# [Introduction](https://plaid.com/docs/quickstart/" \l "introduction)

Let’s test out running Plaid locally by cloning the Quickstart app. You’ll need API keys, which you can receive by signing up in the [Dashboard](https://dashboard.plaid.com/developers/keys).

You'll have two different API keys, and there are three different Plaid environments. Today we'll start in the Sandbox environment. View the API Keys section of the Dashboard to find your Sandbox secret.

# API Key

## client\_id

Private identifier for your team

## secret

Private key, one for each of the three environments

# Environment

## Sandbox

Get started with test credentials and life-like data

## Development

Build out your app with up to 100 live credentials

## Production

Launch your app with unlimited live credentials

If you get stuck at any point in the Quickstart, help is just a click away! Check the Quickstart [troubleshooting guide](https://github.com/plaid/quickstart#troubleshooting), ask other developers in our [Stack Overflow community](https://stackoverflow.com/questions/tagged/plaid) or submit a [Support ticket](https://dashboard.plaid.com/support/new/product-and-development/developer-lifecycle/quickstart-issues).

# [Quickstart setup](https://plaid.com/docs/quickstart/" \l "quickstart-setup)

Once you have your API keys, it's time to run the Plaid Quickstart locally! The instructions below will guide you through the process of cloning the [Quickstart repository](https://github.com/plaid/quickstart), customizing the .env file with your own Plaid client ID and Sandbox secret, and finally, building and running the app.

Plaid offers both Docker and non-Docker options for the Quickstart. If you don't have Docker installed, you may wish to use the non-Docker version; this path is especially recommended for Windows users who do not have Docker installations. However, if you already have Docker installed, we recommend the Docker option because it is simpler and easier to run the Quickstart. Below are instructions on setting up the Quickstart with Docker and non-Docker configurations.

Select group for content switcher

## Non-DockerDocker

[Setting up without Docker](https://plaid.com/docs/quickstart/" \l "setting-up-without-docker)

Make sure you have [npm installed](https://docs.npmjs.com/downloading-and-installing-node-js-and-npm) before following along. If you're using Windows, ensure you have a terminal capable of running basic Unix shell commands.

1. Clone the Quickstart and run the backend

`1 # not applicable`

Open a new shell and start the frontend app. Your app will be running at http://localhost:3000.

2. Run the Quickstart frontend

```

1# Install dependencies

2cd quickstart/frontend

3npm install

4

5# Start the frontend app

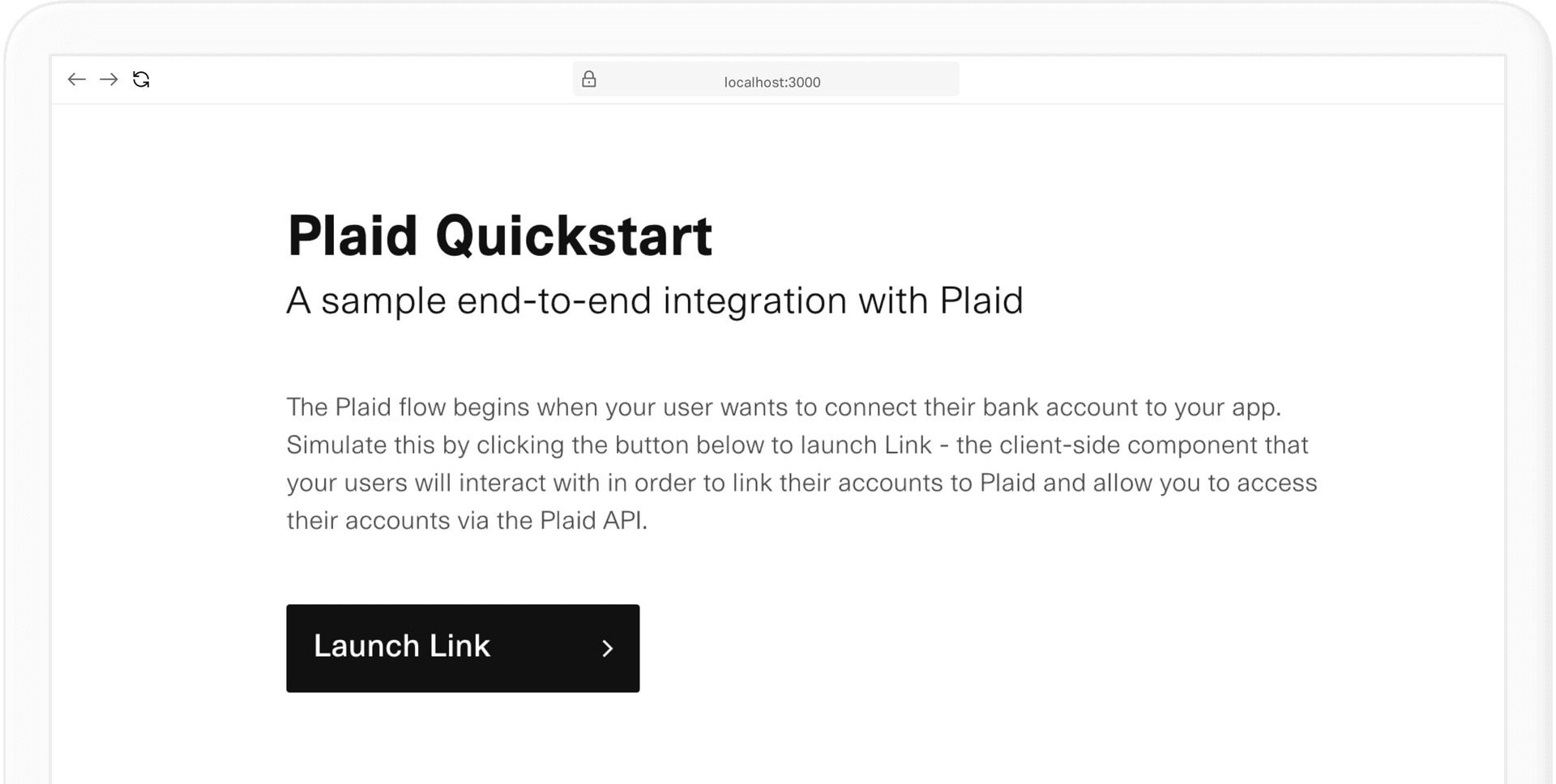
6npm start

7

8# Go to <http://localhost:3000>

```

Visit localhost and log in with Sandbox credentials (typically user\_good and pass\_good, as indicated at the bottom of the page).



[Setting up with Docker](https://plaid.com/docs/quickstart/" \l "setting-up-with-docker)

Docker is a software platform that packages software into standardized units called containers that have everything the software needs to run, including libraries, system tools, code, and runtime. If you don't already have Docker, you can [download it from the Docker site](https://docs.docker.com/get-docker/). Note that Windows users may need to take some additional steps, such installing a Linux environment; if you are using Windows and do not already have a Linux environment installed, we recommend selecting the non-Docker instructions instead.

Once Docker is installed, launch the Docker app, then use the following commands at the command line to configure and run the Quickstart. If the make commands do not work, ensure that Docker is running. You may also need to prefix the make commands with sudo, depending on your environment.

1. Clone and run the Quickstart

```

1# Note: If on Windows, run

2# git clone -c core.symlinks=true https://github.com/plaid/quickstart

3# instead to ensure correct symlink behavior

4

5git clone https://github.com/plaid/quickstart.git

6cd quickstart

7

8# Copy the .env.example file to .env, then fill

9# out PLAID\_CLIENT\_ID and PLAID\_SECRET in .env

10cp .env.example .env

11

12# start the container for one of these languages:

13# node, python, java, ruby, or go

14

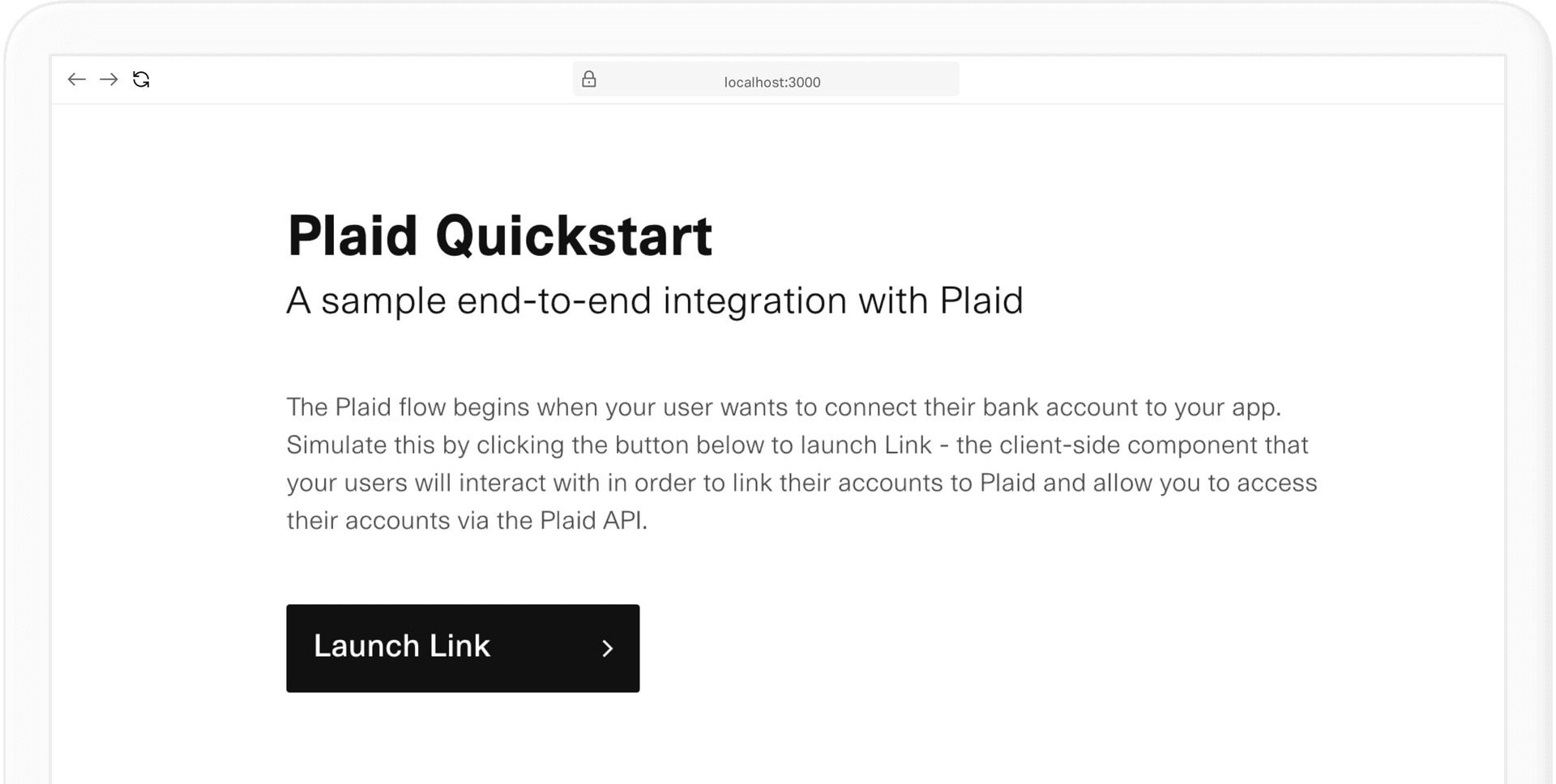
15make up language=node

16

17# Go to <http://localhost:3000>

```

Visit localhost and log in with Sandbox credentials (typically user\_good and pass\_good, as indicated at the bottom of the page).



View the logs

`1$ make logs language=node`

Stop the container

`1$ make stop language=node`

[Create your first Item](https://plaid.com/docs/quickstart/" \l "create-your-first-item)

Most API requests interact with an *Item*, which is a Plaid term for a login at a financial institution. A single end-user of your application might have accounts at different financial institutions, which means they would have multiple different Items. An Item is not the same as a financial institution account, although every account will be associated with an Item. For example, if a user has one login at their bank that allows them to access both their checking account and their savings account, a single Item would be associated with both of those accounts.

Now that you have the Quickstart running, you’ll add your first Item in the Sandbox environment. Once you’ve opened the Quickstart app on localhost, click the **Launch Link** button and select any institution. Use the Sandbox credentials to simulate a successful login.

**[Sandbox credentials](https://plaid.com/docs/quickstart/" \l "sandbox-credentials)**

```

1username: user\_good

2password: pass\_good

3If prompted to enter a 2FA code: 1234

```

Once you have entered your credentials and moved to the next screen, you have created your first Item! You can now make API calls for that Item by using the buttons in the Quickstart. In the next section, we'll explain what actually happened and how the Quickstart works.

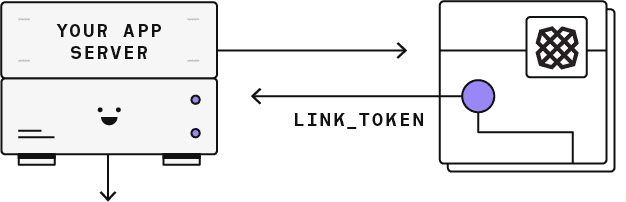
[How it works](https://plaid.com/docs/quickstart/" \l "how-it-works)

As you might have noticed, you use both a server and a client-side component to access the Plaid APIs. The flow looks like this:

**The Plaid flow** begins when your user wants to connect their bank account to your app.



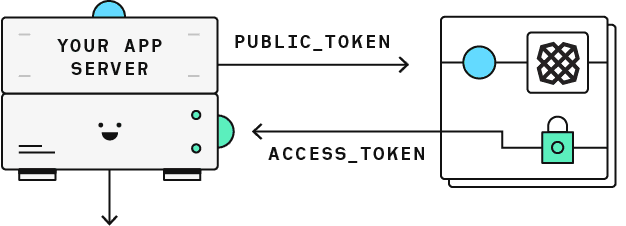
**1**Call [**/link/token/create**](https://plaid.com/docs/api/tokens/#linktokencreate) to create a link\_token and pass the temporary token to your app's client.



**2**Use the link\_token to open Link for your user. In the [**onSuccess** callback](https://plaid.com/docs/link/web/#onsuccess), Link will provide a temporary public\_token.



**3**Call [**/item/public\_token/exchange**](https://plaid.com/docs/api/tokens/#itempublic_tokenexchange) to exchange the public\_token for a permanent access\_token and item\_id for the new Item.



**4**Store the access\_token and use it to make product requests for your user's Item.



The first step is to create a new link\_token by making a [**/link/token/create**](https://plaid.com/docs/api/tokens/#linktokencreate) request and passing in the required configurations. This link\_token is a short lived, one-time use token that authenticates your app with [Plaid Link](https://plaid.com/docs/link/), our frontend module. Several of the environment variables you configured when launching the Quickstart, such as PLAID\_PRODUCTS, are used as parameters for the link\_token.

Select group for content switcher

Current librariesLegacy libraries

```

1curl -X POST https://sandbox.plaid.com/link/token/create \

2-H 'Content-Type: application/json' \

3-d '{

4 "client\_id": "${PLAID\_CLIENT\_ID}",

5 "secret": "${PLAID\_SECRET}",

6 "client\_name": "Plaid Test App",

7 "user": { "client\_user\_id": "${UNIQUE\_USER\_ID}" },

8 "products": ["${PRODUCT}"],

9 "country\_codes": ["US"],

10 "language": "en",

11 "webhook": "https://webhook.example.com",

12 "redirect\_uri": "https://domainname.com/oauth-page.html",

13}'

```

Once you have a link\_token, you can use it to initialize [Link](https://plaid.com/docs/link/). Link is a drop-in client-side module available for web, iOS, and Android that handles the authentication process. The Quickstart uses Plaid's optional React bindings for an integration that you trigger via your own client-side code. This is what your users use to log into their financial institution accounts.

After a user submits their credentials within Link, Link provides you with a public\_token via the onSuccess callback. The code below shows how the Quickstart passes the public\_token from client-side code to the server. Both React and vanilla JavaScript examples are shown.

Initialize Link

```

1<button id="link-button">Link Account</button>

2<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

3<script src="https://cdn.plaid.com/link/v2/stable/link-initialize.js"></script>

4<script type="text/javascript">

5(async function($) {

6 var handler = Plaid.create({

7 // Create a new link\_token to initialize Link

8 token: (await $.post('/create\_link\_token')).link\_token,

9 onLoad: function() {

10 // Optional, called when Link loads

11 },

12 onSuccess: function(public\_token, metadata) {

13 // Send the public\_token to your app server.

14 // The metadata object contains info about the institution the

15 // user selected and the account ID or IDs, if the

16 // Account Select view is enabled.

17 $.post('/exchange\_public\_token', {

18 public\_token: public\_token,

19 });

20 },

21 onExit: function(err, metadata) {

22 // The user exited the Link flow.

23 if (err != null) {

24 // The user encountered a Plaid API error prior to exiting.

25 }

26 // metadata contains information about the institution

27 // that the user selected and the most recent API request IDs.

28 // Storing this information can be helpful for support.

29 },

30 onEvent: function(eventName, metadata) {

31 // Optionally capture Link flow events, streamed through

32 // this callback as your users connect an Item to Plaid.

33 // For example:

34 // eventName = "TRANSITION\_VIEW"

35 // metadata = {

36 // link\_session\_id: "123-abc",

37 // mfa\_type: "questions",

38 // timestamp: "2017-09-14T14:42:19.350Z",

39 // view\_name: "MFA",

40 // }

41 }

42 });

43

44 $('#link-button').on('click', function(e) {

45 handler.open();

46 });

47})(jQuery);

48</script>

```

Next, on the server side, the Quickstart calls [**/item/public\_token/exchange**](https://plaid.com/docs/api/tokens/#itempublic_tokenexchange) to obtain an access\_token, as illustrated in the code excerpt below. The access\_token uniquely identifies an Item and is a required argument for most Plaid API endpoints. In your own code, you'll need to securely store your access\_token in order to make API requests for that Item.

Select group for content switcher

Current librariesLegacy libraries

```

1curl -X POST https://sandbox.plaid.com/item/public\_token/exchange \

2-H 'Content-Type: application/json' \

3-d '{

4 "client\_id": "${PLAID\_CLIENT\_ID}",

5 "secret": "${PLAID\_SECRET}",

6 "public\_token": "public-sandbox-12345678-abcd-1234-abcd-1234567890ab"

7}'

```

[Making API requests](https://plaid.com/docs/quickstart/" \l "making-api-requests)

Now that we've gone over the Link flow and token exchange process, we can explore what happens when you press a button in the Quickstart to make an API call. As an example, we'll look at the Quickstart's call to [**/accounts/get**](https://plaid.com/docs/api/accounts/#accountsget), which retrieves basic information, such as name and balance, about the accounts associated with an Item. The call is fairly straightforward and uses the access\_token as a single argument to the Plaid client object.

