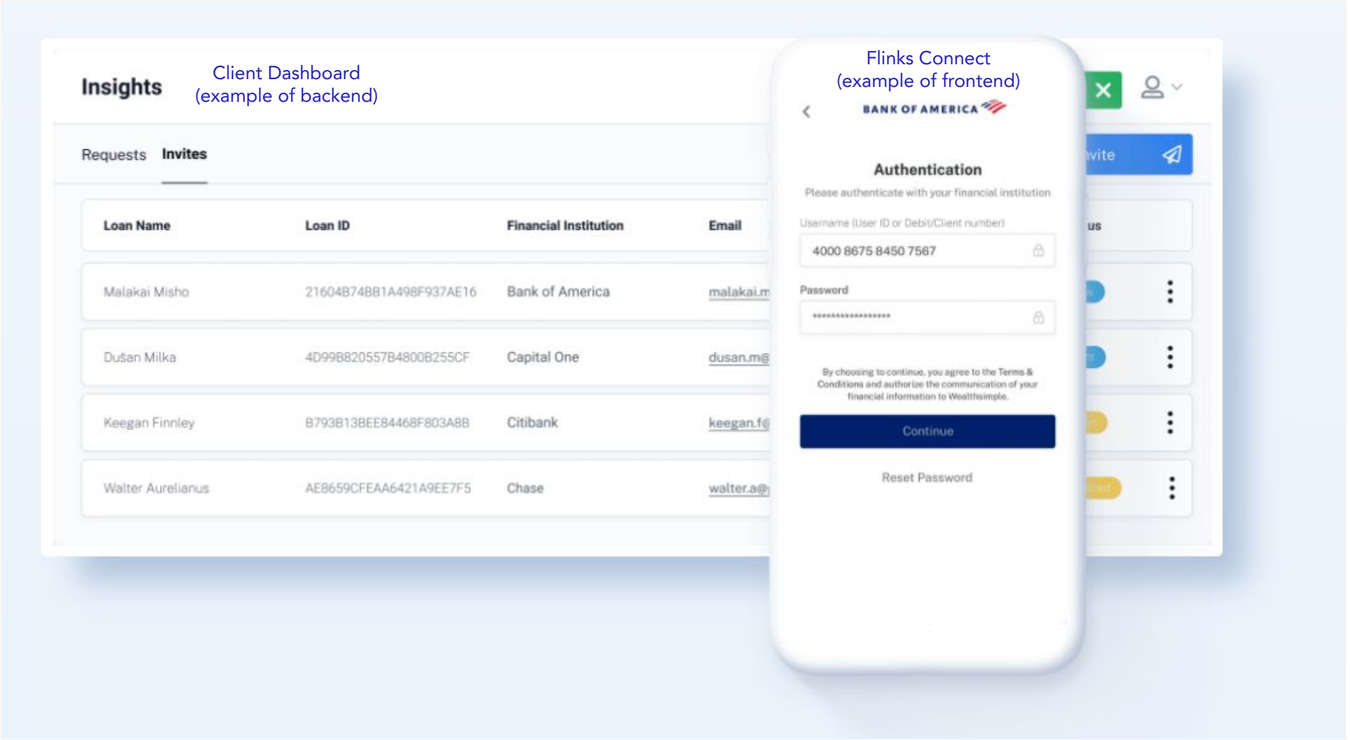
2. Frontend vs. Backend distinction

Before we dive in, there is an important distinction to make.

Flinks is basically composed of two different and independent portions:

- Frontend - what the end user interacts with in order to connect to their financial institution
- Backend - what our partner (meaning you!) will use to have access to the end user's data

For both frontend and backend, Flinks provides different alternatives so you can choose the right solution for your specific needs.



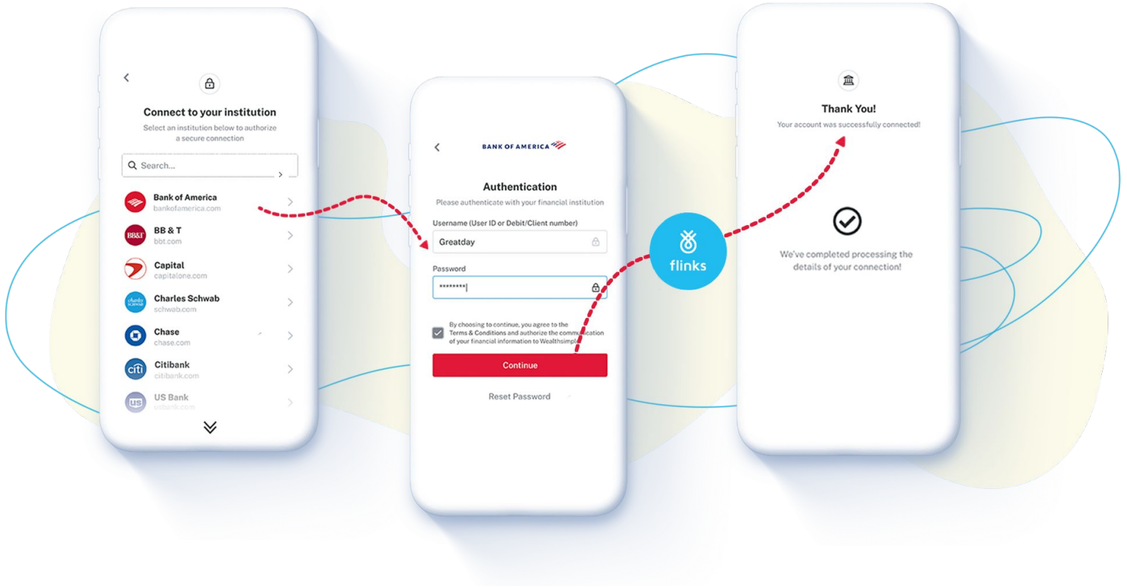
Frontend

3. Frontend: Flinks Connect

Flinks Connect is the iframe which your end-users will interact with in order to:

1. Select their financial institution
2. Provide their credentials (username and password)
3. Respond to MFA - multi-factor authentication, if applicable
4. Connect to their account ✔

Through Flinks Connect, you are able to specify exactly what data points you are looking for by setting the appropriate iframe URL parameters.



Frequently used parameters:

Parameter	Description	Valid Values
consentEnable	Enables our default consent page	True, False
redirectUrl	Redirects the user after connection's completion to URL of choice for	[string URL] ex.: https://www.example.com/thank-you Requires whitelisting on Flinks's side
accountSelectorEnable	Displays list of accounts so the user can select the one of their choice	True, False
withTransactions	Enables transaction pulling	True, False
daysOfTransactions	Date range of transactions	Days90, Days365
detailsAndStatementEnable	Enables PDF statements pulling	True, False
scheduleRefresh	Enables Nightly Refreshes	True, False

Relevant Doc Links:

- Flinks Connect Configurator
- Parameters

3. Frontend: Flinks Connect

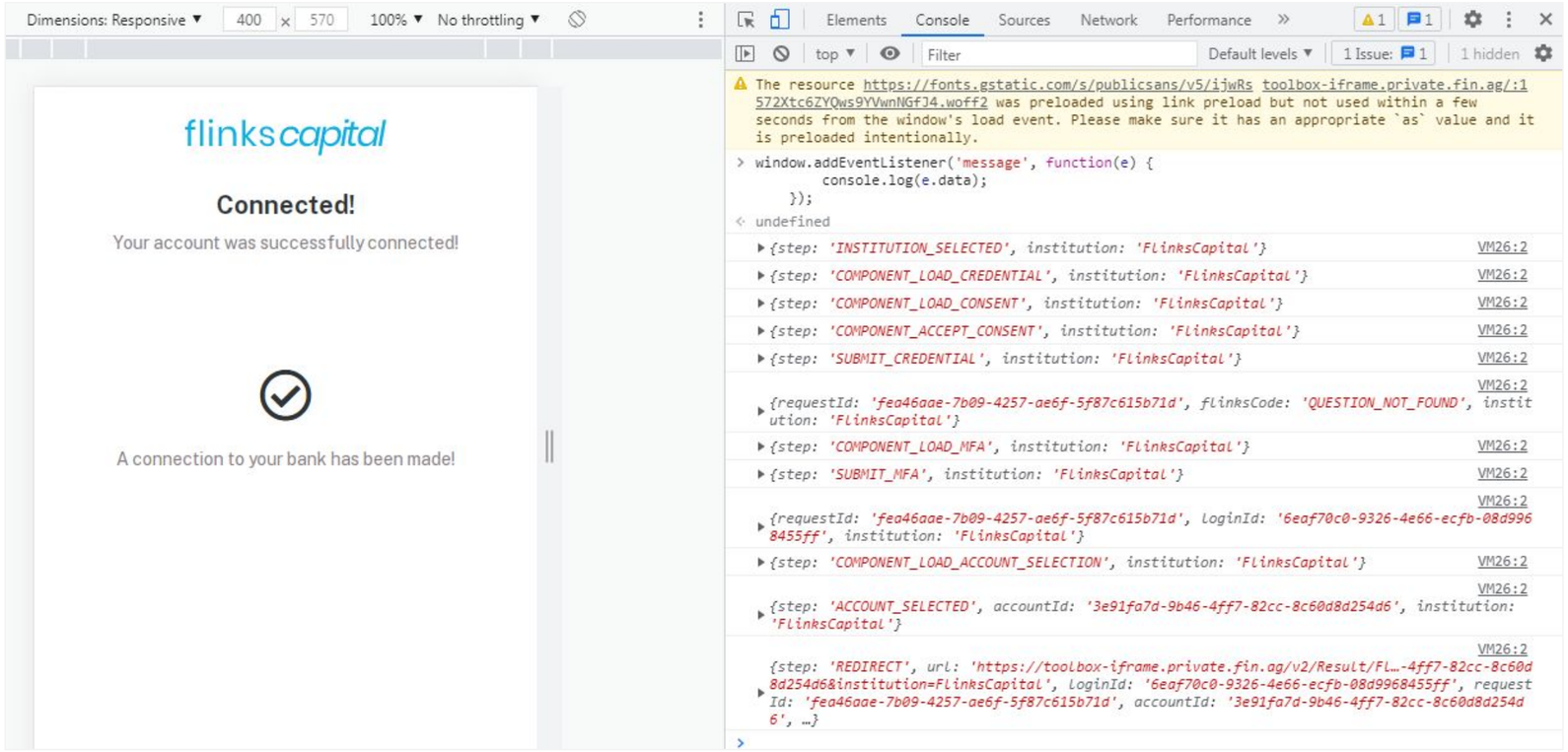
You can keep track of the users' steps within the iframe, and most importantly, fetch the IDs that identify that connection and are necessary to pull the data from our API.