/accounts/get

```

1 curl -X POST https://sandbox.plaid.com/accounts/get \

2 -H 'Content-Type: application/json' \

3 -d '{

4 "client\_id": "${PLAID\_CLIENT\_ID}"

5 "secret": "${PLAID\_SECRET}"

6 "access\_token": String

7 }'

```

Example response data:

/accounts/get response

```

1{

2 "accounts": [

3 {

4 "account\_id": "A3wenK5EQRfKlnxlBbVXtPw9gyazDWu1EdaZD",

5 "balances": {

6 "available": 100,

7 "current": 110,

8 "iso\_currency\_code": "USD",

9 "limit": null,

10 "unofficial\_currency\_code": null

11 },

12 "mask": "0000",

13 "name": "Plaid Checking",

14 "official\_name": "Plaid Gold Standard 0% Interest Checking",

15 "subtype": "checking",

16 "type": "depository"

17 },

18 {

19 "account\_id": "GPnpQdbD35uKdxndAwmbt6aRXryj4AC1yQqmd",

20 "balances": {

21 "available": 200,

22 "current": 210,

23 "iso\_currency\_code": "USD",

24 "limit": null,

25 "unofficial\_currency\_code": null

26 },

27 "mask": "1111",

28 "name": "Plaid Saving",

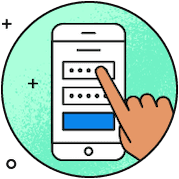
29 "official\_name": "Plaid Silver Standard 0.1% Interest Saving",

```

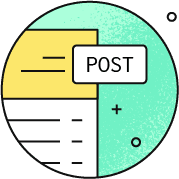
# AUTH

# Introduction to Auth

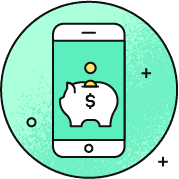
#### Instantly retrieve account information to set up online payments via ACH and more.



LINK AN ACCOUNT



FETCH ACCOUNT INFO



INITIATE A PAYMENT

#### [API Reference](https://plaid.com/docs/api/products/auth/)

[View Auth requests, responses, and example code](https://plaid.com/docs/api/products/auth/)

**[View Auth API](https://plaid.com/docs/api/products/auth/)**

#### [Quickstart](https://plaid.com/docs/quickstart/)

[Learn about Plaid's key concepts and run starter code](https://plaid.com/docs/quickstart/)

**[Get started](https://plaid.com/docs/quickstart/)**

Auth allows you to request a user's checking or savings account information, making it easy for you to initiate credits or debits. For example, your app might allow users to accumulate a credit balance that they can cash out to a bank account, or it might allow users to pay you using their bank account information. With Auth, a user can provide this information in a frictionless way, simply by authenticating into their bank account.

For an all-in-one solution that includes payment processor functionality, see [Transfer](https://plaid.com/docs/transfer/) (US-only). Otherwise, Auth is used with a third-party payment processor to move money: either a [Plaid Partner](https://plaid.com/docs/auth/partnerships/) or another third party. For more information, see [Using a Payments Service](https://plaid.com/docs/auth/#using-a-payments-service).

Note that Auth can only be used with checking or savings accounts. Auth cannot be used with other depository accounts, such as money market accounts. Credit-type accounts, including credit cards, cannot receive payments directly via electronic interbank transfers, and Auth data cannot be used to set up credit card payments.

**Prefer to learn by watching? Get an overview of how Auth works in just 3 minutes!**

**Enhancing Auth with related products**

Auth is commonly used in combination with other Plaid APIs that reduce risk and support compliance.

For more information, see [Balance](https://plaid.com/docs/balance/) (to verify accounts have sufficient funds), [Signal](https://plaid.com/docs/signal/) (to calculate the risk of ACH returns with ML-powered analysis), and [Identity](https://plaid.com/docs/identity/) (to verify ownership information on the account).

For account funding use cases, see [Identity Verification](https://plaid.com/docs/identity-verification/) for an end-to-end KYC compliance solution with optional AML capabilities.

#### [Auth integration process](https://plaid.com/docs/auth/" \l "auth-integration-process)

Below is a high level overview of the Auth integration process. For a more detailed walkthrough, see [Add auth to your app](https://plaid.com/docs/auth/add-to-app/) or (if applicable) the docs for the specific [partner](https://plaid.com/docs/auth/partnerships/) you are using.

1. Create a Link token by calling [**/link/token/create**](https://plaid.com/docs/api/tokens/#linktokencreate) with auth in the products parameter. You may also set optional parameters here to extend coverage to more banks. For details, see [Additional auth coverage](https://plaid.com/docs/auth/coverage/). If you want to limit the user to linking only a single account, you can do so via a Link customization. For details, see [Configuring Link with Auth](https://plaid.com/docs/auth/#configuring-link-with-auth).
2. Initialize Link with the Link token from the previous step. Once the user has successfully completed the Link flow, exchange the public\_token for an access\_token. For more details, see [Link](https://plaid.com/docs/link/).
3. If using a Plaid partner for payment processing, ensure the partner is enabled on your [Plaid Dashboard](https://dashboard.plaid.com/developers/integrations), then call [**/processor/token/create**](https://plaid.com/docs/api/processors/#processortokencreate) or [**/processor/stripe/bank\_account\_token/create**](https://plaid.com/docs/api/processors/#processorstripebank_account_tokencreate) to obtain a token that you will provide to the partner to enable funds transfers. For more detailed instructions, including a full walkthrough, see [Auth payment partners](https://plaid.com/docs/auth/partnerships/).
4. If not using a Plaid partner, call [**/auth/get**](https://plaid.com/docs/api/products/auth/#authget) to obtain the account and routing number, then provide these fields to your payment processing system.
5. (Optional) To reduce fraud and complement know-your-customer processes, call [**/identity/get**](https://plaid.com/docs/api/products/identity/#identityget) to verify that the identity information held by the bank matches the information that the user has provided to you. For more details, see [Identity](https://plaid.com/docs/identity/).
6. (Optional) If you plan to transfer funds from the linked bank account, check the account's balance beforehand by calling [**/accounts/balance/get**](https://plaid.com/docs/api/products/balance/#accountsbalanceget) to avoid risk of an overdraft. For more details, see [Balance](https://plaid.com/docs/balance/).

#### [Using a payments service](https://plaid.com/docs/auth/" \l "using-a-payments-service)

Looking for bank-to-bank transfer capabilities and don't have a payment processor yet? Check out [Transfer](https://plaid.com/docs/transfer/) (US only) for a money movement solution with built-in payment processing capabilities.

When using Auth, you will send Auth data to a payments service to initiate an interbank transfer; Plaid does not function as the payment processor. While Plaid is processor-agnostic and allows you to work with any partner you want, one easy way to make transfers is to work with a payments platform that partners with Plaid, such as Dwolla or Stripe. Working with these partners, you will not call the [**/auth/get**](https://plaid.com/docs/api/products/auth/#authget) endpoint, so you will not obtain a user's bank account information. Instead, you will call [**/processor/token/create**](https://plaid.com/docs/api/processors/#processortokencreate) or [**/processor/stripe/bank\_account\_token/create**](https://plaid.com/docs/api/processors/#processorstripebank_account_tokencreate) to obtain a Plaid token that you will provide to the partner and that allows them to make these Plaid API calls as needed. For detailed instructions on how to set up Auth with a Plaid partner, as well as a list of supported funds transfer partners, see [Auth Partnerships](https://plaid.com/docs/auth/partnerships/).

If you choose to use a payments provider who is not a Plaid partner, you will need to obtain bank account numbers and codes using [**/auth/get**](https://plaid.com/docs/api/products/auth/#authget). Given the sensitive nature of this information, we recommend that you consult the [Open Finance Data Security Standard](https://ofdss.org/) for security best practices. We also recommend that you do not store account numbers. You can integrate with one of Plaid's [Data Security partners](https://plaid.com/partner-directory/) to process and share tokenized bank account numbers instead of raw bank account numbers. Contact your Account Manager to learn more about these partnerships.

#### [Covering more institutions](https://plaid.com/docs/auth/" \l "covering-more-institutions)

When used in its default configuration, Auth provides access to approximately 90% of financial institution accounts in supported countries. In the US, Plaid also offers additional flows you can implement to cover more institutions with Auth. For more information, see [Full Auth coverage](https://plaid.com/docs/auth/coverage/).

#### [Configuring Link with Auth](https://plaid.com/docs/auth/" \l "configuring-link-with-auth)

By default, only checking and savings accounts will appear when using Auth, and institutions that do not support these accounts will not appear in the Institution Select pane.

When using Auth, you will typically only need access to the one specific account that the end user wants to use to fund the payment, rather than all checking and savings accounts they may have at the same institution. Because of this, it is recommended to use the [Account Select](https://plaid.com/docs/link/customization/#account-select) Link customization when configuring Link for use with Auth, to limit unnecessary access to user accounts. You can enable Account Select from the [Dashboard](https://dashboard.plaid.com/link).

#### [Auth details by country](https://plaid.com/docs/auth/" \l "auth-details-by-country)

Just as there are multiple systems for bank codes and account numbers, there are also multiple systems for moving money from bank to bank. These systems are known as interbank transfer systems. The interbank transfer system used for an account depends on which country it is located in. Depending on the country, a bank account may participate in multiple interbank transfer systems and thus have multiple bank codes. For example, UK bank accounts participate in both SEPA and BACS payments. By calling [**/auth/get**](https://plaid.com/docs/api/products/auth/#authget), you will receive all account numbers and bank codes associated with an Item.

##### [United States (ACH)](https://plaid.com/docs/auth/" \l "united-states-ach)

In the United States, the bank code is known as a routing number, and the interbank transfer system is the Automated Clearing House (ACH) system. Because the ACH system can take several days to move money and can only be used within the United States, banks also have a second routing number, known as the wire transfer routing number, used to accept wire transfers. Wire transfers can be used to receive international payments and are faster than ACH transfers, but often involve a fee.

For a detailed, comprehensive guide to ACH transfers and payments, see Plaid's [Modern Guide to ACH](https://plaid.com/solutions/ACH-guide/).

Example Auth data for US bank account

**Copy**

1"numbers": {

2 "ach": [

3 {

4 "account": "1111222233330000",

5 "account\_id": "bWG9l1DNpdc8zDk3wBm9iMkW3p1mVWCVKVmJ6",

6 "routing": "011401533",

7 "wire\_routing": "021000021"

8 }

9 ],

10 "bacs": [],

11 "eft": [],

12 "international": []

13}

##### [Canada (EFT)](https://plaid.com/docs/auth/" \l "canada-eft)

In Canada, the bank code is also known as a routing number, but it is in a different format than US routing numbers and broken into two pieces: the transit number (also known as the branch number), followed by the institution number. The interbank transfer system in Canada is the Electronic Funds Transfer (EFT) system.

Example Auth data for Canadian bank account

**Copy**

1"numbers": {

2 "ach": [],

3 "bacs": [],

4 "eft": [

5 {

6 "account": "111122220000",

7 "account\_id": "qVZ3Bwbo5wFmoVneZxMksBvN6vDad6idkndAB",

8 "branch": "01533",

9 "institution": "114"

10 }

11 ],

12 "international": []

13}

##### [Europe (SEPA transfers)](https://plaid.com/docs/auth/" \l "europe-sepa-transfers)

In the European Economic Area member states (which includes Euro zone nations, as well as the United Kingdom), the bank code is called a Bank Identifier Code (BIC), also known as a SWIFT code. Each bank account has a number in a standard format, known as the International Bank Account Number (IBAN), which is used along with the BIC for funds transfers. Many bank accounts also have internal, non-IBAN account numbers that cannot be used for funds transfers. The funds transfer system is known as the Shared European Payment Area (SEPA), and it supports three types of funds transfers: SEPA credit transfer, SEPA instant credit transfer, and SEPA direct debit.

Example Auth data for European bank account

**Copy**

1"numbers": {

2 "ach": [],

3 "bacs": [],

4 "eft": [],

5 "international": [

6 {

7 "account\_id": "blgvvBlXw3cq5GMPwqB6s6q4dLKB9WcVqGDGo",

8 "bic": "IE64BOFI90583812345678",

9 "iban": "IE64BOFI90583812345678"

10 }

11 ]

12}

##### [United Kingdom (BACS)](https://plaid.com/docs/auth/" \l "united-kingdom-bacs)

The UK uses the SEPA system as well as its own system, known as the Bankers Automated Clearing System (BACS), in which the bank code is known as a sort code. Similar to the ACH system, payments within the BACS system cannot be made outside the UK and take several days to process. BACS payments are typically used for recurring direct debit payments, such as payroll. UK bank accounts will typically have both a BACS sort code and an IBAN and support both BACS transfers and SEPA transfers.

Example Auth data for UK bank account

**Copy**

1"numbers": {

2 "ach": [],

3 "bacs": [

4 {

5 "account": "80000000",

6 "account\_id": "blgvvBlXw3cq5GMPwqB6s6q4dLKB9WcVqGDGo",

7 "sort\_code": "040004"

8 }

9 ],

10 "eft": [],

11 "international": [

12 {

13 "account\_id": "blgvvBlXw3cq5GMPwqB6s6q4dLKB9WcVqGDGo",

14 "bic": "MONZGB21XXX",

15 "iban": "GB23MONZ04000480000000"

16 }

17 ]

18}

For UK-based implementations, also consider the [Payment Initiation API](https://plaid.com/docs/payment-initiation/), which allows end-to-end payments directly, without having to integrate an additional payment processor.

#### [Sample app code](https://plaid.com/docs/auth/" \l "sample-app-code)

For a real-life example of an app that incorporates Auth, see the Node-based [Plaid Pattern Account Funding](https://github.com/plaid/pattern-account-funding) sample app. Pattern Account Funding is a sample account funding app that fetches Auth data in order to set up funds transfers. The Auth code can be found in [items.js](https://github.com/plaid/pattern-account-funding/blob/master/server/routes/items.js#L81-L135).

#### [Testing Auth](https://plaid.com/docs/auth/" \l "testing-auth)

Plaid provides a [GitHub repo](https://github.com/plaid/sandbox-custom-users) with test data for testing Auth in Sandbox, helping you test configuration options beyond those offered by the default Sandbox user. For more information on configuring custom Sandbox data, see [Configuring the custom user account](https://plaid.com/docs/sandbox/user-custom/#configuring-the-custom-user-account).

For details on testing Auth with more complex Auth flows such as micro-deposit-based Auth, first familiarize yourself with [Adding Institution Coverage](https://plaid.com/docs/auth/coverage/), then see [Test in Sandbox](https://plaid.com/docs/auth/coverage/testing/).

# Add Auth to your app

#### Use Auth to connect user bank accounts

In this guide, we'll demonstrate how to add [Auth](https://plaid.com/docs/api/products/auth/) to your app so that you can connect to your users' bank accounts and obtain the information needed to set up funds transfers.

If you're already familiar with using Plaid and are set up to make calls to the Plaid API, see [Getting Auth data](https://plaid.com/docs/auth/add-to-app/#getting-auth-data). If you're interested in using a Plaid partner, such as Stripe or Dwolla, to process payments, see [Moving funds with a payment partner](https://plaid.com/docs/auth/add-to-app/#moving-funds-with-a-payment-partner).

Prefer to learn by watching? A [video guide](https://youtu.be/FlZ5nzlIq74) is available for this topic.

#### [Get Plaid API keys and complete application and company profile](https://plaid.com/docs/auth/add-to-app/" \l "get-plaid-api-keys-and-complete-application-and-company-profile)

If you don't already have one, you'll need to [create a Plaid developer account](https://dashboard.plaid.com/signup). After creating your account, you can find your [API keys](https://dashboard.plaid.com/developers/keys) under the Team Settings menu on the Plaid Dashboard.

You will also need to complete your [application profile](https://dashboard.plaid.com/settings/company/app-branding) and [company profile](https://dashboard.plaid.com/settings/company/profile) in the Dashboard. The information in your profile will be shared with users of your application when they manage their connection on the [Plaid Portal](https://my.plaid.com/). Your application profile and company profile must be completed before connecting to certain institutions in Production.

#### [Install and initialize Plaid libraries](https://plaid.com/docs/auth/add-to-app/" \l "install-and-initialize-plaid-libraries)

You can use our official server-side client libraries to connect to the Plaid API from your application:

Select group for content switcher

Current librariesLegacy libraries

Terminal

**Curl**

**Select Language**

**Copy**

1## no setup

After you've installed Plaid's client libraries, you can initialize them by passing in your client\_id, secret, and the environment you wish to connect to (Sandbox, Development, or Production). This will make sure the client libraries pass along your client\_id and secret with each request, and you won't need to explicitly include them in any other calls.

In the code samples below, you will need to replace PLAID\_CLIENT\_ID and PLAID\_SECRET with your own keys, which you can obtain from the [Dashboard](https://dashboard.plaid.com/developers/keys). These code samples also demonstrate starting up a server commonly used in each framework (such as Express or Flask).

Select group for content switcher

Current librariesLegacy libraries

**Curl**

**Select Language**

**Copy**

1## no setup

#### [Create an Item using Link](https://plaid.com/docs/auth/add-to-app/" \l "create-an-item-using-link)

These instructions cover Link for web applications. For instructions on using Link in mobile apps, see the [Link documentation](https://plaid.com/docs/link/).

Plaid Link is the client-side component that your users interact with to securely connect their bank accounts to your app. An [Item](https://plaid.com/docs/quickstart/glossary/#item) is created when a user successfully logs into their financial institution using Link. An Item represents a single login at a financial institution. Items do not represent individual bank accounts, although all bank accounts are associated with an Item. For example, if a user has a single login at a bank that allows them to access both a checking account and a savings account, only a single Item would be associated with both of the accounts.

When using Auth, you will typically only need access to the specific bank account that will be used to transfer funds, rather than all of the accounts a user may have at an institution. Because of this, it is recommended that you configure Link with [Account Select](https://plaid.com/docs/link/customization/#account-select) when using Auth. Configuring Link with Account Select will limit unnecessary access to user accounts. You can [enable Account Select from the Dashboard](https://dashboard.plaid.com/link/account-select).

To create an Item, you'll first need to create a [Link token](https://plaid.com/docs/quickstart/glossary/#link-token) on your application server. You can create a Link token by calling the [**/link/token/create**](https://plaid.com/docs/api/tokens/#linktokencreate) endpoint. Then, on the client-side of your application, you'll initialize Link with using the Link token you created.

The code samples below demonstrate how to create a Link token and how to initialize Link using the token.