And how to do that?

For each step, the iframe triggers events. Make sure to enable the Event Listener in your script to leverage this feature.

Relevant Doc Links:

- Events

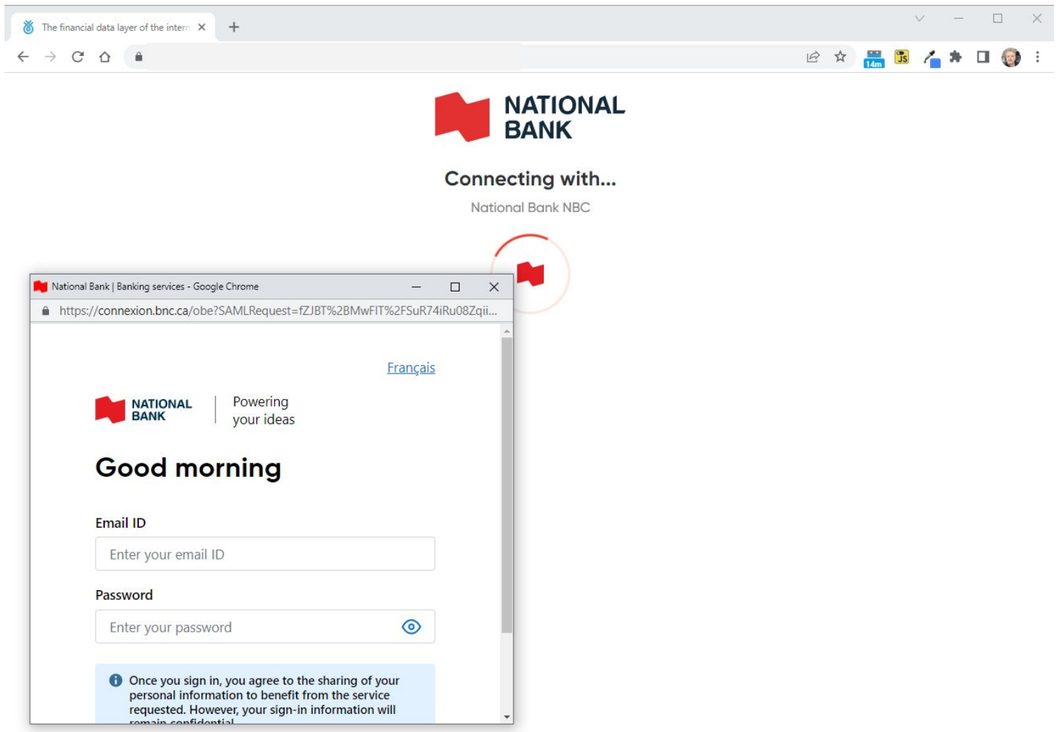


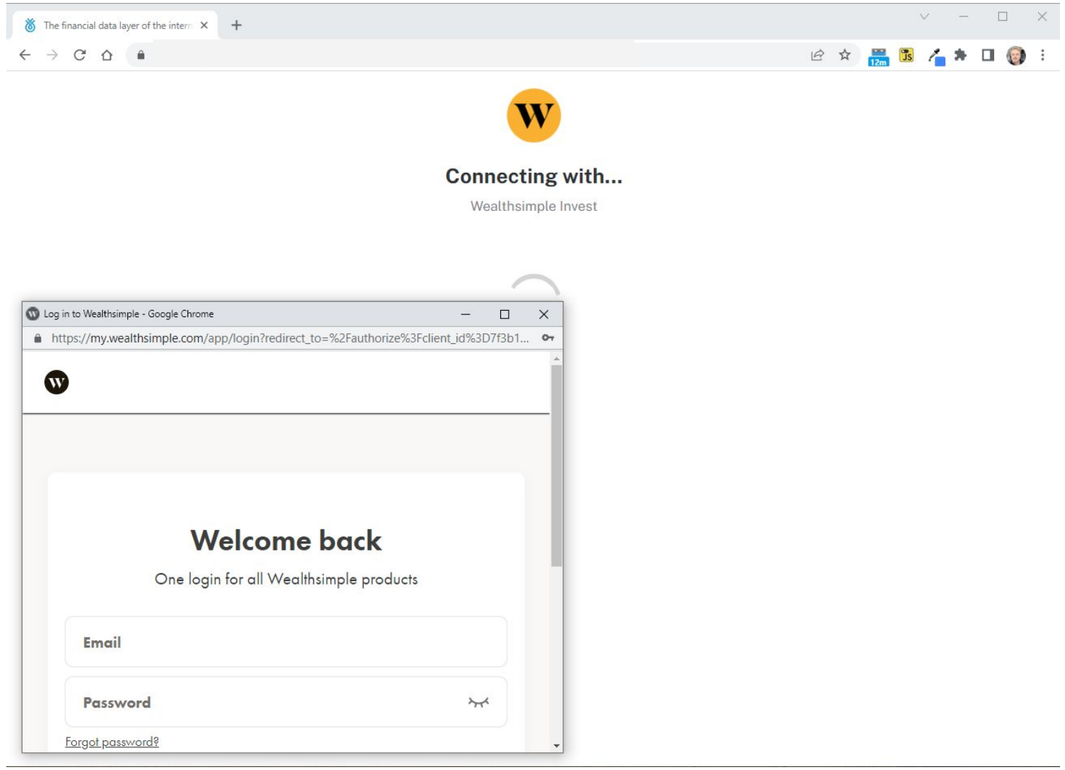
3. Frontend: Flinks Connect (oAuth)

In oAuth connections, users will no longer be required to provide their credentials through Flinks Connect. If oAuth connectivity is available for an FI, a pop-up will appear and their users then will login directly into their institutions.

Please make sure that your page or app allows this pop-up to open!

For this implementation, we strongly recommend the use of the redirectUrl parameter, which will allow Flinks Connect to bring the user back to your page or app* after completing the flow.

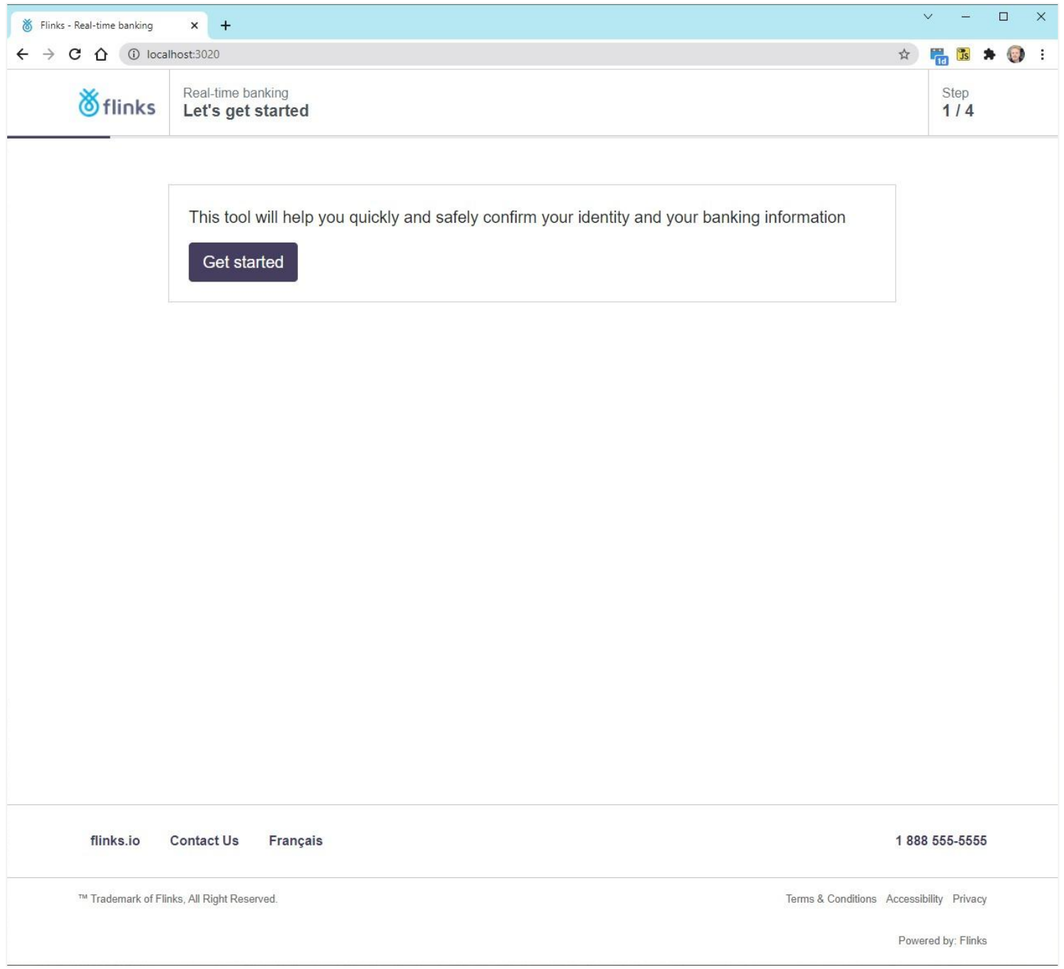




*: this URL needs to deep link back to your app

3. Frontend: Flinks Express

Flinks Express is the fastest way of using our products - no code or developer needed!
It consists of a 4-step flow where the user will be able to connect to Flinks, without the need of embedding the iframe in your script.
By opting to this solution, we will customize a template that embeds the iframe for you, as if the page were your own.



What is customizable??
In Flinks Express, we can use your own:

1. Logo and brand's theme color;
2. Wording/language of your choosing, appropriate for your public;
3. Links to your company's Privacy Policy, Terms & Conditions, etc.;
4. Know-your-customer form for additional information on the users.

Through this link, you will be able to check and submit all of the customizations for your use case. Make sure to let your Onboarding & Implementation Specialist know after your submission!