##### [Create a Link token](https://plaid.com/docs/auth/add-to-app/" \l "create-a-link-token)

Select group for content switcher

Current librariesLegacy libraries

**Curl**

**Select Language**

**Copy**

1curl -X POST https://sandbox.plaid.com/link/token/create \

2-H 'Content-Type: application/json' \

3-d '{

4 "client\_id": "${PLAID\_CLIENT\_ID}",

5 "secret": "${PLAID\_SECRET}",

6 "client\_name": "Plaid Test App",

7 "user": { "client\_user\_id": "${UNIQUE\_USER\_ID}" },

8 "products": ["${PRODUCT}"],

9 "country\_codes": ["US"],

10 "language": "en",

11 "webhook": "https://webhook.example.com",

12 "redirect\_uri": "https://domainname.com/oauth-page.html",

13}'

##### [Initialize Link with the Link token](https://plaid.com/docs/auth/add-to-app/" \l "initialize-link-with-the-link-token)

app.js

**Copy**

1const handler = Plaid.create({

2 token: (await $.post('/create\_link\_token')).link\_token,

3 onSuccess: (public\_token, metadata) => {

4 // Upon successful linking of a bank account,

5 // Link will return a public token.

6 // Exchange the public token for an access token

7 // to make calls to the Plaid API.

8 $.post('/exchange\_public\_token', {

9 public\_token: public\_token,

10 });

11 },

12 onLoad: () => {},

13 onExit: (err, metadata) => {

14 // Optionally capture when your user exited the Link flow.

15 // Storing this information can be helpful for support.

16 },

17 onEvent: (eventName, metadata) => {

18 // Optionally capture Link flow events, streamed through

19 // this callback as your users connect an Item to Plaid.

20 },

21});

22

23handler.open();

#### [Obtain an access token](https://plaid.com/docs/auth/add-to-app/" \l "obtain-an-access-token)

When a user successfully links an Item via Link, the [**onSuccess**](https://plaid.com/docs/link/web/#onsuccess) callback will be called. The onSuccess callback returns a [public token](https://plaid.com/docs/quickstart/glossary/#public-token). On your application server, exchange the public token for an [access token](https://plaid.com/docs/quickstart/glossary/#access-token) and an [Item ID](https://plaid.com/docs/quickstart/glossary/#item-id) by calling /item/public\_token/exchange/. The access token will allow you to make authenticated calls to the Plaid API.

Store the access token and Item ID in a secure datastore, as they’re used to access Item data and identify webhooks, respectively. The access token will remain valid unless you actively expire it via rotation or remove it by calling [**/item/remove**](https://plaid.com/docs/api/items/#itemremove) on the corresponding Item. For security purposes, never store the access token in client-side code. The public token is a one-time use token with a lifetime of 30 minutes, so there is no need to store it.

##### [Exchange public token for an access token](https://plaid.com/docs/auth/add-to-app/" \l "exchange-public-token-for-an-access-token)

Select group for content switcher

Current librariesLegacy libraries

**Curl**

**Select Language**

**Copy**

1curl -X POST https://sandbox.plaid.com/item/public\_token/exchange \

2-H 'Content-Type: application/json' \

3-d '{

4 "client\_id": "${PLAID\_CLIENT\_ID}",

5 "secret": "${PLAID\_SECRET}",

6 "public\_token": "public-sandbox-12345678-abcd-1234-abcd-1234567890ab"

7}'

#### [Getting Auth data](https://plaid.com/docs/auth/add-to-app/" \l "getting-auth-data)

Now that you have an access token, you can begin making authenticated calls to the Plaid API. If you are processing payments yourself, the [**/auth/get**](https://plaid.com/docs/api/products/auth/#authget) endpoint to retrieve the bank account and bank identification numbers (such as routing numbers, for US accounts) associated with an Item's accounts. You can then supply these to your payment system. If you are using a Plaid partner to move funds, you will not need to call [**/auth/get**](https://plaid.com/docs/api/products/auth/#authget). Instead, see [Moving funds with a payment partner](https://plaid.com/docs/auth/add-to-app/#moving-funds-with-a-payment-partner).

Select group for content switcher

Current librariesLegacy libraries

**Curl**

**Select Language**

**Copy**

1curl -X POST https://sandbox.plaid.com/auth/get \

2-H 'Content-Type: application/json' \

3-d '{

4 "client\_id": "${PLAID\_CLIENT\_ID}",

5 "secret": "${PLAID\_SECRET}",

6 "access\_token": String

7}'

Example response data is below. Note that this is test account data; real accounts would not include all four sets of numbers.

/auth/get sample response

**Copy**

1{

2 "accounts": [

3 {

4 "account\_id": "vzeNDwK7KQIm4yEog683uElbp9GRLEFXGK98D",

5 "balances": {

6 "available": 100,

7 "current": 110,

8 "limit": null,

9 "iso\_currency\_code": "USD",

10 "unofficial\_currency\_code": null

11 },

12 "mask": "9606",

13 "name": "Plaid Checking",

14 "official\_name": "Plaid Gold Checking",

15 "subtype": "checking",

16 "type": "depository"

17 }

18 ],

19 "numbers": {

20 "ach": [

21 {

22 "account": "9900009606",

23 "account\_id": "vzeNDwK7KQIm4yEog683uElbp9GRLEFXGK98D",

24 "routing": "011401533",

25 "wire\_routing": "021000021"

26 }

27 ],

28 "eft": [

29 {

30 "account": "111122223333",

31 "account\_id": "vzeNDwK7KQIm4yEog683uElbp9GRLEFXGK98D",

32 "institution": "021",

33 "branch": "01140"

34 }

35 ],

36 "international": [

37 {

38 "account\_id": "vzeNDwK7KQIm4yEog683uElbp9GRLEFXGK98D",

39 "bic": "NWBKGB21",

40 "iban": "GB29NWBK60161331926819"

41 }

42 ],

43 "bacs": [

44 {

45 "account": "31926819",

46 "account\_id": "vzeNDwK7KQIm4yEog683uElbp9GRLEFXGK98D",

47 "sort\_code": "601613"

48 }

49 ]

50 },

51 "item": {

52 "available\_products": [

53 "balance",

54 "identity",

55 "payment\_initiation",

56 "transactions"

57 ],

58 "billed\_products": ["assets", "auth"],

59 "consent\_expiration\_time": null,

60 "error": null,

61 "institution\_id": "ins\_117650",

62 "item\_id": "DWVAAPWq4RHGlEaNyGKRTAnPLaEmo8Cvq7na6",

63 "webhook": "https://www.genericwebhookurl.com/webhook"

64 },

65 "request\_id": "m8MDnv9okwxFNBV"

66}

#### [Moving funds with a payment partner](https://plaid.com/docs/auth/add-to-app/" \l "moving-funds-with-a-payment-partner)

You can move money via ACH transfer by pairing Auth with one of Plaid's payment partners. When using a partner to move money, the partner's payments service will initiate the transfer; Plaid does not function as the payment processor. For the full list of payments platforms who have partnered with Plaid to provide ACH money movement, see [Auth Partnerships](https://plaid.com/docs/auth/partnerships/).

To move money using a Plaid partner, first create an Item using Link and obtain an access token as described above. Then, instead of calling Plaid's Auth endpoints, call one of Plaid's [processor token endpoints](https://plaid.com/docs/api/processors/) to create a processor token. You can then send this processor token to one of Plaid's partners by using endpoints that are specific to the payment platform. Refer to the partner's technical documentation for more information. Using a partner to transfer funds gives you access to payment functionality while freeing you from having to securely store sensitive bank account information.

The sample code below demonstrates a call to [**/processor/token/create**](https://plaid.com/docs/api/processors/#processortokencreate) using Dwolla as the payment processor.

Select group for content switcher

Current librariesLegacy libraries

Sample /processor/token/create call

**Curl**

**Select Language**

**Copy**

1# Create a processor token for a specific account id.

2curl \

3 -H 'Content-Type: application/json' \

4 -d '{

5 "client\_id": "${PLAID\_CLIENT\_ID}",

6 "secret": "${PLAID\_SECRET}",

7 "access\_token": "${ACCESS\_TOKEN}",

8 "account\_id": "${ACCOUNT\_ID}",

9 "processor": "dwolla"

10 }' \

11 -X POST \

12 https://sandbox.plaid.com/processor/token/create

# Instant Auth, Instant Match, and Instant Micro-Deposits

#### Learn how to authenticate your users instantly

#### [Instant Auth](https://plaid.com/docs/auth/coverage/instant/" \l "instant-auth)

Instant Auth supports more than 3,800 financial institutions with credential-based login. Instant Auth is the default Auth flow and does not require extra configuration steps if Auth is already configured in your app. For clarity and completeness, the section below explains how to configure Instant Auth.

# Introduction to Balance

#### Retrieve real-time balance information to prevent NSFs and ACH returns

#### [API Reference](https://plaid.com/docs/api/products/balance/)

[View Balance requests, responses, and example code](https://plaid.com/docs/api/products/balance/)

**[View Balance API](https://plaid.com/docs/api/products/balance/)**

#### [Quickstart](https://plaid.com/docs/quickstart/)

[Learn about Plaid's key concepts and run starter code](https://plaid.com/docs/quickstart/)

**[Get started](https://plaid.com/docs/quickstart/)**

#### [Overview](https://plaid.com/docs/balance/" \l "overview)

Balance is Plaid's product for receiving real-time Balance information via [**/accounts/balance/get**](https://plaid.com/docs/api/products/balance/#accountsbalanceget). This real-time Balance data can be helpful when checking to see if an account has sufficient funds before using it as a funding source for a money transfer. Balance is available for use exclusively in combination with other Plaid products, such as [Auth](https://plaid.com/docs/auth/) for money movement or account funding use cases or [Transactions](https://plaid.com/docs/transactions/) for personal finance use cases.

**Prefer to learn by watching? Get an overview of how Balance works in just 3 minutes!**

#### [Balance data](https://plaid.com/docs/balance/" \l "balance-data)

Sample balance data

**Copy**

1"balances": {

2 "available": 100,

3 "current": 110,

4 "iso\_currency\_code": "USD",

5 "limit": null,

6 "unofficial\_currency\_code": null

7}

#### [Cached and realtime balance](https://plaid.com/docs/balance/" \l "cached-and-realtime-balance)

You can retrieve cached balance data for any Item (except for Items with manual connections, such as those using Same-Day Microdeposits) by calling the free-to-use [**/accounts/get**](https://plaid.com/docs/api/accounts/#accountsget) endpoint. Many Plaid products other than Balance also provide cached balance data as part of the accounts object. Because this data is cached, it is not suitable for situations where you need real-time balance information. The frequency with which cached balance data is updated varies depending on the product. An Item with Transactions, for example, may update cached balances once a day or more, while an Item with only Auth may update every 30 days or even less frequently.

For retrieving real-time balance for payments use cases, only [**/accounts/balance/get**](https://plaid.com/docs/api/products/balance/#accountsbalanceget) should be used. The exception is immediately after an Item has been linked, when the balance information is guaranteed to be recent. Note that because Balance always retrieves fresh data from the institution, latency will be higher when calling [**/accounts/balance/get**](https://plaid.com/docs/api/products/balance/#accountsbalanceget) than when calling other Plaid endpoints. While [**/accounts/balance/get**](https://plaid.com/docs/api/products/balance/#accountsbalanceget) typically takes about 10 seconds or less, the latency will vary by institution, and in some cases it may take 30 seconds or more to retrieve balance information.

If you want to evaluate the risk of an ACH return and need lower latency than [**/accounts/balance/get**](https://plaid.com/docs/api/products/balance/#accountsbalanceget), try [Signal](https://plaid.com/docs/signal/). Signal uses machine learning instead of real-time balance checks to evaluate the risk of an ACH return, and can return results in approximately one second or less.

#### [Current and available balance](https://plaid.com/docs/balance/" \l "current-and-available-balance)

Balance typically returns both current and available balance information. For fraud detection and insufficient funds (NSF) prevention use cases, available balance should be used, as it represents predicted balance net of any pending transactions, while current balance does not take pending transactions into account.

Available balance indicates balance that is available to spend, which is not the same as the overall value of the account. For example, for credit accounts, it indicates the amount that can be spent without hitting the credit limit (net of any pending transactions), while in investment accounts, it indicates the total available cash.

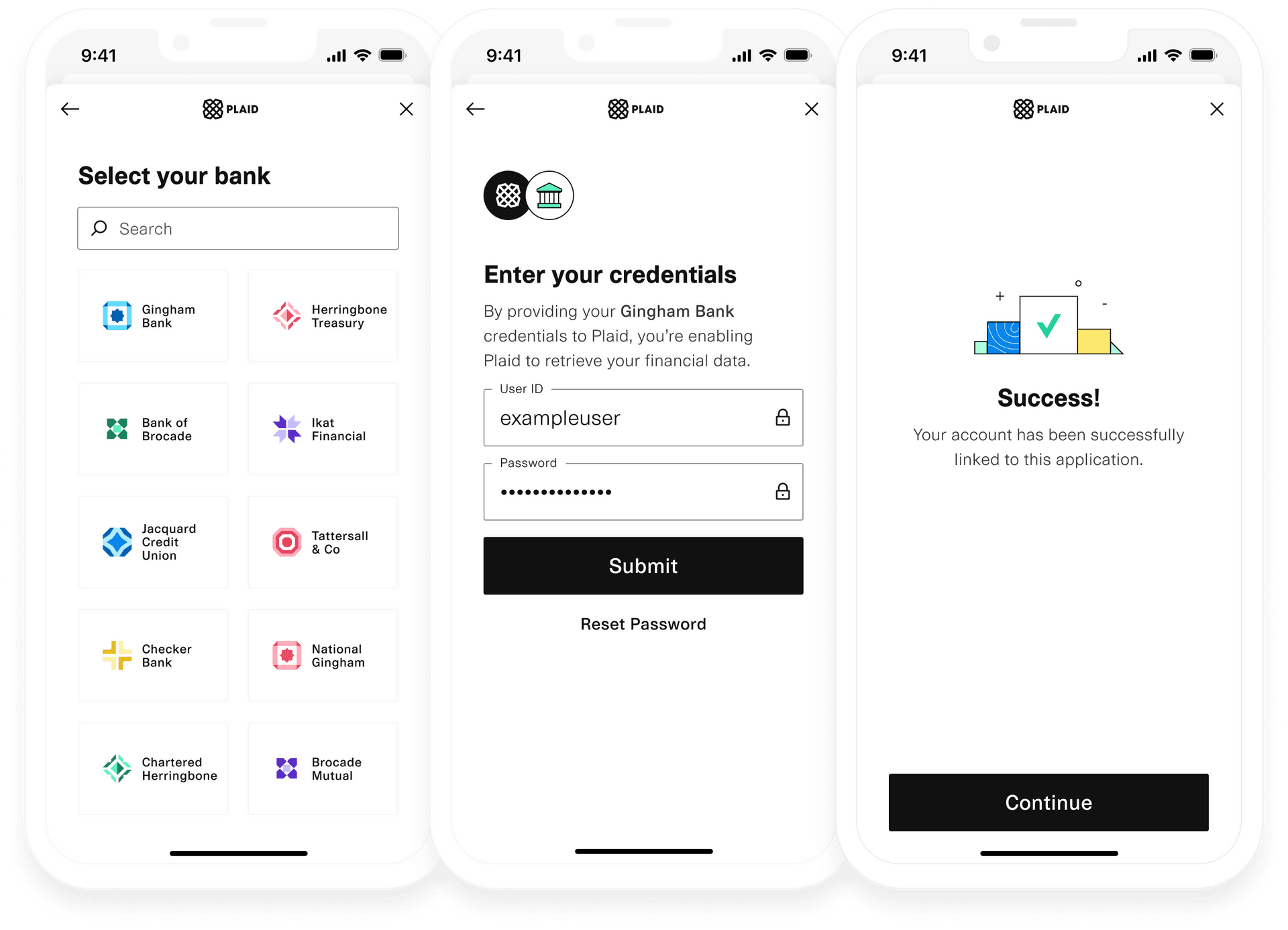
In some cases, a financial institution may not return available balance information. If that happens, you can calculate available balance by starting with the current balance, then using the [Transactions](https://plaid.com/docs/transactions/) product to detect any pending transactions and adjusting the balance accordingly.

##### [Typical balance fill rates by field](https://plaid.com/docs/balance/" \l "typical-balance-fill-rates-by-field)

| **FIELD** | **TYPICAL FILL RATE** |
| --- | --- |
| Current balance | 99% |
| Available balance | 91% |

#### [Sample app code](https://plaid.com/docs/balance/" \l "sample-app-code)

For a real-life example of an app that incorporates Balance, see the Node-based [Plaid Pattern Account Funding](https://github.com/plaid/pattern-account-funding) sample app. Pattern Account Funding is a sample account funding app that fetches Balance data to check for sufficient funds before a funds transfer. The Balance code can be found in [items.js](https://github.com/plaid/pattern-account-funding/blob/master/server/routes/items.js#L211-L234).



You can try out the Instant Match flow in [Link Demo](https://plaid.com/demo/). See more details in our [testing guide](https://plaid.com/docs/auth/coverage/testing/#testing-instant-match).

##### [Configure & Create a link\_token](https://plaid.com/docs/auth/coverage/instant/" \l "configure--create-a-link_token)

Create a link\_token with the following parameters:

* products array containing auth – If you are using only auth and no other products, auth must be specified in the Products array. Other products (such as identity) may be specified as well. If you are using multiple products, auth is not required to be specified in the products array, but including it is recommended for the best user experience.

Select group for content switcher

Current librariesLegacy libraries

**Curl**

**Select Language**

**Copy**

1curl -X POST https://sandbox.plaid.com/link/token/create \

2-H 'Content-Type: application/json' \

3-d '{

4 "client\_id": "${PLAID\_CLIENT\_ID}",

5 "secret": "${PLAID\_SECRET}",

6 "client\_name": "Plaid Test App",

7 "user": { "client\_user\_id": "${UNIQUE\_USER\_ID}" },

8 "products": ["${PRODUCT}"],

9 "country\_codes": ["US"],

10 "language": "en",

11 "webhook": "https://webhook.example.com",

12 "redirect\_uri": "https://domainname.com/oauth-page.html",

13}'

##### [Initialize Link with a link\_token](https://plaid.com/docs/auth/coverage/instant/" \l "initialize-link-with-a-link_token)

After creating a link\_token for the auth product, use it to initialize Plaid Link.

When the user inputs their username and password for the financial institution, the onSuccess() callback function will return a public\_token.

App.js

**Curl**

**Select Language**

**Copy**

1Plaid.create({

2 // Fetch a link\_token configured for 'auth' from your app server

3 token: (await $.post('/create\_link\_token')).link\_token,

4 onSuccess: (public\_token, metadata) => {

5 // Send the public\_token and accounts to your app server

6 $.post('/exchange\_public\_token', {

7 publicToken: public\_token,

8 accounts: metadata.accounts,

9 });

10 },

11});

##### [Exchange the public\_token and fetch Auth data](https://plaid.com/docs/auth/coverage/instant/" \l "exchange-the-public_token-and-fetch-auth-data)

In your own backend server, call the [**/item/public\_token/exchange**](https://plaid.com/docs/api/tokens/#itempublic_tokenexchange) endpoint with the Link public\_token received in the onSuccess callback to obtain an access\_token. Persist the returned access\_token and item\_id in your database in relation to the user. You will use the access\_token when making requests to the [**/auth/get**](https://plaid.com/docs/api/products/auth/#authget) endpoint.