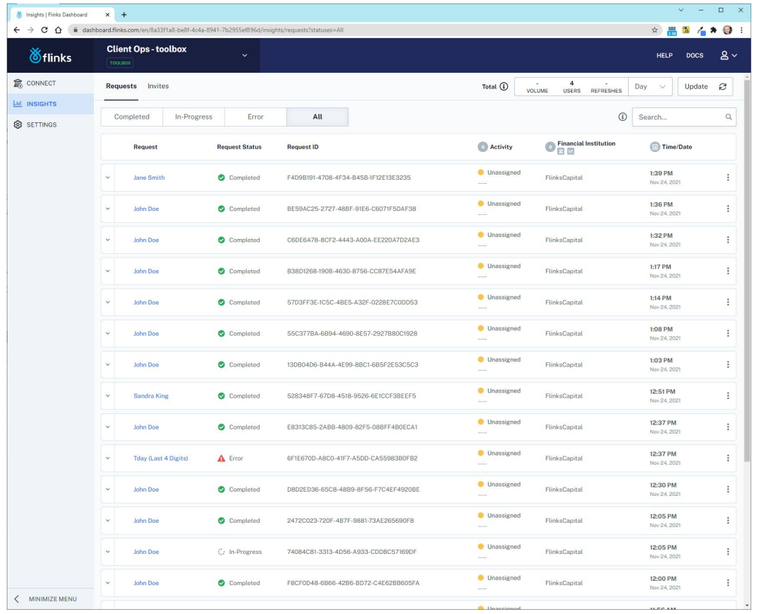
Sample: https://flinkstoolbox.flinksapp.io

Backend

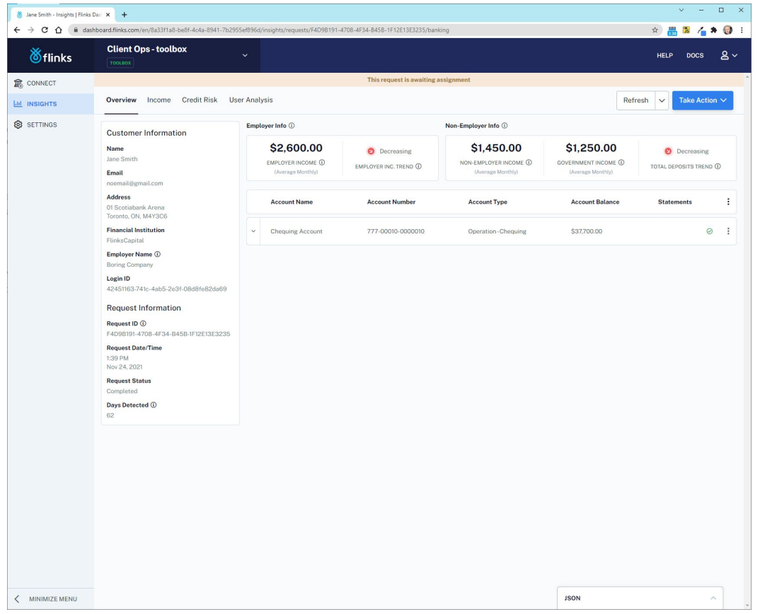
4. Backend: Client Dashboard

The Client Dashboard is the most straightforward solution in order to have access to the data returned by Flinks. This tool is available regardless of the chosen type of integration, hence it coexists with your API pulling or webhook integration.

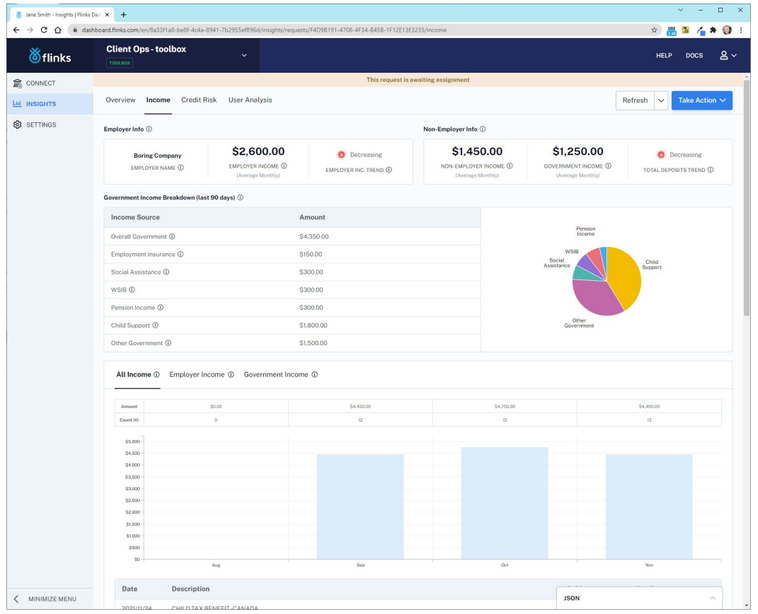
After your production instance is created, our team will link it and invite you to open your Dashboard account. A walkthrough will be provided then, but you can always refer to our Training Videos at anytime to make sure you extract the most of this amazing tool.