Select group for content switcher

Current librariesLegacy libraries

Exchange token and fetch Auth data

**Curl**

**Select Language**

**Copy**

1curl -X POST https://sandbox.plaid.com/item/public\_token/exchange \

2 -H 'Content-Type: application/json' \

3 -d '{

4 "client\_id": "${PLAID\_CLIENT\_ID}",

5 "secret": "${PLAID\_SECRET}",

6 "public\_token": "public-sandbox-b0e2c4ee-a763-4df5-bfe9-46a46bce993d"

7 }'

8

9curl -X POST https://sandbox.plaid.com/auth/get \

10 -H 'Content-Type: application/json' \

11 -d '{

12 "client\_id": "${PLAID\_CLIENT\_ID}",

13 "secret": "${PLAID\_SECRET}",

14 "access\_token": "${ACCESS\_TOKEN}"

15 }'

Check out the [**/auth/get**](https://plaid.com/docs/api/products/auth/#authget) API reference documentation to see the full Auth request and response schema.

Auth response

**Copy**

1{

2 "numbers": {

3 "ach": [

4 {

5 "account\_id": "vzeNDwK7KQIm4yEog683uElbp9GRLEFXGK98D",

6 "account": "9900009606",

7 "routing": "011401533",

8 "wire\_routing": "021000021"

9 }

10 ],

11 "eft": [],

12 "international": [],

13 "bacs": []

14 },

15 "accounts": [{ Object }],

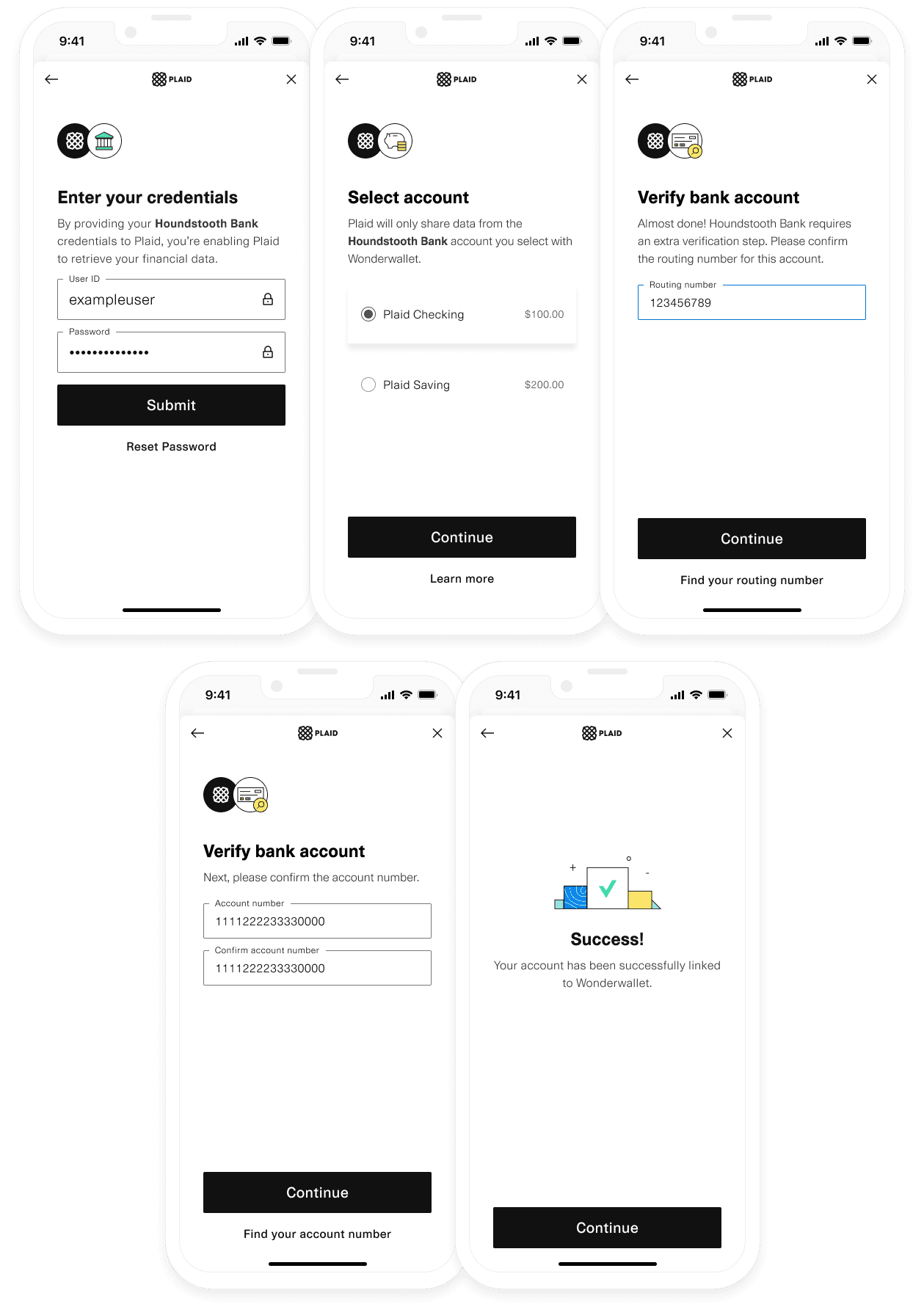
16 "item": { Object },

17 "request\_id": "m8MDnv9okwxFNBV"

18}

#### [Instant Match](https://plaid.com/docs/auth/coverage/instant/" \l "instant-match)

Instant Match is available for approximately 1,500 U.S. additional financial institutions where Instant Auth is not available. Instant Match is enabled automatically for Auth customers and is automatically provided at supported institutions as a fallback experience when Instant Auth is not available. When using Instant Match, Plaid Link will prompt your user to enter their account number and routing number for a depository account. Plaid will then verify the last four digits of the user-provided account number against the account mask retrieved from the financial institution.



You can try out the Instant Match flow in [Link Demo](https://plaid.com/demo/). See more details in our [testing guide](https://plaid.com/docs/auth/coverage/testing/#testing-instant-match).

When using the Instant Match flow, the user can verify only a single account. Even if the Account Select properties allow selecting all or multiple accounts, the ability to select multiple depository accounts for Auth will be disabled in Link if the institution is using the Instant Match flow.

##### [Configuring in Link](https://plaid.com/docs/auth/coverage/instant/" \l "configuring-in-link)

Instant Match will be enabled automatically if you configure the link\_token with the following parameters:

* add "auth" to products array
* country\_codes set to ['US'] (adding any other countries to the array will disable Instant Match)

Optionally, you can disable Instant Match on a per-session basis via the [**/link/token/create**](https://plaid.com/docs/api/tokens/#linktokencreate) call, by setting "auth.instant\_match\_enabled": false in the request body. If you would like to disable Instant Match automatically for all Link sessions, contact your Account Manager or file a support ticket via the Dashboard.

Instant Match Configuration

**Curl**

**Select Language**

**Copy**

1curl -X POST https://sandbox.plaid.com/link/token/create

2-H 'Content-Type: application/json'

3-d '{

4 "client\_id": "${PLAID\_CLIENT\_ID}",

5 "secret": "${PLAID\_SECRET}",

6 "user": {"client\_user\_id": "${UNIQUE\_USER\_ID}"},

7 "client\_name": "Plaid App",

8 "products": ["auth"],

9 "country\_codes": ["US"],

10 "language": "en",

11 "webhook": "https://sample-web-hook.com"

12}'

##### [Handling Link events](https://plaid.com/docs/auth/coverage/instant/" \l "handling-link-events)

For a user who goes through the Instant Match flow, the TRANSITION\_VIEW (view\_name = NUMBERS) event will occur after SUBMIT\_CREDENTIALS, and in the onSuccess callback the verification\_status will be null because the user would have been verified instantly.

Sample Link events for Instant Match flow

**Copy**

1OPEN (view\_name = CONSENT)

2TRANSITION\_VIEW view\_name = SELECT\_INSTITUTION)

3SEARCH\_INSTITUTION

4SELECT\_INSTITUTION

5TRANSITION\_VIEW (view\_name = CREDENTIAL)

6SUBMIT\_CREDENTIALS

7TRANSITION\_VIEW (view\_name = LOADING)

8TRANSITION\_VIEW (view\_name = MFA, mfa\_type = code)

9SUBMIT\_MFA (mfa\_type = code)

10TRANSITION\_VIEW (view\_name = LOADING)

11TRANSITION\_VIEW (view\_name = SELECT\_ACCOUNT)

12TRANSITION\_VIEW (view\_name = NUMBERS)

13TRANSITION\_VIEW (view\_name = LOADING)

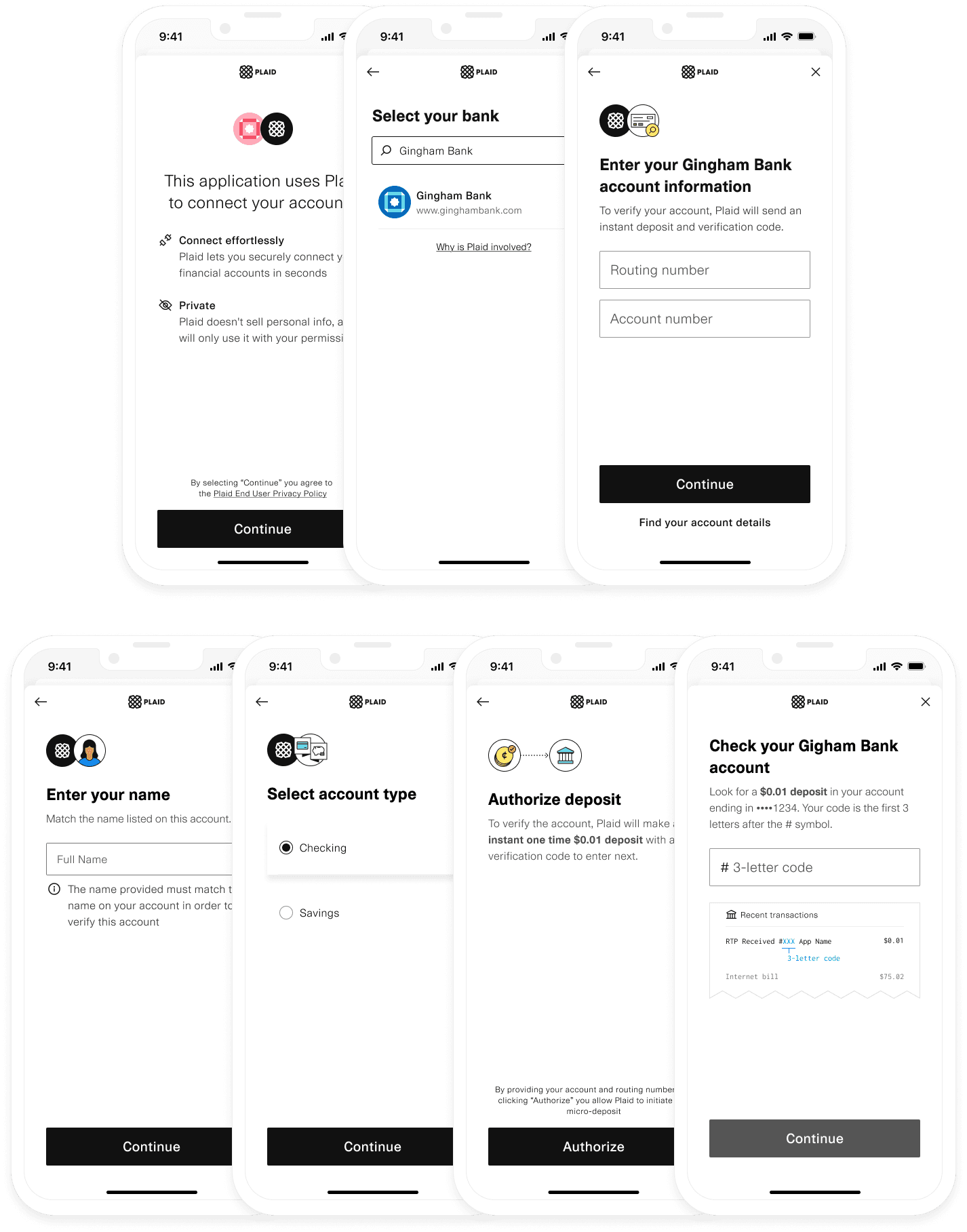
14TRANSITION\_VIEW (view\_name = CONNECTED)

15HANDOFF

16onSuccess (verification\_status: null)

#### [Instant Micro-deposits](https://plaid.com/docs/auth/coverage/instant/" \l "instant-micro-deposits)

Instant Micro-deposits is the Plaid product term for our ability to authenticate any bank account in the US that is supported by RTP or FedNow. For over 20 Plaid-supported banks, Instant Micro-deposits is the fastest and highest-converting form of Auth support available.



##### [Instant Micro-deposit flow](https://plaid.com/docs/auth/coverage/instant/" \l "instant-micro-deposit-flow)

1. Starting on a page in your app, the user clicks an action that opens Plaid Link.
2. Inside of Plaid Link, the user enters the micro-deposit initiation flow and provides their legal name, account and routing number.
3. Plaid sends a micro-deposit to the user's account that will post within 5 seconds, and directs the user to log into their bank account to obtain the code from the micro-deposit description.
4. The user enters the code from the micro-deposit description into Plaid Link.
5. Upon success, Link closes with a public\_token and a metadata account status of manually\_verified.

Plaid will not reverse the $0.01 micro-deposit credit.

When these steps are done, your user's Auth data is verified and ready to fetch.

##### [Configuring in Link](https://plaid.com/docs/auth/coverage/instant/" \l "configuring-in-link-1)

Instant Micro-deposits will be enabled automatically if you configure the link\_token with the following parameters:

* Set the products array to ["auth"]. While in most cases additional products can be added to existing Plaid Items, Items created for micro-deposit verification are an exception and cannot be used with any Plaid products other than Auth or Transfer.

Approximately 30% of Items verified by Instant micro-deposits can also be verified by [**/identity/match**](https://plaid.com/docs/api/products/identity/#identitymatch). For more details on using Identity Match with these Items, see [Identity Match](https://plaid.com/docs/identity/#identity-match).

* country\_codes set to ['US'] (adding any other countries to the array will disable Instant Micro-deposits)
* Add auth.instant\_microdeposits\_enabled: true to the [**/link/token/create**](https://plaid.com/docs/api/tokens/#linktokencreate) request body.

Beginning November 1, 2023, Instant Micro-deposits will be enabled by default for all eligible Auth sessions if auth.instant\_microdeposits\_enabled is not set. If your use case requires that you call [**/accounts/balance/get**](https://plaid.com/docs/api/products/balance/#accountsbalanceget) and cannot function if you cannot call that endpoint, you should disable Instant Micro-deposits.

Optionally, you can disable Instant Micro-deposits on a per-session basis via the [**/link/token/create**](https://plaid.com/docs/api/tokens/#linktokencreate) call, by setting "auth.instant\_microdeposits\_enabled": false in the request body. If you would like to disable Instant Micro-deposits automatically for all Link sessions, contact your Account Manager or file a support ticket via the Dashboard.

##### [Entering the Instant Micro-deposit flow](https://plaid.com/docs/auth/coverage/instant/" \l "entering-the-instant-micro-deposit-flow)

Your user will enter the Instant Micro-deposit flow in the following scenarios:

* The user selects an eligible institution that is not enabled for Instant Auth, Instant Match, or Automated Micro-deposits.
* The Link session has [Same Day Micro-deposits](https://plaid.com/docs/auth/coverage/same-day/) enabled and the user enters an eligible routing number during the Same Day Micro-deposits flow. In this case, the session will be "upgraded" to use Instant Micro-deposits rather than Same-Day Micro-deposits.

###### Instant Micro-deposit events

When a user goes through the Instant micro-deposits flow, the session will have the TRANSITION\_VIEW (view\_name = NUMBERS) event, and in the onSuccess callback the verification\_status will be manually\_verified.

##### [Testing the Instant Micro-deposit flow](https://plaid.com/docs/auth/coverage/instant/" \l "testing-the-instant-micro-deposit-flow)

For credentials that can be used to test Instant Micro-deposits in Sandbox, see [Auth testing flows](https://plaid.com/docs/auth/coverage/testing/#testing-instant-micro-deposits).

# Introduction to Balance

#### Retrieve real-time balance information to prevent NSFs and ACH returns

#### [API Reference](https://plaid.com/docs/api/products/balance/)

[View Balance requests, responses, and example code](https://plaid.com/docs/api/products/balance/)

**[View Balance API](https://plaid.com/docs/api/products/balance/)**

#### [Quickstart](https://plaid.com/docs/quickstart/)

[Learn about Plaid's key concepts and run starter code](https://plaid.com/docs/quickstart/)

**[Get started](https://plaid.com/docs/quickstart/)**

#### [Overview](https://plaid.com/docs/balance/" \l "overview)

Balance is Plaid's product for receiving real-time Balance information via [**/accounts/balance/get**](https://plaid.com/docs/api/products/balance/#accountsbalanceget). This real-time Balance data can be helpful when checking to see if an account has sufficient funds before using it as a funding source for a money transfer. Balance is available for use exclusively in combination with other Plaid products, such as [Auth](https://plaid.com/docs/auth/) for money movement or account funding use cases or [Transactions](https://plaid.com/docs/transactions/) for personal finance use cases.

**Prefer to learn by watching? Get an overview of how Balance works in just 3 minutes!**

#### [Balance data](https://plaid.com/docs/balance/" \l "balance-data)

Sample balance data

**Copy**

1"balances": {

2 "available": 100,

3 "current": 110,

4 "iso\_currency\_code": "USD",

5 "limit": null,

6 "unofficial\_currency\_code": null

7}

#### [Cached and realtime balance](https://plaid.com/docs/balance/" \l "cached-and-realtime-balance)

You can retrieve cached balance data for any Item (except for Items with manual connections, such as those using Same-Day Microdeposits) by calling the free-to-use [**/accounts/get**](https://plaid.com/docs/api/accounts/#accountsget) endpoint. Many Plaid products other than Balance also provide cached balance data as part of the accounts object. Because this data is cached, it is not suitable for situations where you need real-time balance information. The frequency with which cached balance data is updated varies depending on the product. An Item with Transactions, for example, may update cached balances once a day or more, while an Item with only Auth may update every 30 days or even less frequently.