List of Requests



User account overview

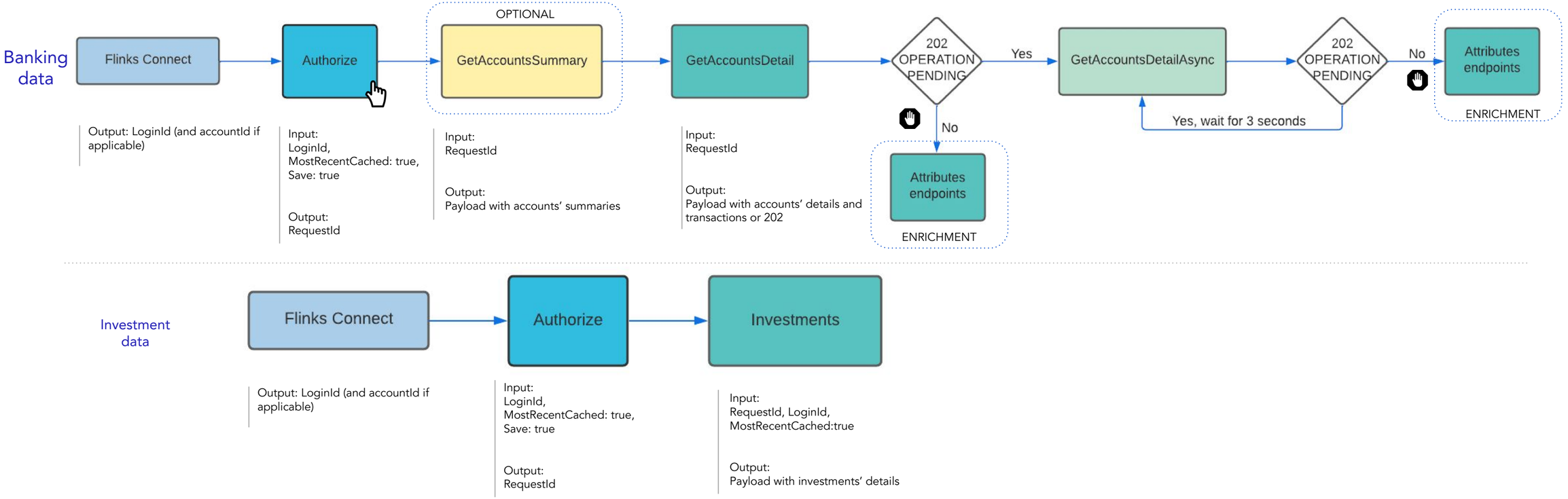


Attributes report

4. Backend: API requests

After the user has connected through Flinks Connect (iframe) and the LoginId (and accountId, if applicable) have been fetched, it's time to engage the API flow in order to retrieve their data. On a high level, the following diagram represents the sequence on how the endpoints should be called.

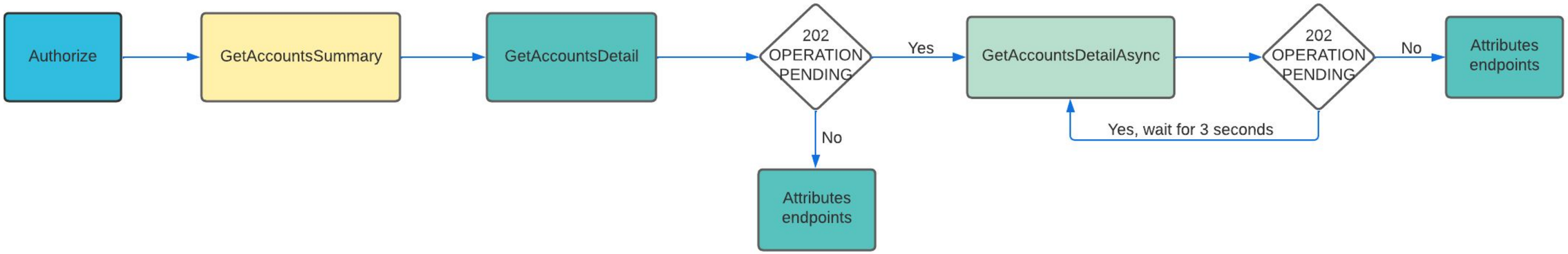
Below each box that represents a specific endpoint, you will find the common inputs (parameters) and outputs (responses) for most use cases. Nevertheless, you will also find all the possibilities at Flinks API Reference in our public documentation. If you'd like to jump right into any of the following endpoints, just click on their respective box!



4. Backend: API requests (async flow)

The Asynchronous flow may occur when following the previous steps. In practice, it happens when /GetAccountsDetail is not done aggregating the data, so it will return a 202 - OPERATION PENDING status code.

If so, you need to start calling a different endpoint, called /GetAccountsDetailAsync until it returns the data, as described in the flow below.



Relevant Doc Links:

- Asynchronous GetAccountsDetail Flow

4. Backend: Webhooks

The last integration type is the most powerful: webhooks. By opting for it, you will not need to be concerned about which endpoints to call, neither with the correct order. We will deliver the data to you as soon as it's ready to go.

We currently support 3 different kinds of webhooks:

- KYC - anticipates the return of user's personal information as soon as fetched from the FI, while the accounts' information is still being pulled from the FI;
- GetAccountsDetail - returns this endpoints payload as a webhook (KYC + IBV + Transactions);
- Investments - returns this endpoints payload as a webhook (Securities + Positions + Transactions)

And how do I enable it??

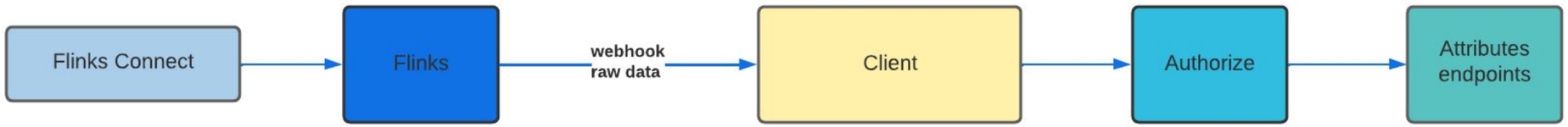
It's simple, just send us a webhook URL and we will set everything up on our end. Please make sure to return a 200 whenever you receive a webhook, so we can know that everything worked fine.

What if I also have Attributes??