For retrieving real-time balance for payments use cases, only [**/accounts/balance/get**](https://plaid.com/docs/api/products/balance/#accountsbalanceget) should be used. The exception is immediately after an Item has been linked, when the balance information is guaranteed to be recent. Note that because Balance always retrieves fresh data from the institution, latency will be higher when calling [**/accounts/balance/get**](https://plaid.com/docs/api/products/balance/#accountsbalanceget) than when calling other Plaid endpoints. While [**/accounts/balance/get**](https://plaid.com/docs/api/products/balance/#accountsbalanceget) typically takes about 10 seconds or less, the latency will vary by institution, and in some cases it may take 30 seconds or more to retrieve balance information.

If you want to evaluate the risk of an ACH return and need lower latency than [**/accounts/balance/get**](https://plaid.com/docs/api/products/balance/#accountsbalanceget), try [Signal](https://plaid.com/docs/signal/). Signal uses machine learning instead of real-time balance checks to evaluate the risk of an ACH return, and can return results in approximately one second or less.

#### [Current and available balance](https://plaid.com/docs/balance/" \l "current-and-available-balance)

Balance typically returns both current and available balance information. For fraud detection and insufficient funds (NSF) prevention use cases, available balance should be used, as it represents predicted balance net of any pending transactions, while current balance does not take pending transactions into account.

Available balance indicates balance that is available to spend, which is not the same as the overall value of the account. For example, for credit accounts, it indicates the amount that can be spent without hitting the credit limit (net of any pending transactions), while in investment accounts, it indicates the total available cash.

In some cases, a financial institution may not return available balance information. If that happens, you can calculate available balance by starting with the current balance, then using the [Transactions](https://plaid.com/docs/transactions/) product to detect any pending transactions and adjusting the balance accordingly.

##### [Typical balance fill rates by field](https://plaid.com/docs/balance/" \l "typical-balance-fill-rates-by-field)

| **FIELD** | **TYPICAL FILL RATE** |
| --- | --- |
| Current balance | 99% |
| Available balance | 91% |

#### [Sample app code](https://plaid.com/docs/balance/" \l "sample-app-code)

For a real-life example of an app that incorporates Balance, see the Node-based [Plaid Pattern Account Funding](https://github.com/plaid/pattern-account-funding) sample app. Pattern Account Funding is a sample account funding app that fetches Balance data to check for sufficient funds before a funds transfer. The Balance code can be found in [items.js](https://github.com/plaid/pattern-account-funding/blob/master/server/routes/items.js#L211-L234).

The entire github for this sample app can be found at [plaid/pattern-account-funding: Sample code to demonstrate Plaid-powered account funding use cases using the Auth, Identity, and Balance APIs. Includes examples of using Auth partners for end-to-end funds transfers. (github.com)](https://github.com/plaid/pattern-account-funding)

The sample balance code is as follows:  
```  
/\*\*

\* @file Defines all routes for the Items route.

\*/

const express = require('express');

const axios = require('axios');

const Boom = require('@hapi/boom');

const {

retrieveItemById,

retrieveItemByPlaidInstitutionId,

retrieveAccountsByItemId,

createItem,

deleteItem,

updateItemStatus,

createAccount,

updateBalances,

createTransfer,

} = require('../db/queries');

const { asyncWrapper } = require('../middleware');

const plaid = require('../plaid');

const {

sanitizeAccounts,

sanitizeItems,

isValidItemStatus,

validItemStatuses,

} = require('../util');

const router = express.Router();

const DWOLLA\_ACCESS\_TOKEN = process.env.DWOLLA\_ACCESS\_TOKEN;

const DWOLLA\_MASTER\_ACCOUNT\_ID = process.env.DWOLLA\_MASTER\_ACCOUNT\_ID;

const DWOLLA\_BASE\_URL = 'https://api-sandbox.dwolla.com';

/\*\*

\* First exchanges a public token for a private token via the Plaid API and

\* stores the newly created item in the DB. Then fetches auth data or processor token and identity data from

\* the Plaid API and creates and stores newly created account in the DB.

\*

\* @param {string} publicToken public token returned from the onSuccess call back in Link.

\* @param {string} institutionId the Plaid institution ID of the new item.

\* @param {string} userId the Plaid user ID of the active user.

\* @param {object} accounts the accounts chosen by the user from the onSuccess metadata.

\* @param {boolean} isProcessor false if developer is using a Plaid partner (processor)

\* @param {boolean} isIdentity true if in identity mode.

\*/

// create Dwolla Customer and obtain customer url

const createDwollaCustomer = async (firstName, lastName) => {

try {

const response = await axios.post(

`${DWOLLA\_BASE\_URL}/customers`,

{

firstName: firstName,

lastName: lastName,

email: `${Math.random() // because Dwolla does not allow identical emails, and sandbox data is always the same.

.toString(36)

.slice(2)}@sampleApp.com`,

ipAddress: '99.99.99.99', // dummy data: a unique identifier for Dwolla

},

{

headers: {

'content-type': 'application/json',

Authorization: `Bearer ${DWOLLA\_ACCESS\_TOKEN}`,

Accept: 'application/vnd.dwolla.v1.hal+json',

},

}

);

return response.headers.location;

} catch (error) {

console.log('error:', error);

res.status(500);

}

};

// send processor token to Dwolla customer url to create customer Funding source and obtain customer funding source url

const createDwollaCustomerFundingSource = async (

account,

customerUrl,

processorToken

) => {

try {

const response = await axios.post(

`${customerUrl}/funding-sources`,

{

plaidToken: processorToken,

name: account.subtype,

},

{

headers: {

'content-type': 'application/json',

Authorization: `Bearer ${DWOLLA\_ACCESS\_TOKEN}`,

Accept: 'application/vnd.dwolla.v1.hal+json',

},

}

);

return response.headers.location;

} catch (error) {

console.log('error:', error);

res.status(500);

}

};

router.post(

'/',

asyncWrapper(async (req, res) => {

const {

publicToken,

institutionId,

userId,

accounts,

isProcessor,

isIdentity,

} = req.body;

// exchange the public token for a private access token and store with the item.

const response = await plaid.itemPublicTokenExchange({

public\_token: publicToken,

});

const accessToken = response.data.access\_token;

const itemId = response.data.item\_id;

const newItem = await createItem(

institutionId,

accessToken,

itemId,

userId

);

// in case developer did not customize their Account Select in the dashboard to enable only one account,

// choose the checking or savings account.

const checkingAccount = accounts.filter(

account => account.subtype === 'checking'

);

const savingsAccount = accounts.filter(

account => account.subtype === 'savings'

);

const account =

accounts.length === 1

? accounts[0]

: checkingAccount.length > 0

? checkingAccount[0]

: savingsAccount[0];

// the request is the same for both auth and identity calls

const authAndIdRequest = {

access\_token: accessToken,

options: {

account\_ids: [account.id],

},

};

// identity info will remain null if not identity

let emails = null;

let ownerNames = null;

let firstName = 'firstName';

let lastName = 'lastName';

// auth numbers will remain null if not auth

let authNumbers = {

account: null,

routing: null,

wire\_routing: null,

};

// balances will be null if not auth or identity, only until the first transfer is made

// and accounts/balance/get is called

let balances = {

available: null,

current: null,

iso\_currency\_code: null,

unofficial\_currency\_code: null,

};

if (isIdentity) {

const identityResponse = await plaid.identityGet(authAndIdRequest);

emails = identityResponse.data.accounts[0].owners[0].emails.map(email => {

return email.data;

});

ownerNames = identityResponse.data.accounts[0].owners[0].names;

const fullName = ownerNames[0].split(' ');

firstName = fullName[0];

lastName = fullName[fullName.length - 1];

if (isProcessor) {

// in this case, authGet will not be called, so we'll need the balance from the identity call.

balances = identityResponse.data.accounts[0].balances;

}

}

// processorToken is only set if IS\_PROCESSOR is true in .env file

let processorToken = null;

let customerUrl = null;

let fundingSourceUrl = null;

if (!isProcessor) {

authResponse = await plaid.authGet(authAndIdRequest);

authNumbers = authResponse.data.numbers.ach[0];

balances = authResponse.data.accounts[0].balances;

} else {

const processorRequest = {

access\_token: accessToken,

account\_id: account.id,

processor: 'dwolla',

};

const processorTokenResponse = await plaid.processorTokenCreate(

processorRequest

);

processorToken = processorTokenResponse.data.processor\_token;

// create Dwolla Customer and obtain customer url

customerUrl = await createDwollaCustomer(firstName, lastName);

// send processor token to Dwolla customer url to create customer Funding source and obtain customer funding source url

fundingSourceUrl = await createDwollaCustomerFundingSource(

account,

customerUrl,

processorToken

);

}

// if not isProcessor, processorToken, customerUrl and fundingSouceUrl will all be null

const newAccount = await createAccount(

itemId,

userId,

account,

balances,

authNumbers,

ownerNames,

emails,

processorToken,

customerUrl,

fundingSourceUrl

);

res.json({

items: sanitizeItems(newItem),

accounts: sanitizeAccounts(newAccount),

});

})

);

// Make Dwolla transfer from customer funding source to app's master account. Obtain Dwolla transfer url

// and save to transfers table.

// TODO: make transfers provider and obtain transfers by item to disply on UI.

router.post(

'/makeTransfer',

asyncWrapper(async (req, res) => {

const { fundingSourceUrl, amount, itemId } = req.body;

let transUrl = null;

let confirmedAmount = null;

try {

const response = await axios.post(

`${DWOLLA\_BASE\_URL}/transfers`,

{

\_links: {

source: {

href: fundingSourceUrl,

},

destination: {

href: `${DWOLLA\_BASE\_URL}/funding-sources/${DWOLLA\_MASTER\_ACCOUNT\_ID}`,

},

},

amount: {

currency: 'USD',

value: `${amount}`,

},

},

{

headers: {

'content-type': 'application/json',

Authorization: `Bearer ${DWOLLA\_ACCESS\_TOKEN}`,

Accept: 'application/vnd.dwolla.v1.hal+json',

},

}

);

transUrl = response.headers.location;

} catch (error) {

console.log('error:', error);

res.status(500);

}

// get the confirmed amount from the transfer url

try {

const response = await axios.get(transUrl, {

headers: {

'content-type': 'application/json',

Authorization: `Bearer ${DWOLLA\_ACCESS\_TOKEN}`,

Accept: 'application/vnd.dwolla.v1.hal+json',

},

});

confirmedAmount = response.data.amount.value;

} catch (error) {

console.log('error:', error);

res.status(500);

}

const transfer = await createTransfer(itemId, confirmedAmount, transUrl);

res.json({ transfer: transfer });

})

);

/\*\*

\* Retrieves a single item.

\*

\* @param {string} itemId the ID of the item.

\* @returns {Object[]} an array containing a single item.

\*/

router.get(

'/:itemId',

asyncWrapper(async (req, res) => {

const { itemId } = req.params;

const item = await retrieveItemById(itemId);

res.json(sanitizeItems(item));

})

);

/\*\*

\* Updates a single item.

\*

\* @param {string} itemId the ID of the item.

\* @returns {Object[]} an array containing a single item.

\*/

router.put(

'/:itemId',

asyncWrapper(async (req, res) => {

const { itemId } = req.params;

const { status } = req.body;

if (status) {

if (!isValidItemStatus(status)) {

throw new Boom(

'Cannot set item status. Please use an accepted value.',

{

statusCode: 400,

acceptedValues: [validItemStatuses.values()],

}

);

}

await updateItemStatus(itemId, status);

const item = await retrieveItemById(itemId);

res.json(sanitizeItems(item));

} else {

throw new Boom('You must provide updated item information.', {

statusCode: 400,

acceptedKeys: ['status'],

});

}

})

);

/\*\*

\* Updates balances on account

\*

\* @param {number} itemId the ID of the item.

\* @param {string} accountId the account id.

\* @returns {Object[]} an array containing a single account.

\*/

router.put(

'/:itemId/balance',

asyncWrapper(async (req, res) => {

const { itemId } = req.params;

const { accountId } = req.body;

const { plaid\_access\_token: accessToken } = await retrieveItemById(itemId);

const balanceRequest = {

access\_token: accessToken,

options: {

account\_ids: [accountId],

},

};

const balanceResponse = await plaid.accountsBalanceGet(balanceRequest);

const account = balanceResponse.data.accounts[0];

const updatedAccount = await updateBalances(

accountId,

account.balances.current,

account.balances.available

);

res.json(updatedAccount[0]);

})

);

/\*\*

\* Deletes a single item and related accounts and transactions.

\* Also removes the item from the Plaid API

\* access\_token associated with the Item is no longer valid

\* https://plaid.com/docs/#remove-item-request

\* @param {string} itemId the ID of the item.

\* @returns status of 204 if successful

\*/

router.delete(

'/:itemId',

asyncWrapper(async (req, res) => {

const { itemId } = req.params;

const { plaid\_access\_token: accessToken } = await retrieveItemById(itemId);

/\* eslint-disable camelcase \*/

try {

const response = await plaid.itemRemove({

access\_token: accessToken,

});

const removed = response.data.removed;

const status\_code = response.data.status\_code;

} catch (error) {

if (!removed)

throw new Boom('Item could not be removed in the Plaid API.', {

statusCode: status\_code,

});

}

await deleteItem(itemId);

res.sendStatus(204);

})

);

/\*\*

\* Retrieves all accounts associated with a single item.

\*

\* @param {string} itemId the ID of the item.

\* @returns {Object[]} an array of accounts.

\*/

router.get(

'/:itemId/accounts',

asyncWrapper(async (req, res) => {

const { itemId } = req.params;

const accounts = await retrieveAccountsByItemId(itemId);

res.json(sanitizeAccounts(accounts));

})

);

/\*\*

\* -- This endpoint will only work in the sandbox enviornment --

\* Forces an Item into an ITEM\_LOGIN\_REQUIRED (bad) error state.

\* An ITEM\_LOGIN\_REQUIRED webhook will be fired after a call to this endpoint.

\* https://plaid.com/docs/#managing-item-states

\*

\* @param {string} itemId the Plaid ID of the item.

\* @return {Object} the response from the Plaid API.

\*/

router.post(

'/sandbox/item/reset\_login',

asyncWrapper(async (req, res) => {

try {

const { itemId } = req.body;

const { plaid\_access\_token: accessToken } = await retrieveItemById(

itemId

);

const resetResponse = await plaid.sandboxItemResetLogin({

access\_token: accessToken,

});

res.json(resetResponse.data);

} catch (error) {

if (error.response && error.response.status === 400) {

console.log(

'Ngrok webhook addresses are only valid for 2 hours and only during the session in which an item is created; for previously created items, no webhook will be received from the call to sandboxItemResetLogin. If your current session has been longer than 2 hours, restart your server to test the item reset login. Otherwise, create a new item to test. For more information, see the troubleshooting guide in the readme file.'

);

}

}

})

);

module.exports = router;

```

# Add Balance to your app

#### Use Balance to fetch real-time balance data

In this guide, we'll start from scratch and walk through how to use [Balance](https://plaid.com/docs/api/products/balance/) to retrieve real-time balance information. If you are already familiar with using Plaid and are set up to make calls to the Plaid API, make sure to note that you should not include balance in the [**/link/token/create**](https://plaid.com/docs/api/tokens/#linktokencreate) products array; you can then skip ahead to [Fetching balance data](https://plaid.com/docs/balance/add-to-app/#fetching-balance-data).

#### [Get Plaid API keys and complete application and company profile](https://plaid.com/docs/balance/add-to-app/" \l "get-plaid-api-keys-and-complete-application-and-company-profile)

If you don't already have one, you'll need to [create a Plaid developer account](https://dashboard.plaid.com/signup). After creating your account, you can find your [API keys](https://dashboard.plaid.com/developers/keys) under the Team Settings menu on the Plaid Dashboard.

You will also need to complete your [application profile](https://dashboard.plaid.com/settings/company/app-branding) and [company profile](https://dashboard.plaid.com/settings/company/profile) in the Dashboard. The information in your profile will be shared with users of your application when they manage their connection on the [Plaid Portal](https://my.plaid.com/). Your application profile and company profile must be completed before connecting to certain institutions in Production.

#### [Install and initialize Plaid libraries](https://plaid.com/docs/balance/add-to-app/" \l "install-and-initialize-plaid-libraries)

You can use our official server-side client libraries to connect to the Plaid API from your application:

Select group for content switcher

Current librariesLegacy libraries

Terminal

**Curl**

**Select Language**

**Copy**

1## no setup

After you've installed Plaid's client libraries, you can initialize them by passing in your client\_id, secret, and the environment you wish to connect to (Sandbox, Development, or Production). This will make sure the client libraries pass along your client\_id and secret with each request, and you won't need to explicitly include them in any other calls.

In the code samples below, you will need to replace PLAID\_CLIENT\_ID and PLAID\_SECRET with your own keys, which you can obtain from the [Dashboard](https://dashboard.plaid.com/developers/keys). These code samples also demonstrate starting up a server commonly used in each framework (such as Express or Flask).

Select group for content switcher

Current librariesLegacy libraries

**Curl**

**Select Language**

**Copy**

1## no setup

#### [Create an Item in Link](https://plaid.com/docs/balance/add-to-app/" \l "create-an-item-in-link)

Plaid Link is a drop-in module that provides a secure, elegant authentication flow for each institution that Plaid supports. Link makes it secure and easy for users to connect their bank accounts to Plaid. Note that these instructions cover Link on the web. For instructions on using Link within mobile apps, see the [Link documentation](https://plaid.com/docs/link/).

Using Link, we will create a Plaid Item, which is a Plaid term for a login at a financial institution. An Item is not the same as a financial institution account, although every account will be associated with an Item. For example, if a user has one login at their bank that allows them to access both their checking account and their savings account, a single Item would be associated with both of those accounts. If you want to customize Link's look and feel, you can do so from the [Dashboard](https://dashboard.plaid.com/link).

Before initializing Link, you will need to create a new link\_token on the server side of your application. A link\_token is a short-lived, one-time use token that is used to authenticate your app with Link. You can create one using the [**/link/token/create**](https://plaid.com/docs/api/tokens/#linktokencreate) endpoint. Then, on the client side of your application, you'll need to initialize Link with the link\_token that you just created.

balance cannot be used as a product when calling [**/link/token/create**](https://plaid.com/docs/api/tokens/#linktokencreate). You must create the link\_token with the product or products you will be using with Balance, instead. You will then automatically get access to the Balance product.

##### [Create a link\_token](https://plaid.com/docs/balance/add-to-app/" \l "create-a-link_token)

Select group for content switcher

Current librariesLegacy libraries

**Curl**

**Select Language**

**Copy**

1curl -X POST https://sandbox.plaid.com/link/token/create \

2-H 'Content-Type: application/json' \

3-d '{

4 "client\_id": "${PLAID\_CLIENT\_ID}",

5 "secret": "${PLAID\_SECRET}",

6 "client\_name": "Plaid Test App",

7 "user": { "client\_user\_id": "${UNIQUE\_USER\_ID}" },

8 "products": ["${PRODUCT}"],

9 "country\_codes": ["US"],

10 "language": "en",

11 "webhook": "https://webhook.example.com",

12 "redirect\_uri": "https://domainname.com/oauth-page.html",

13}'

##### [Install Link dependency](https://plaid.com/docs/balance/add-to-app/" \l "install-link-dependency)

index.html

**Curl**

**Select Language**

**Copy**

1<head>

2 <title>Connect a bank</title>

3 <script src="https://cdn.plaid.com/link/v2/stable/link-initialize.js"></script>

4</head>

##### [Configure the client-side Link handler](https://plaid.com/docs/balance/add-to-app/" \l "configure-the-client-side-link-handler)

app.js

**Copy**

1const linkHandler = Plaid.create({

2 token: (await $.post('/create\_link\_token')).link\_token,

3 onSuccess: (public\_token, metadata) => {

4 // Send the public\_token to your app server.