For the moment, we do not support Attributes' payloads to be delivered via webhooks. However, the webhook with raw aggregated data can be used as a trigger to your Attributes API calls!

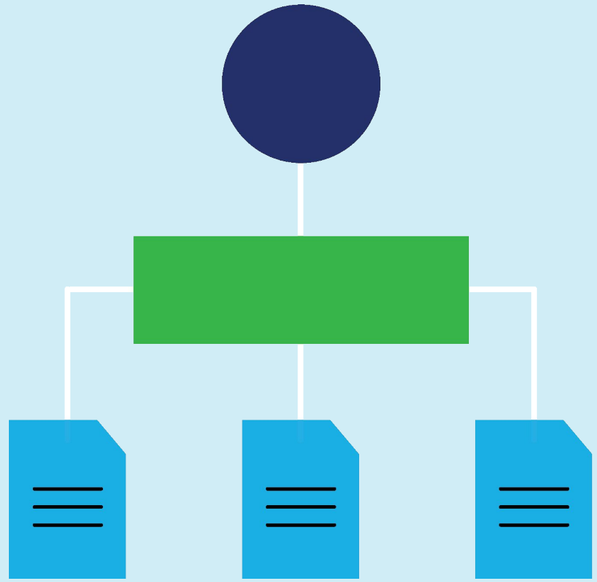
SAMPLE PAYLOAD


```
{
  "HttpStatusCode": 200,
  "Accounts": [
    {
      "Transactions": [
        {
          "Date": "2022-06-23",
          "Code": null,
          "Description": "national money",
          "Debit": 23.66,
          "Credit": null,
          "Balance": 49976.34,
          "Id": "16e2ca5a-db75-49ac-bde3-25af0e0e3982"
        },...
      ],
      "TransitNumber": "77777",
      "InstitutionNumber": "777",
      "OverdraftLimit": 0,
      "Title": "Chequing CAD",
      "AccountNumber": "1111000",
      "Balance": {
        "Available": null,
        "Current": 49976.34,
        "Limit": null
      },
      "Category": "Operations",
      "Type": "Chequing",
      "Currency": "CAD",
      "Holder": {
        "Name": "John Doe",
        "Address": {
          "CivicAddress": "1275 avenue des Canadiens-de-Montréal",
          "City": "Montréal",
          "Province": "QC",
          "PostalCode": "H3B 5E8",
          "POBox": null,
          "Country": "CA"
        },
        "Email": "johndoe@flinks.io",
        "PhoneNumber": "(514) 333-7777"
      },
      "Id": "cce27243-24fc-4250-a937-f4da0b0deb26"
    }
  ],
  "Login": {
    "Username": "Greatday",
    "IsScheduledRefresh": false,
    "LastRefresh": "2022-06-23T21:09:50.7410533Z",
    "Type": "Personal",
    "Id": "ce8a9c30-5ff1-41dc-c4da-08da553f5f08"
  },
  "Institution": "FlinksCapital",
  "RequestId": "438ee77d-3f44-47a6-8fb8-5dbef83322a0"
}
```

Flow with Webhooks and Attributes

External Data Attributes



5. External Data Attributes

In case you have transactional data from different providers, or even data from previous Flinks connections that is no longer available in your private instance, External Data Attributes can be used to enrich it!

This can be done by simply uploading the transactions through one of our /Upload endpoints, and as a response you will receive the exact same Attributes report, according to the package of your choosing.

Please reach out to our team for more information around the expected transaction format.

Example: Lending


```
{
    "Options": {
        "OriginCountry": "ca",
        "OldestBalance": 1000
    },
    "transactions": [{
        "AccountId": "3b9ddd96-ef7a-49cf-acb6-e2a891d13daa",
        "Description": "THOM CONTRA CR PAY",
        "Debit": 0.0,
        "Credit": 2541.33,
        "TransactionDate": "2021-08-06",
        "TransactionId": "9191bbc5-825b-47e0-85d7-9a6d1a425ab6"
    }, {
        "AccountId": "3b9ddd96-ef7a-49cf-acb6-e2a891d13daa",
        "Description": "PAYROLL DEPOSIT PAY",
        "Debit": 0.0,
        "Credit": 1549.3,
        "TransactionDate": "2021-09-03",
        "TransactionId": "d12403a8-51aa-407f-8723-7b9278c4fe09"
    }, {
        "AccountId": "3b9ddd96-ef7a-49cf-acb6-e2a891d13daa",
        "Description": "1663522 AB LTD PAY",
        "Debit": 0.0,
        "Credit": 771.52,
        "TransactionDate": "2021-09-17",
        "TransactionId": "beee2d1f-0f25-414c-a060-63a5e705e647"
    }, {
        "AccountId": "3b9ddd96-ef7a-49cf-acb6-e2a891d13daa",
        "Description": "Fort McKay Stra PAY",
        "Debit": 0.0,
        "Credit": 2227.98,
        "TransactionDate": "2021-09-17",
        "TransactionId": "ce65a676-1578-4a15-b162-20f3475f1081"
    }]
}
```