5 $.post('/exchange\_public\_token', {

6 public\_token: public\_token,

7 });

8 },

9 onExit: (err, metadata) => {

10 // Optionally capture when your user exited the Link flow.

11 // Storing this information can be helpful for support.

12 },

13 onEvent: (eventName, metadata) => {

14 // Optionally capture Link flow events, streamed through

15 // this callback as your users connect an Item to Plaid.

16 },

17});

18

19linkHandler.open();

#### [Get a persistent access\_token](https://plaid.com/docs/balance/add-to-app/" \l "get-a-persistent-access_token)

Next, on the server side, we need to exchange our public\_token for an access\_token and item\_id. The access\_token will allow us to make authenticated calls to the Plaid API. Doing so is as easy as calling the [**/item/public\_token/exchange**](https://plaid.com/docs/api/tokens/#itempublic_tokenexchange) endpoint from our server-side handler. We'll use the client library we configured earlier to make the API call.

Save the access\_token and item\_id in a secure datastore, as they’re used to access Item data and identify webhooks, respectively. The access\_token will remain valid unless you actively chose to expire it via rotation or remove the corresponding Item via [**/item/remove**](https://plaid.com/docs/api/items/#itemremove). The access\_token should be stored securely, and never in client-side code. A public\_token is a one-time use token with a lifetime of 30 minutes, so there is no need to store it.

Select group for content switcher

Current librariesLegacy libraries

**Curl**

**Select Language**

**Copy**

1curl -X POST https://sandbox.plaid.com/item/public\_token/exchange \

2-H 'Content-Type: application/json' \

3-d '{

4 "client\_id": "${PLAID\_CLIENT\_ID}",

5 "secret": "${PLAID\_SECRET}",

6 "public\_token": "public-sandbox-12345678-abcd-1234-abcd-1234567890ab"

7}'

#### [Fetching Balance data](https://plaid.com/docs/balance/add-to-app/" \l "fetching-balance-data)

Now that the authentication step is out of the way, we can begin using authenticated endpoints from the Plaid API. Once you've retrieved balance data for an account, you can then use it to evaluate whether a potential transaction will go through. For more detailed information on the schema returned, see [**/accounts/balance/get**](https://plaid.com/docs/api/products/balance/#accountsbalanceget).

Select group for content switcher

Current librariesLegacy libraries

/accounts/balance/get

**Curl**

**Select Language**

**Copy**

1curl -X POST https://sandbox.plaid.com/accounts/balance/get \

2-H 'Content-Type: application/json' \

3-d '{

4 "client\_id": "${PLAID\_CLIENT\_ID}",

5 "secret": "${PLAID\_SECRET}",

6 "access\_token": String,

7 "options": {

8 "account\_ids": [String]

9 }

10}'

Example response data is below.

Balance sample response

**Copy**

1{

2 "accounts": [

3 {

4 "account\_id": "BxBXxLj1m4HMXBm9WZZmCWVbPjX16EHwv99vp",

5 "balances": {

6 "available": 100,

7 "current": 110,

8 "iso\_currency\_code": "USD",

9 "limit": null,

10 "unofficial\_currency\_code": null

11 },

12 "mask": "0000",

13 "name": "Plaid Checking",

14 "official\_name": "Plaid Gold Standard 0% Interest Checking",

15 "subtype": "checking",

16 "type": "depository"

17 },

18 {

19 "account\_id": "dVzbVMLjrxTnLjX4G66XUp5GLklm4oiZy88yK",

20 "balances": {

21 "available": null,

22 "current": 410,

23 "iso\_currency\_code": "USD",

24 "limit": 2000,

25 "unofficial\_currency\_code": null

26 },

27 "mask": "3333",

28 "name": "Plaid Credit Card",

29 "official\_name": "Plaid Diamond 12.5% APR Interest Credit Card",

30 "subtype": "credit card",

31 "type": "credit"

32 },

33 {

34 "account\_id": "Pp1Vpkl9w8sajvK6oEEKtr7vZxBnGpf7LxxLE",

35 "balances": {

36 "available": null,

37 "current": 65262,

38 "iso\_currency\_code": "USD",

39 "limit": null,

40 "unofficial\_currency\_code": null

41 },

42 "mask": "7777",

43 "name": "Plaid Student Loan",

44 "official\_name": null,

45 "subtype": "student",

46 "type": "loan"

47 }

48 ],

49 "item": {

50 "available\_products": [

51 "balance",

52 "credit\_details",

53 "identity",

54 "investments"

55 ],

56 "billed\_products": ["assets", "auth", "liabilities", "transactions"],

57 "consent\_expiration\_time": null,

58 "error": null,

59 "institution\_id": "ins\_3",

60 "item\_id": "eVBnVMp7zdTJLkRNr33Rs6zr7KNJqBFL9DrE6",

61 "webhook": "https://www.genericwebhookurl.com/webhook"

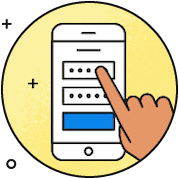
62 },

63 "request\_id": "qk5Bxes3gDfv4F2"

64}

# [Introduction to Transactions](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)

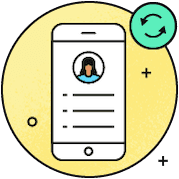
#### [Retrieve up to 24 months of transaction data and stay up-to-date with webhooks.](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)

[](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)

[LINK AN ACCOUNT](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)

[](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)

[FETCH TRANSACTIONS](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)

[](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)

[SHOW TRANSACTIONS](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)

#### [API Reference](https://plaid.com/docs/api/products/transactions/)

[View Transactions requests, responses, and example code](https://plaid.com/docs/api/products/transactions/)

**[View Transactions API](https://plaid.com/docs/api/products/transactions/)**

#### [Quickstart](https://plaid.com/docs/quickstart/)

[Learn about Plaid's key concepts and run starter code](https://plaid.com/docs/quickstart/)

**[Get started](https://plaid.com/docs/quickstart/)**

#### [Overview](https://plaid.com/docs/transactions/" \l "overview)

[Transactions data can be useful for many different applications, including personal finance management, expense reporting, cash flow modeling, risk analysis, and more. Plaid's Transactions product allows you to access a user's transaction history for depository type accounts such as checking and savings accounts, credit type accounts such as credit cards, and student loan accounts. For transaction history from investment accounts, use Plaid's](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)[[Investments](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)](https://plaid.com/docs/investments/)[product.](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)

[Transactions data available via](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)[**[/transactions/sync](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)**](https://plaid.com/docs/api/products/transactions/#transactionssync)[includes transaction date, amount, category, merchant, location, and more. Transaction data is lightly cleaned to populate the name field, and more thoroughly processed to populate the merchant\_name field.](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)

**[Copy](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)**

[1{](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)

[2 "account\_id": "BxBXxLj1m4HMXBm9WZZmCWVbPjX16EHwv99vp",](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)

[3 "amount": 2307.21,](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)

[4 "iso\_currency\_code": "USD",](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)

[5 "unofficial\_currency\_code": null,](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)

[6 "category": ["Shops", "Computers and Electronics"],](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)

[7 "category\_id": "19013000",](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)

[8 "check\_number": null,](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)

[9 "date": "2023-01-29",](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)

[10 "datetime": "2023-01-27T11:00:00Z",](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)

[11 "authorized\_date": "2023-01-27",](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)

[12 "authorized\_datetime": "2023-01-27T10:34:50Z",](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)

[13 "location": {](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)

[14 "address": "300 Post St",](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)

[15 "city": "San Francisco",](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)

[16 "region": "CA",](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)

[17 "postal\_code": "94108",](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)

[18 "country": "US",](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)

[19 "lat": 40.740352,](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)

[20 "lon": -74.001761,](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)

[21 "store\_number": "1235"](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)

[22 },](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)

[23 "name": "Apple Store",](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)

[24 "merchant\_name": "Apple",](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)

[25 "payment\_meta": {](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)

[26 "by\_order\_of": null,](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)

[27 "payee": null,](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)

[28 "payer": null,](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)

[29 "payment\_method": null,](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)

[30 "payment\_processor": null,](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)

[31 "ppd\_id": null,](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)

[32 "reason": null,](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)

[33 "reference\_number": null](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)

[34 },](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)

[35 "payment\_channel": "in store",](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)

[36 "pending": false,](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)

[37 "pending\_transaction\_id": null,](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)

[38 "personal\_finance\_category": {](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)

[39 "primary": "GENERAL\_MERCHANDISE",](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)

[40 "detailed": "GENERAL\_MERCHANDISE\_ELECTRONICS"](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)

[41 },](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)

[42 "account\_owner": null,](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)

[43 "transaction\_id": "lPNjeW1nR6CDn5okmGQ6hEpMo4lLNoSrzqDje",](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)

[44 "transaction\_code": null,](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)

[45 "transaction\_type": "place"](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)

[46}](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)

#### [Transactions updates](https://plaid.com/docs/transactions/" \l "transactions-updates)

[Transactions data is not static. As time passes, your users will make new transactions, and transactions they made in the past will change as they are processed by the financial institution. To learn more about how transactions are processed and can change, see](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)[[Transaction states](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)](https://plaid.com/docs/transactions/transactions-data/)[.](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)

[Plaid checks for updated transactions regularly, and uses webhooks to notify you of any changes so you can keep your app up to date. For more detail on how to listen and respond to transaction update webhooks, see](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)[[Transaction webhooks](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)](https://plaid.com/docs/transactions/webhooks/)[.](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)

[The frequency of transactions update checks is typically one or more times a day. The exact frequency will depend on the institution. To learn when an Item was last checked for updates, you can view the Item in the](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)[[Item Debugger](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)](https://dashboard.plaid.com/activity/debugger)[. If you would like to display this information in your app's UI to help users understand the freshness of their data, it can also be retrieved via API, using the](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)[**[/item/get](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)**](https://plaid.com/docs/api/items/#itemget)[endpoint.](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)

[You can also request an update on-demand via the](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)[**[/transactions/refresh](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)**](https://plaid.com/docs/api/products/transactions/#transactionsrefresh)[endpoint, which is available as an add-on for Transactions customers. To request access to this endpoint, submit a](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)[[product access request](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)](https://dashboard.plaid.com/settings/team/products)[or contact your Plaid account manager.](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)

#### [Recurring transactions](https://plaid.com/docs/transactions/" \l "recurring-transactions)

[If your app involves personal financial management functionality, you may want to view a summary of a user's inflows and outflows. The](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)[**[/transactions/recurring/get](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)**](https://plaid.com/docs/api/products/transactions/#transactionsrecurringget)[endpoint provides a summary of the recurring outflow and inflow streams and includes insights about each recurring stream including the category, merchant, last amount, and more.](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)[**[/transactions/recurring/get](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)**](https://plaid.com/docs/api/products/transactions/#transactionsrecurringget)[is available as an add-on for Transactions customers in the US, Canada, and UK. To request access to this endpoint, submit a](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)[[product access request](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)](https://dashboard.plaid.com/settings/team/products)[or contact your Plaid account manager.](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)

#### [Sample app demo and code](https://plaid.com/docs/transactions/" \l "sample-app-demo-and-code)

[Try out the](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)[[Pattern Demo](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)](https://pattern.plaid.com/)[for a demonstration of a sample app that uses Plaid's Transactions product for the personal financial management use case to calculate and display a customer's net worth and spending habits and allows them to link multiple financial institutions.](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)

[For a more robust example of an app that incorporates transactions, along with sample code for transactions reconciliation, see the Node-based](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)[[Plaid Pattern](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)](https://github.com/plaid/pattern)[sample app.](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)

#### [Updating your app](https://plaid.com/docs/transactions/" \l "updating-your-app)

[If you've already built an app using Plaid's](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)[**[/transactions/get](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)**](https://plaid.com/docs/api/products/transactions/#transactionsget)[endpoint, see the](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)[[Transactions Sync migration guide](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)](https://plaid.com/docs/transactions/sync-migration/)[to learn about Plaid's latest Transactions API and how to simplify your integration by migrating to](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)[**[/transactions/sync](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)**](https://plaid.com/docs/api/products/transactions/#transactionssync)[.](https://plaid.com/docs/balance/add-to-app/" \l "tutorial-and-example-code-in-plaid-pattern)

# Add Transactions to your app

#### Learn how to fetch Transactions data for your users

Try out the [Pattern Demo](https://pattern.plaid.com/) for a demonstration of a sample app that uses Plaid's Transactions product for the personal financial management use case.

In this guide, we'll start from scratch and walk through how to use Transactions to perform an initial fetch of a user's transaction history. If you are already familiar with using Plaid and are set up to make calls to the Plaid API, you can skip ahead to [Fetching transaction data](https://plaid.com/docs/transactions/add-to-app/#fetching-transaction-data).

Prefer to learn by watching? A [video tutorial](https://youtu.be/Pin0-ceDKcI) is available for this topic.

#### [Get Plaid API keys and complete application and company profile](https://plaid.com/docs/transactions/add-to-app/" \l "get-plaid-api-keys-and-complete-application-and-company-profile)

If you don't already have one, you'll need to [create a Plaid developer account](https://dashboard.plaid.com/signup). After creating your account, you can find your [API keys](https://dashboard.plaid.com/developers/keys) under the Team Settings menu on the Plaid Dashboard.

You will also need to complete your [application profile](https://dashboard.plaid.com/settings/company/app-branding) and [company profile](https://dashboard.plaid.com/settings/company/profile) on the Dashboard. The information in your profile will be shared with users of your application when they manage their connection on the [Plaid Portal](https://my.plaid.com/). Your application profile and company profile must be completed before connecting to certain institutions in Production.

#### [Install and initialize Plaid libraries](https://plaid.com/docs/transactions/add-to-app/" \l "install-and-initialize-plaid-libraries)

You can use our official server-side client libraries to connect to the Plaid API from your application:

Select group for content switcher

Current librariesLegacy libraries

Terminal

**Curl**

**Select Language**

**Copy**

1## no setup

After you've installed Plaid's client libraries, you can initialize them by passing in your client\_id, secret, and the environment you wish to connect to (Sandbox, Development, or Production). This will make sure the client libraries pass along your client\_id and secret with each request, and you won't need to explicitly include them in any other calls.

In the code samples below, you will need to replace PLAID\_CLIENT\_ID and PLAID\_SECRET with your own keys, which you can obtain from the [Dashboard](https://dashboard.plaid.com/developers/keys). These code samples also demonstrate starting up a server commonly used in each framework (such as Express or Flask).

Select group for content switcher

Current librariesLegacy libraries

**Curl**

**Select Language**

**Copy**

1## no setup

#### [Create an Item with Link](https://plaid.com/docs/transactions/add-to-app/" \l "create-an-item-with-link)

Plaid Link is a drop-in module that provides a secure, elegant authentication flow for each institution that Plaid supports. Link makes it secure and easy for users to connect their bank accounts to Plaid. Note that these instructions cover Link on the web. For instructions on using Link within mobile apps, see the [Link documentation](https://plaid.com/docs/link/).

Using Link, we will create a Plaid Item, which is a Plaid term for a login at a financial institution. An Item is not the same as a financial institution account, although every account will be associated with an Item. For example, if a user has one login at their bank that allows them to access both their checking account and their savings account, a single Item would be associated with both of those accounts. If you want to customize Link's look and feel, you can do so from the [Dashboard](https://dashboard.plaid.com/link).

Before initializing Link, you will need to create a new link\_token on the server side of your application. A link\_token is a short-lived, one-time use token that is used to authenticate your app with Link. You can create one using the [**/link/token/create**](https://plaid.com/docs/api/tokens/#linktokencreate) endpoint. Then, on the client side of your application, you'll need to initialize Link with the link\_token that you just created.

The [**/link/token/create**](https://plaid.com/docs/api/tokens/#linktokencreate) sample code below will create an Item with a maximum of 90 days of transaction history. To request more, set the transactions.days\_requested parameter in the [**/link/token/create**](https://plaid.com/docs/api/tokens/#linktokencreate) request.

##### [Create a link\_token](https://plaid.com/docs/transactions/add-to-app/" \l "create-a-link_token)

Select group for content switcher

Current librariesLegacy libraries

**Curl**

**Select Language**

**Copy**

1curl -X POST https://sandbox.plaid.com/link/token/create \

2-H 'Content-Type: application/json' \

3-d '{

4 "client\_id": "${PLAID\_CLIENT\_ID}",

5 "secret": "${PLAID\_SECRET}",

6 "client\_name": "Plaid Test App",

7 "user": { "client\_user\_id": "${UNIQUE\_USER\_ID}" },

8 "products": ["${PRODUCT}"],

9 "country\_codes": ["US"],

10 "language": "en",

11 "webhook": "https://webhook.example.com",

12 "redirect\_uri": "https://domainname.com/oauth-page.html",

13}'

##### [Install Link dependency](https://plaid.com/docs/transactions/add-to-app/" \l "install-link-dependency)

index.html

**Curl**

**Select Language**

**Copy**

1<head>

2 <title>Connect a bank</title>

3 <script src="https://cdn.plaid.com/link/v2/stable/link-initialize.js"></script>

4</head>

##### [Configure the client-side Link handler](https://plaid.com/docs/transactions/add-to-app/" \l "configure-the-client-side-link-handler)

app.js

**Copy**

1const linkHandler = Plaid.create({

2 token: (await $.post('/create\_link\_token)).link\_token,

3 onSuccess: (public\_token, metadata) => {

4 // Send the public\_token to your app server.

5 $.post('/exchange\_public\_token', {

6 public\_token: public\_token,

7 });

8 },

9 onExit: (err, metadata) => {

10 // Optionally capture when your user exited the Link flow.

11 // Storing this information can be helpful for support.

12 },

13 onEvent: (eventName, metadata) => {

14 // Optionally capture Link flow events, streamed through

15 // this callback as your users connect an Item to Plaid.

16 },

17});

18

19linkHandler.open();

#### [Get a persistent access token](https://plaid.com/docs/transactions/add-to-app/" \l "get-a-persistent-access-token)

Next, on the server side, we need to exchange our public\_token for an access\_token and item\_id. The access\_token will allow us to make authenticated calls to the Plaid API. Doing so is as easy as calling the [**/item/public\_token/exchange**](https://plaid.com/docs/api/tokens/#itempublic_tokenexchange) endpoint from our server-side handler. We'll use the client library we configured earlier to make the API call.

Save the access\_token and item\_id in a secure datastore, as they’re used to access Item data and identify webhooks, respectively. The access\_token will remain valid unless you actively chose to expire it via rotation or remove the corresponding Item via [**/item/remove**](https://plaid.com/docs/api/items/#itemremove). The access\_token should be stored securely, and never in client-side code. A public\_token is a one-time use token with a lifetime of 30 minutes, so there is no need to store it.

Select group for content switcher

Current librariesLegacy libraries

**Curl**

**Select Language**

**Copy**

1curl -X POST https://sandbox.plaid.com/item/public\_token/exchange \

2-H 'Content-Type: application/json' \

3-d '{

4 "client\_id": "${PLAID\_CLIENT\_ID}",

5 "secret": "${PLAID\_SECRET}",

6 "public\_token": "public-sandbox-12345678-abcd-1234-abcd-1234567890ab"

7}'

#### [Fetching transaction data](https://plaid.com/docs/transactions/add-to-app/" \l "fetching-transaction-data)

Now that the authentication step is out of the way, we can begin using authenticated endpoints from the Plaid API and fetch transaction data using the [**/transactions/sync**](https://plaid.com/docs/api/products/transactions/#transactionssync) endpoint.

The [**/transactions/sync**](https://plaid.com/docs/api/products/transactions/#transactionssync) endpoint is used to both initialize your view of transactions data, and keep you current with any changes that have occurred. When you first call it on an Item with no cursor parameter, transactions data available at that time is returned. If more updates are available than requested with the count parameter (maximum of 500), has\_more will be set to true, indicating the endpoint should be called again, using the next\_cursor from the previous response in the cursor field of the next request, to receive another page of data. After successfully pulling all currently available pages, you can store the cursor for later requests, allowing Plaid to send you new updates from when you last queried the endpoint.

Note that if you encounter an error during pagination, it's important to restart the pagination loop from the beginning. For more details, see the documentation for [**TRANSACTIONS\_SYNC\_MUTATION\_DURING\_PAGINATION**](https://plaid.com/docs/errors/transactions/#transactions_sync_mutation_during_pagination). For sample code for handling the error, see the [**/transactions/sync**](https://plaid.com/docs/api/products/transactions/#transactionssync) API reference.

Typically, the first 30 days of transaction history is available to be fetched almost immediately, but full transaction history may take a minute or more to become available. If you get an empty response when calling [**/transactions/sync**](https://plaid.com/docs/api/products/transactions/#transactionssync) shortly after linking an Item, it's likely that the first 30 days of transaction history has not yet been pulled. You will need to call the endpoint when the data is pulled. Similarly, if you only get the first 30 days of transaction history, you will need to wait until it is complete, and call the endpoint again.

To be notified whenever additional data becomes available, see [Transaction webhooks](https://plaid.com/docs/transactions/webhooks/).

/transactions/sync

**Curl**

**Select Language**

**Copy**

1curl -X POST https://sandbox.plaid.com/transactions/sync \

2-H 'Content-Type: application/json' \

3-d '{

4 "client\_id": "${PLAID\_CLIENT\_ID}"

5 "secret": "${PLAID\_SECRET}"

6 "access\_token": String,

7 "cursor": String,

8 "count": 250

9}'

#### [Updating transaction data](https://plaid.com/docs/transactions/add-to-app/" \l "updating-transaction-data)

After your initial [**/transactions/sync**](https://plaid.com/docs/api/products/transactions/#transactionssync) request, you may want your application to be notified when any transactions are added, removed, or modified in order to immediately fetch them from [**/transactions/sync**](https://plaid.com/docs/api/products/transactions/#transactionssync). To learn how, see [Transaction Webhooks](https://plaid.com/docs/transactions/webhooks/).

#### [Example code in Plaid Pattern](https://plaid.com/docs/transactions/add-to-app/" \l "example-code-in-plaid-pattern)

For a real-life example of an app that incorporates transactions, see the Node-based [Plaid Pattern](https://github.com/plaid/pattern) sample app. Pattern is a sample financial management app that fetches transactions data upon receipt of transactions webhooks. Transactions code in Plaid Pattern can be found in [handleTransactionsWebhook.js](https://github.com/plaid/pattern/blob/master/server/webhookHandlers/handleTransactionsWebhook.js). Here is the handleTransactionsWebhook.js code as well:

,,,

/\*\*

\* @file Defines the handler for Transactions webhooks.

\* https://plaid.com/docs/#transactions-webhooks

\*/

const {

retrieveItemByPlaidItemId,

} = require('../db/queries');

const updateTransactions = require('../update\_transactions');

/\*\*

\* Handles all transaction webhook events. The transaction webhook notifies

\* you that a single item has new transactions available.

\*

\* @param {Object} requestBody the request body of an incoming webhook event

\* @param {Object} io a socket.io server instance.

\*/

const handleTransactionsWebhook = async (requestBody, io) => {

const {

webhook\_code: webhookCode,

item\_id: plaidItemId,

} = requestBody;

const serverLogAndEmitSocket = (additionalInfo, itemId) => {

console.log(

`WEBHOOK: TRANSACTIONS: ${webhookCode}: Plaid\_item\_id ${plaidItemId}: ${additionalInfo}`

);

// use websocket to notify the client that a webhook has been received and handled

if (webhookCode) io.emit(webhookCode, { itemId });

};

switch (webhookCode) {

case 'SYNC\_UPDATES\_AVAILABLE': {

// Fired when new transactions data becomes available.

const {

addedCount,

modifiedCount,

removedCount,

} = await updateTransactions(plaidItemId);

const { id: itemId } = await retrieveItemByPlaidItemId(plaidItemId);

serverLogAndEmitSocket(`Transactions: ${addedCount} added, ${modifiedCount} modified, ${removedCount} removed`, itemId);

break;

}

case 'DEFAULT\_UPDATE':

case 'INITIAL\_UPDATE':

case 'HISTORICAL\_UPDATE':

/\* ignore - not needed if using sync endpoint + webhook \*/

break;

default:

serverLogAndEmitSocket(`unhandled webhook type received.`, plaidItemId);

}

};