Transactions upload

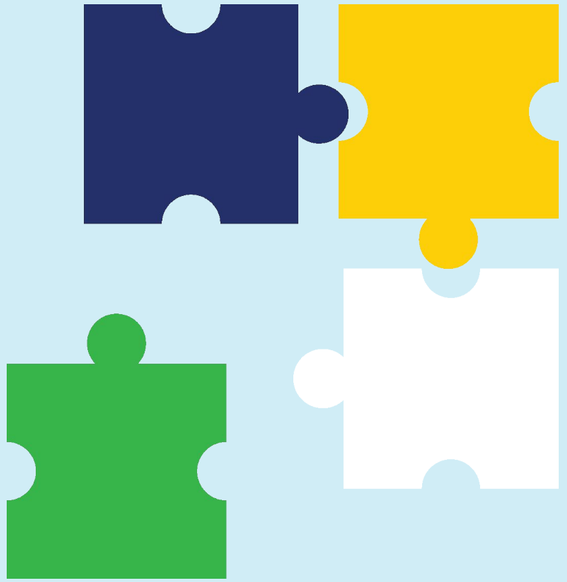

```
{
    "HttpStatusCode": 200,
    "Card": {
        "Id": "916b23e0-5d9a-42f2-a04a-08da60320fe0",
        "account_age_days": 85,
        "active_days_trend": 1.42,
        "auto_loan_payment_frequency": "NONE_DETECTED",
        "average_closing_balance_day_after_employer_income": 7922.99,
        "average_closing_balance_day_after_government_income": "NONE_DETECTED",
        "average_closing_balance_day_after_income": 7922.99,
        "average_closing_balance_day_of_employer_income": 7922.99,
        "average_closing_balance_day_of_government_income": "NONE_DETECTED",
        "average_closing_balance_day_of_income": 6898.26,
        "average_disability_deposit": 0.0,
        "average_employer_a_income_deposit": 2541.33,
        "average_employer_b_income_deposit": 2106.51,
        "average_employer_income_deposit": 1511.68,
        "average_employer_other_income_deposit": 650.03,
        "average_employment_insurance_deposit": 0.0,
        "average_government_income_deposit": 0.0,
        "average_micro_loan_payment": 0.0,
        "average_monthly_auto_loan_payments_complex": 0.0,
        "average_monthly_child_support_income_government_complex": 0.0,
        "average_monthly_disability_income_complex": 0.0,
        "average_monthly_employer_a_income_complex": 0.0,
        "average_monthly_employer_b_income_complex": 4213.02,
        "average_monthly_employer_income_complex": 4548.80,
        "average_monthly_employer_other_income_complex": 1300.06,
        "average_monthly_employment_insurance_income_complex": 0.0,
        "average_monthly_free_cash_flow": 5335.34,
        "average_monthly_government_income_complex": 0.0,
        "average_monthly_loan_deposits_complex": 0.0,
        "average_monthly_loan_payments_complex": 0.0,
        "average_monthly_micro_loan_payments_complex": 0.0,
        "average_monthly_mortgage_payments_complex": 0.0,
        "average_monthly_non_employer_income_complex": 0.0,
        "average_monthly_nsf_fees_count": 0.0,
        "average_monthly_other_loan_payments_complex": 0.0,
        "average_monthly_pension_income_complex": 0.0,
        "average_monthly_recurring_transfers_in_complex": 0.0,
        "average_monthly_recurring_transfers_out_complex": 0.0,
        "average_monthly_social_assistance_income_complex": 0.0,
    }
```


Response (extract)

Relevant Doc Links:

- External Data Overview
- Transforming External Transactions
- /Upload (Base Example)
  In certain steps, the above link will refer to our core Attributes Documentation, so please find below a link to the Attributes API docs: Attributes
- Common Error Codes

The Integration Journey



6. The Integration Journey

Signed with Flinks already? Here's what to expect as next steps:

1 Kick-off: week 1

- Introduction to Flinks team, roles & responsibilities, and support.
- Flinks demo of end-user experience on the front end.
- Flinks demo of customer dashboard

2 Flinks Setup: week 2

- Flinks to setup credentials, instance, and client dashboard
- Customer to provide who needs access to client dashboard with what access
- Flinks to set up test accounts as required by Customer

3 Docs & Data Review: week 3 and 4

- Customer to review API Documentation, Integration Documentation, and UX/UI Best Practices
- Customer to map Flink's data points internally

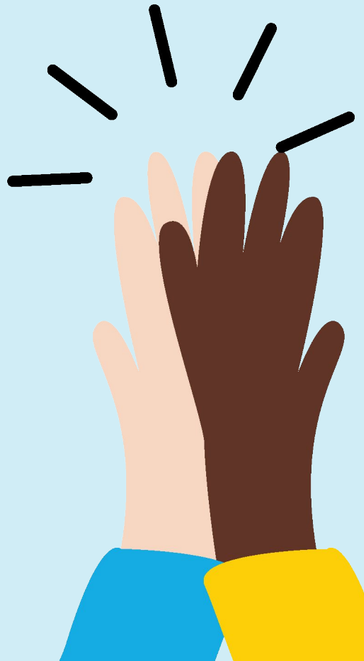
4 Implementation: weeks 5 to 7

- Timeline governed by Customer
- Weekly integration calls and support provided by Flinks

Communication channels:

- Onboarding & Implementation Specialist *(anything integration related)*
- Technical support channels:
  - help-integration@flinks.com
  - Your dedicated Technical Account Manager *(if applicable)*

Already live? Then these are your champions:

- Relationship Manager *(products, billing, marketing initiatives, …)*
- Support channels *(support with specific connections - RequestIds or LoginIds required):*
  - Canada: help-ca@flinks.com
  - US: help-us@flinks.com
  - Your dedicated Technical Account Manager *(if applicable)*

flinks