module.exports = handleTransactionsWebhook;6 "access\_token": String,

```

#### [Fetching by date](https://plaid.com/docs/transactions/add-to-app/" \l "fetching-by-date)

If you want to fetch transactions data by date range, you can use the [**/transactions/get**](https://plaid.com/docs/api/products/transactions/#transactionsget) endpoint.

# Add Transactions to your app

#### Learn how to fetch Transactions data for your users

Try out the [Pattern Demo](https://pattern.plaid.com/) for a demonstration of a sample app that uses Plaid's Transactions product for the personal financial management use case.

In this guide, we'll start from scratch and walk through how to use Transactions to perform an initial fetch of a user's transaction history. If you are already familiar with using Plaid and are set up to make calls to the Plaid API, you can skip ahead to [Fetching transaction data](https://plaid.com/docs/transactions/add-to-app/#fetching-transaction-data).

Prefer to learn by watching? A [video tutorial](https://youtu.be/Pin0-ceDKcI) is available for this topic.

#### [Get Plaid API keys and complete application and company profile](https://plaid.com/docs/transactions/add-to-app/" \l "get-plaid-api-keys-and-complete-application-and-company-profile)

If you don't already have one, you'll need to [create a Plaid developer account](https://dashboard.plaid.com/signup). After creating your account, you can find your [API keys](https://dashboard.plaid.com/developers/keys) under the Team Settings menu on the Plaid Dashboard.

You will also need to complete your [application profile](https://dashboard.plaid.com/settings/company/app-branding) and [company profile](https://dashboard.plaid.com/settings/company/profile) on the Dashboard. The information in your profile will be shared with users of your application when they manage their connection on the [Plaid Portal](https://my.plaid.com/). Your application profile and company profile must be completed before connecting to certain institutions in Production.

#### [Install and initialize Plaid libraries](https://plaid.com/docs/transactions/add-to-app/" \l "install-and-initialize-plaid-libraries)

You can use our official server-side client libraries to connect to the Plaid API from your application:

Select group for content switcher

Current librariesLegacy libraries

Terminal

**Curl**

**Select Language**

**Copy**

1## no setup

After you've installed Plaid's client libraries, you can initialize them by passing in your client\_id, secret, and the environment you wish to connect to (Sandbox, Development, or Production). This will make sure the client libraries pass along your client\_id and secret with each request, and you won't need to explicitly include them in any other calls.

In the code samples below, you will need to replace PLAID\_CLIENT\_ID and PLAID\_SECRET with your own keys, which you can obtain from the [Dashboard](https://dashboard.plaid.com/developers/keys). These code samples also demonstrate starting up a server commonly used in each framework (such as Express or Flask).

Select group for content switcher

Current librariesLegacy libraries

**Curl**

**Select Language**

**Copy**

1## no setup

#### [Create an Item with Link](https://plaid.com/docs/transactions/add-to-app/" \l "create-an-item-with-link)

Plaid Link is a drop-in module that provides a secure, elegant authentication flow for each institution that Plaid supports. Link makes it secure and easy for users to connect their bank accounts to Plaid. Note that these instructions cover Link on the web. For instructions on using Link within mobile apps, see the [Link documentation](https://plaid.com/docs/link/).

Using Link, we will create a Plaid Item, which is a Plaid term for a login at a financial institution. An Item is not the same as a financial institution account, although every account will be associated with an Item. For example, if a user has one login at their bank that allows them to access both their checking account and their savings account, a single Item would be associated with both of those accounts. If you want to customize Link's look and feel, you can do so from the [Dashboard](https://dashboard.plaid.com/link).

Before initializing Link, you will need to create a new link\_token on the server side of your application. A link\_token is a short-lived, one-time use token that is used to authenticate your app with Link. You can create one using the [**/link/token/create**](https://plaid.com/docs/api/tokens/#linktokencreate) endpoint. Then, on the client side of your application, you'll need to initialize Link with the link\_token that you just created.

The [**/link/token/create**](https://plaid.com/docs/api/tokens/#linktokencreate) sample code below will create an Item with a maximum of 90 days of transaction history. To request more, set the transactions.days\_requested parameter in the [**/link/token/create**](https://plaid.com/docs/api/tokens/#linktokencreate) request.

##### [Create a link\_token](https://plaid.com/docs/transactions/add-to-app/" \l "create-a-link_token)

Select group for content switcher

Current librariesLegacy libraries

**Curl**

**Select Language**

**Copy**

1curl -X POST https://sandbox.plaid.com/link/token/create \

2-H 'Content-Type: application/json' \

3-d '{

4 "client\_id": "${PLAID\_CLIENT\_ID}",

5 "secret": "${PLAID\_SECRET}",

6 "client\_name": "Plaid Test App",

7 "user": { "client\_user\_id": "${UNIQUE\_USER\_ID}" },

8 "products": ["${PRODUCT}"],

9 "country\_codes": ["US"],

10 "language": "en",

11 "webhook": "https://webhook.example.com",

12 "redirect\_uri": "https://domainname.com/oauth-page.html",

13}'

##### [Install Link dependency](https://plaid.com/docs/transactions/add-to-app/" \l "install-link-dependency)

index.html

**Curl**

**Select Language**

**Copy**

1<head>

2 <title>Connect a bank</title>

3 <script src="https://cdn.plaid.com/link/v2/stable/link-initialize.js"></script>

4</head>

##### [Configure the client-side Link handler](https://plaid.com/docs/transactions/add-to-app/" \l "configure-the-client-side-link-handler)

app.js

**Copy**

1const linkHandler = Plaid.create({

2 token: (await $.post('/create\_link\_token)).link\_token,

3 onSuccess: (public\_token, metadata) => {

4 // Send the public\_token to your app server.

5 $.post('/exchange\_public\_token', {

6 public\_token: public\_token,

7 });

8 },

9 onExit: (err, metadata) => {

10 // Optionally capture when your user exited the Link flow.

11 // Storing this information can be helpful for support.

12 },

13 onEvent: (eventName, metadata) => {

14 // Optionally capture Link flow events, streamed through

15 // this callback as your users connect an Item to Plaid.

16 },

17});

18

19linkHandler.open();

#### [Get a persistent access token](https://plaid.com/docs/transactions/add-to-app/" \l "get-a-persistent-access-token)

Next, on the server side, we need to exchange our public\_token for an access\_token and item\_id. The access\_token will allow us to make authenticated calls to the Plaid API. Doing so is as easy as calling the [**/item/public\_token/exchange**](https://plaid.com/docs/api/tokens/#itempublic_tokenexchange) endpoint from our server-side handler. We'll use the client library we configured earlier to make the API call.

Save the access\_token and item\_id in a secure datastore, as they’re used to access Item data and identify webhooks, respectively. The access\_token will remain valid unless you actively chose to expire it via rotation or remove the corresponding Item via [**/item/remove**](https://plaid.com/docs/api/items/#itemremove). The access\_token should be stored securely, and never in client-side code. A public\_token is a one-time use token with a lifetime of 30 minutes, so there is no need to store it.

Select group for content switcher

Current librariesLegacy libraries

**Curl**

**Select Language**

**Copy**

1curl -X POST https://sandbox.plaid.com/item/public\_token/exchange \

2-H 'Content-Type: application/json' \

3-d '{

4 "client\_id": "${PLAID\_CLIENT\_ID}",

5 "secret": "${PLAID\_SECRET}",

6 "public\_token": "public-sandbox-12345678-abcd-1234-abcd-1234567890ab"

7}'

#### [Fetching transaction data](https://plaid.com/docs/transactions/add-to-app/" \l "fetching-transaction-data)

Now that the authentication step is out of the way, we can begin using authenticated endpoints from the Plaid API and fetch transaction data using the [**/transactions/sync**](https://plaid.com/docs/api/products/transactions/#transactionssync) endpoint.

The [**/transactions/sync**](https://plaid.com/docs/api/products/transactions/#transactionssync) endpoint is used to both initialize your view of transactions data, and keep you current with any changes that have occurred. When you first call it on an Item with no cursor parameter, transactions data available at that time is returned. If more updates are available than requested with the count parameter (maximum of 500), has\_more will be set to true, indicating the endpoint should be called again, using the next\_cursor from the previous response in the cursor field of the next request, to receive another page of data. After successfully pulling all currently available pages, you can store the cursor for later requests, allowing Plaid to send you new updates from when you last queried the endpoint.

Note that if you encounter an error during pagination, it's important to restart the pagination loop from the beginning. For more details, see the documentation for [**TRANSACTIONS\_SYNC\_MUTATION\_DURING\_PAGINATION**](https://plaid.com/docs/errors/transactions/#transactions_sync_mutation_during_pagination). For sample code for handling the error, see the [**/transactions/sync**](https://plaid.com/docs/api/products/transactions/#transactionssync) API reference.

Typically, the first 30 days of transaction history is available to be fetched almost immediately, but full transaction history may take a minute or more to become available. If you get an empty response when calling [**/transactions/sync**](https://plaid.com/docs/api/products/transactions/#transactionssync) shortly after linking an Item, it's likely that the first 30 days of transaction history has not yet been pulled. You will need to call the endpoint when the data is pulled. Similarly, if you only get the first 30 days of transaction history, you will need to wait until it is complete, and call the endpoint again.

To be notified whenever additional data becomes available, see [Transaction webhooks](https://plaid.com/docs/transactions/webhooks/).

/transactions/sync

**Curl**

**Select Language**

**Copy**

1curl -X POST https://sandbox.plaid.com/transactions/sync \

2-H 'Content-Type: application/json' \

3-d '{

4 "client\_id": "${PLAID\_CLIENT\_ID}"

5 "secret": "${PLAID\_SECRET}"

6 "access\_token": String,

7 "cursor": String,

8 "count": 250

9}'

#### [Updating transaction data](https://plaid.com/docs/transactions/add-to-app/" \l "updating-transaction-data)

After your initial [**/transactions/sync**](https://plaid.com/docs/api/products/transactions/#transactionssync) request, you may want your application to be notified when any transactions are added, removed, or modified in order to immediately fetch them from [**/transactions/sync**](https://plaid.com/docs/api/products/transactions/#transactionssync). To learn how, see [Transaction Webhooks](https://plaid.com/docs/transactions/webhooks/).

# Transactions webhooks

#### Listen for Transaction webhooks to learn when transactions are ready for retrieval or when transactions have been updated.

#### [Introduction](https://plaid.com/docs/transactions/webhooks/" \l "introduction)

Webhooks are a useful part of the Transactions product that notifies you when Plaid has new or updated transaction information. This guide will explain how to use webhooks to make sure you have up-to-date transaction history.

#### [Configuring Link for transactions webhooks](https://plaid.com/docs/transactions/webhooks/" \l "configuring-link-for-transactions-webhooks)

Before you can listen to webhooks, you must first set up an endpoint and tell Plaid where to find it. To tell Plaid where to send its webhooks, send your webhook endpoint URL as an optional argument via the webhook parameter to [**/link/token/create**](https://plaid.com/docs/api/tokens/#linktokencreate).

You must also initialize your Item with Transactions by including transactions in the products array provided to [**/link/token/create**](https://plaid.com/docs/api/tokens/#linktokencreate). If you do not do this, Plaid will not attempt to retrieve any transactions for your Item until after [**/transactions/sync**](https://plaid.com/docs/api/products/transactions/#transactionssync) or [**/transactions/get**](https://plaid.com/docs/api/products/transactions/#transactionsget) is called for the first time. For more information, see [Choosing how to initialize products](https://plaid.com/docs/link/initializing-products/).

#### [Integrating the update notification webhook](https://plaid.com/docs/transactions/webhooks/" \l "integrating-the-update-notification-webhook)

After [**/transactions/sync**](https://plaid.com/docs/api/products/transactions/#transactionssync) is called for the first time on an Item, [**SYNC\_UPDATES\_AVAILABLE**](https://plaid.com/docs/api/products/transactions/#sync_updates_available) webhooks will begin to be sent to the configured destination endpoint.

This webhook will fire whenever any change has happened to the Item's transactions. The changes can then be retrieved by calling [**/transactions/sync**](https://plaid.com/docs/api/products/transactions/#transactionssync) with the cursor from your last sync call to this Item.

If at least 30 days of history is available with an update, the initial\_update\_complete parameter in the body of the SYNC\_UPDATES\_AVAILABLE webhook will be true. Similarly, historical\_update\_complete will be true if the full history (up to 24 months) is available.

For a real-life example that illustrates how to handle this webhook, see [handleTransactionsWebhook.js](https://github.com/plaid/pattern/blob/master/server/webhookHandlers/handleTransactionsWebhook.js), which contains the webhook handling code for the Node-based [Plaid Pattern](https://github.com/plaid/pattern) app.

#### [Forcing transactions refresh](https://plaid.com/docs/transactions/webhooks/" \l "forcing-transactions-refresh)

Sometimes, checking for transactions a few times a day is not good enough. For example, you might want to build a refresh button in your app that allows your user to check for updated transactions on-demand. To accomplish this, you can use the [**/transactions/refresh**](https://plaid.com/docs/api/products/transactions/#transactionsrefresh) product. After a successful call to [**/transactions/refresh**](https://plaid.com/docs/api/products/transactions/#transactionsrefresh), if there are new updates, SYNC\_UPDATES\_AVAILABLE will be fired (along with DEFAULT\_UPDATE and, if applicable, TRANSACTIONS\_REMOVED).

#### [Instructions for integrations using /transactions/get](https://plaid.com/docs/transactions/webhooks/" \l "instructions-for-integrations-using-transactionsget)

The content in this section and below applies only to existing integrations using the [**/transactions/get**](https://plaid.com/docs/api/products/transactions/#transactionsget) endpoint. It is recommended that any new integrations use [**/transactions/sync**](https://plaid.com/docs/api/products/transactions/#transactionssync) instead of [**/transactions/get**](https://plaid.com/docs/api/products/transactions/#transactionsget), for easier and simpler handling of transaction state changes. For information on migrating an existing [**/transactions/get**](https://plaid.com/docs/api/products/transactions/#transactionsget) integration to [**/transactions/sync**](https://plaid.com/docs/api/products/transactions/#transactionssync), see the [Transactions Sync migration guide](https://plaid.com/docs/transactions/sync-migration/).

When you first connect an Item in Link, transactions data will not immediately be available. [**INITIAL\_UPDATE**](https://plaid.com/docs/api/products/transactions/#initial_update) and [**HISTORICAL\_UPDATE**](https://plaid.com/docs/api/products/transactions/#historical_update) are both webhooks that fire shortly after an Item has been initially linked and initialized with the Transactions product. These webhooks will let you know when your transactions are ready. INITIAL\_UPDATE fires first, after Plaid has successfully pulled 30 days of transactions for an Item. The HISTORICAL\_UPDATE webhook fires next, once all historical transactions data is available. INITIAL\_UPDATE typically fires within 10 seconds, and HISTORICAL\_UPDATE within 1 minute, although these webhooks may take 2 minutes or more. The time required for the webhooks to fire will depend on the institution, as well as on the number of transactions being processed.

If you attempt to call [**/transactions/get**](https://plaid.com/docs/api/products/transactions/#transactionsget) before INITIAL\_UPDATE has fired, you will get a [**PRODUCT\_NOT\_READY**](https://plaid.com/docs/errors/item/#product_not_ready) error. If you attempt to call [**/transactions/get**](https://plaid.com/docs/api/products/transactions/#transactionsget) after INITIAL\_UPDATE has fired, but before HISTORICAL\_UPDATE has fired, you will only be able to receive the last 30 days of transaction data. If you did not initialize the Item with Transactions, your first call to [**/transactions/get**](https://plaid.com/docs/api/products/transactions/#transactionsget) will result in a PRODUCT\_NOT\_READY error and kick off the process of readying transactions. You can then listen for the INITIAL\_UPDATE or HISTORICAL\_UPDATE webhooks to begin receiving transactions.

**Updating transactions**

Plaid fires two types of webhooks that provide information about changes to transaction data: [**DEFAULT\_UPDATE**](https://plaid.com/docs/api/products/transactions/#default_update) and [**TRANSACTIONS\_REMOVED**](https://plaid.com/docs/api/products/transactions/#transactions_removed).

**Adding new transactions**

The [**DEFAULT\_UPDATE**](https://plaid.com/docs/api/products/transactions/#default_update) webhook fires when new transactions are available. Typically, Plaid will check for transactions at a frequency ranging from one to four times per day, depending on factors such as the institution and account type. If new transactions are available, the DEFAULT\_UPDATE webhook will fire.

To reflect up-to-date transactions for a user in your app, handle the DEFAULT\_UPDATE webhook by fetching more transactions. We recommend fetching about 7-14 days of transactions in response to DEFAULT\_UPDATE. This is typically enough history to ensure that you haven't missed any transactions, but not so much that performance or rate limiting is likely to be a problem.

Once you've fetched these transactions, you will need to identify which transactions are new and which are duplicates of existing data that you have. You should not rely on the number in the webhook's new\_transactions field to identify duplicates, since it can be unreliable. For example, new transactions may arrive between your receipt of the webhook and your call to [**/transactions/get**](https://plaid.com/docs/api/products/transactions/#transactionsget). Instead, compare the transaction\_id field of each newly fetched transaction to the transaction\_id fields of your existing transactions, and skip the ones that you already have. For an example, see [Plaid Pattern](https://github.com/plaid/pattern/blob/master/server/webhookHandlers/handleTransactionsWebhook.js#L176).

**Removing stale transactions**

The [**TRANSACTIONS\_REMOVED**](https://plaid.com/docs/api/products/transactions/#transactions_removed) webhook fires when transactions have been removed. The most common reason for this is in the case of pending transactions. In general, transactions start out as pending transactions and then move to the posted state one to two business days later. When Plaid detects that a transaction has moved from pending to posted state, the pending transaction as returned by [**/transactions/get**](https://plaid.com/docs/api/products/transactions/#transactionsget) is not modified. Instead, the pending transaction is removed, and a new transaction is added, representing the posted transaction. For a detailed explanation of pending and posted transactions and how they are handled by Plaid, see [Transaction states](https://plaid.com/docs/transactions/transactions-data/).

Pending transactions can also be removed when they are canceled by the bank or payment processor. A transaction may be removed if its details are changed by the bank so extensively that Plaid can no longer recognize the new and old versions of the transaction as being the same (e.g., a transaction amount and description both being changed simultaneously). In this case, the old transaction will be deleted and a new transaction with the new details will be added. This "transaction churn" can affect both pending and posted transactions.

The TRANSACTIONS\_REMOVED webhook contains the transaction IDs of the removed transactions, which you can use to identify and remove the corresponding transactions in your own application to avoid presenting duplicated or inaccurate data. If you encounter any problems with the webhook, you can also manually query transaction history for deleted transactions using logic similar to that recommended for handling the DEFAULT\_UPDATE webhook.

# Transaction states

#### Learn about the differences between pending and posted transactions

#### [Pending and posted transactions](https://plaid.com/docs/transactions/transactions-data/" \l "pending-and-posted-transactions)

There are two types of transactions: pending and posted. A transaction begins its life as a pending transaction, then becomes posted once the funds have actually been transferred. It typically takes about one to five business days for a transaction to move from pending to posted, although it can take up to fourteen days in rare situations.

When a transaction posts, the transition from a pending to posted transaction will be represented through the [**/transactions/sync**](https://plaid.com/docs/api/products/transactions/#transactionssync) endpoint with the pending transaction's id in the removed field of the response and the new posted transaction in the added section of the response -- note that these aren't guaranteed to be in the same page, but should happen within the same overall update. If Plaid matches the pending transaction to the new posted transaction, the pending transaction's id will be marked in the pending\_transaction\_id of the posted transaction.

Example of /transactions/sync response for transaction posting

**Copy**

1{

2 "added": [{

3 "transaction\_id": "lPNjeW1nR6CDn5okmGQ6hEpMo4lLNoSrzqDje",

4 "pending\_transaction\_id": "BxBXxLj1m4HMXBm9WZZmCWVbPjX16EHwv99vp",

5 "pending": false,

6 "name": "Apple Store",

7 "amount": 2307.21

8 /\* ... \*/

9 }],

10 "removed": [{

11 "transaction\_id": "BxBXxLj1m4HMXBm9WZZmCWVbPjX16EHwv99vp",

12 "pending": true,

13 "name": "Apple Store",

14 "amount": 2307.21

15 /\* ... \*/

16 }],

17 "modified": []

18 /\* ... \*/

19}

The pending and posted versions of a transaction may not necessarily share the same details: their name and amount may change. For example, the pending charge for a meal at a restaurant may not include a tip, but the posted version will include the final amount spent, including the tip. In some cases, a pending transaction may not convert to a posted transaction at all and will simply disappear; this can happen, for example, if the pending transaction was used as an "authorization hold," which is a sort of a deposit for a potential future transaction, frequently used by gas stations, hotels, and rental-car companies. Pending transactions are short-lived and frequently altered or removed by the institution before finally settling as a posted transaction.

Note that while transactions will rarely change once they have posted, a posted transaction cannot necessarily be considered immutable. For example, a refund or a recategorization of a transaction by the institution could cause a previously posted transaction to change. This is why it's important to apply all modified, added, and removed updates surfaced through [**/transactions/sync**](https://plaid.com/docs/api/products/transactions/#transactionssync) in order to maintain consistency with the underlying account data.

#### [Example code in Plaid Pattern](https://plaid.com/docs/transactions/transactions-data/" \l "example-code-in-plaid-pattern)

For a real-life example, see [update\_transactions.js](https://github.com/plaid/pattern/blob/master/server/update_transactions.js). This file demonstrates code for handling transaction states in the Node-based [Plaid Pattern](https://github.com/plaid/pattern) sample app.

#### [Transaction state changes with /transactions/get](https://plaid.com/docs/transactions/transactions-data/" \l "transaction-state-changes-with-transactionsget)

The content in this section and below applies only to existing integrations using the [**/transactions/get**](https://plaid.com/docs/api/products/transactions/#transactionsget) endpoint. It is recommended that any new integrations use [**/transactions/sync**](https://plaid.com/docs/api/products/transactions/#transactionssync) instead of [**/transactions/get**](https://plaid.com/docs/api/products/transactions/#transactionsget), for easier and simpler handling of Transaction state changes.

**Reconciling transactions**

[**/transactions/get**](https://plaid.com/docs/api/products/transactions/#transactionsget) returns both pending and posted transactions; however, some institutions do not provide pending transaction data and will only supply posted transactions. The pending boolean field in the transaction indicates whether the transaction is pending or posted.

Plaid does not model the transition of a pending to posted transaction as a state change for an existing transaction; instead, the posted transaction is a new transaction with a pending\_transaction\_id field that matches it to a corresponding pending transaction. When a pending transaction is converted to a posted transaction, Plaid removes the pending transaction, sends a [**TRANSACTIONS\_REMOVED**](https://plaid.com/docs/api/products/transactions/#transactions_removed) webhook, and returns the new, posted transaction. The posted transaction will have a pending\_transaction\_id field whose value is the transaction\_id of the now-removed pending transaction. The posted transaction’s date will reflect the date the transaction was posted, which may differ from the date on which the transaction actually occurred.

In some rare cases, Plaid will fail to match a posted transaction to its pending counterpart. On such occasions, the posted transaction will be returned without a pending\_transaction\_id field, and its pending transaction is removed.

**Handling pending and posted transactions**

To manage the movement of a transaction from pending to posted state, you will need to handle the TRANSACTIONS\_REMOVED webhook to identify the removed transactions, then delete them from your records. For detailed instructions, see [Transactions webhooks](https://plaid.com/docs/api/products/transactions/#transactions_removed).

# Transactions Sync migration guide

#### Learn how to migrate from the /transactions/get endpoint to the /transactions/sync endpoint

#### [Overview](https://plaid.com/docs/transactions/sync-migration/" \l "overview)

[**/transactions/sync**](https://plaid.com/docs/api/products/transactions/#transactionssync) is a newer endpoint that replaces [**/transactions/get**](https://plaid.com/docs/api/products/transactions/#transactionsget) and provides a simpler and easier model for managing transactions updates. While [**/transactions/get**](https://plaid.com/docs/api/products/transactions/#transactionsget) provides all transactions within a date range, [**/transactions/sync**](https://plaid.com/docs/api/products/transactions/#transactionssync) instead uses a cursor to provide all new, modified, and removed transactions that occurred since your previous request. With this cursor-based pagination, you do not need to worry about making redundant API calls to avoid missing transactions. Updates returned by [**/transactions/sync**](https://plaid.com/docs/api/products/transactions/#transactionssync) can be patched into your database, allowing you to avoid a complex transaction reconciliation process or having to keep track of which updates have already been applied.

This guide outlines how to update your existing [**/transactions/get**](https://plaid.com/docs/api/products/transactions/#transactionsget) integration to use the [**/transactions/sync**](https://plaid.com/docs/api/products/transactions/#transactionssync) endpoint and simplify your Plaid integration.

Looking for an example in code? Check out [Pattern on GitHub](https://github.com/plaid/pattern) for a complete, best-practice implementation of the Transactions Sync API within a sample app.

#### [Update your client library](https://plaid.com/docs/transactions/sync-migration/" \l "update-your-client-library)

If you are using client libraries, you may need to update your current library to use [**/transactions/sync**](https://plaid.com/docs/api/products/transactions/#transactionssync). The following are the minimum Plaid client library versions required to support [**/transactions/sync**](https://plaid.com/docs/api/products/transactions/#transactionssync) for each language:

* Python: 9.4.0
* Node: 10.4.0
* Ruby: 15.5.0
* Java: 11.3.0
* Go: 3.4.0

Detailed upgrade notes are language-specific may be found in the README and Changelog of the specific library. See the library's repo on the [Plaid GitHub](https://github.com/plaid) for more information.

#### [Update callsites and pagination logic](https://plaid.com/docs/transactions/sync-migration/" \l "update-callsites-and-pagination-logic)

Replace all instances of [**/transactions/get**](https://plaid.com/docs/api/products/transactions/#transactionsget) with [**/transactions/sync**](https://plaid.com/docs/api/products/transactions/#transactionssync). [**/transactions/sync**](https://plaid.com/docs/api/products/transactions/#transactionssync) has a slightly different call signature from [**/transactions/get**](https://plaid.com/docs/api/products/transactions/#transactionsget) and does not have the count parameter inside the options object and uses a cursor instead of a start\_date and end\_date. Pagination logic is also different and relies on the has\_more flag instead of the transactions\_count value. Note that when requesting paginated updates with [**/transactions/sync**](https://plaid.com/docs/api/products/transactions/#transactionssync), unlike when using [**/transactions/get**](https://plaid.com/docs/api/products/transactions/#transactionsget), it is important to retrieve all available updates before persisting the transactions updates to your database.

For copy-and-pastable examples of how to call [**/transactions/sync**](https://plaid.com/docs/api/products/transactions/#transactionssync), including complete pagination logic, see the API reference code samples for [**/transactions/sync**](https://plaid.com/docs/api/products/transactions/#transactionssync).

If a call to [**/transactions/sync**](https://plaid.com/docs/api/products/transactions/#transactionssync) fails when retrieving a paginated update as a result of the TRANSACTIONS\_SYNC\_MUTATION\_DURING\_PAGINATION error, the entire pagination request loop must be restarted beginning with the cursor for the first page of the update, rather than retrying only the single request that failed.

#### [Update callsites for account and Item data](https://plaid.com/docs/transactions/sync-migration/" \l "update-callsites-for-account-and-item-data)

Unlike [**/transactions/get**](https://plaid.com/docs/api/products/transactions/#transactionsget), [**/transactions/sync**](https://plaid.com/docs/api/products/transactions/#transactionssync) does not return an account object or Item object. If your app relies on getting account data, such as balance, from the [**/transactions/get**](https://plaid.com/docs/api/products/transactions/#transactionsget) call, use [**/accounts/get**](https://plaid.com/docs/api/accounts/#accountsget) instead to retrieve this information. If it relies on Item data, such as Item health status, use [**/item/get**](https://plaid.com/docs/api/items/#itemget).

#### [Update webhook handlers](https://plaid.com/docs/transactions/sync-migration/" \l "update-webhook-handlers)

When using [**/transactions/sync**](https://plaid.com/docs/api/products/transactions/#transactionssync), you should not listen for the webhooks HISTORICAL\_UPDATE, DEFAULT\_UPDATE, INITIAL\_UPDATE, or TRANSACTIONS\_REMOVED. While these webhooks will still be sent in order to maintain backwards compatibility, they are not required for the business logic used by [**/transactions/sync**](https://plaid.com/docs/api/products/transactions/#transactionssync).

Instead, update your webhook handlers to listen for the [**SYNC\_UPDATES\_AVAILABLE**](https://plaid.com/docs/api/products/transactions/#sync_updates_available) webhook and to call [**/transactions/sync**](https://plaid.com/docs/api/products/transactions/#transactionssync) when this webhook is received.

#### [Update initial call trigger](https://plaid.com/docs/transactions/sync-migration/" \l "update-initial-call-trigger)

Unlike the [**/transactions/get**](https://plaid.com/docs/api/products/transactions/#transactionsget) webhooks, the SYNC\_UPDATES\_AVAILABLE webhook will not be fired for an Item unless [**/transactions/sync**](https://plaid.com/docs/api/products/transactions/#transactionssync) has been called at least once for that Item. For this reason, you must call [**/transactions/sync**](https://plaid.com/docs/api/products/transactions/#transactionssync) at least once before any sync webhook is received. After that point, rely on the SYNC\_UPDATES\_AVAILABLE webhook.

Unlike [**/transactions/get**](https://plaid.com/docs/api/products/transactions/#transactionsget), [**/transactions/sync**](https://plaid.com/docs/api/products/transactions/#transactionssync) will not return the PRODUCT\_NOT\_READY error if transactions data is not yet ready when [**/transactions/sync**](https://plaid.com/docs/api/products/transactions/#transactionssync) is first called. Instead, you will receive a response with no transactions and a null cursor. Even if no transactions data is available, this call will still initialize the SYNC\_UPDATES\_AVAILABLE webhook, and it will fire once data becomes available.

The first call to [**/transactions/sync**](https://plaid.com/docs/api/products/transactions/#transactionssync) once historical updates are available will often have substantially higher latency (up to 8x) than the equivalent call in a [**/transactions/get**](https://plaid.com/docs/api/products/transactions/#transactionsget)-based implementation. Depending on your application's logic, you may need to adjust user-facing messaging or hard-coded timeout settings.

#### [Update transaction reconciliation logic](https://plaid.com/docs/transactions/sync-migration/" \l "update-transaction-reconciliation-logic)

The response to [**/transactions/sync**](https://plaid.com/docs/api/products/transactions/#transactionssync) includes the patches you will need to apply in the added, removed, and modified arrays within its response. You should apply these to your transactions records. Any additional logic required to fetch or reconcile transactions data can be removed.

#### [Migrating existing Items](https://plaid.com/docs/transactions/sync-migration/" \l "migrating-existing-items)

You likely already have transactions stored for existing Items. If you onboard an existing Item onto [**/transactions/sync**](https://plaid.com/docs/api/products/transactions/#transactionssync) with "cursor": "" in the request body, the endpoint will return all historical transactions data associated with that Item up until the time of the API call (as "adds"). You may reconcile these with your stored copy of transactions to ensure that it reflects the the Item's true state.

If you have a large number of Items to update, this reconciliation process may be slow and generate excessive system load. One other option for onboarding existing Items onto [**/transactions/sync**](https://plaid.com/docs/api/products/transactions/#transactionssync) is using "cursor": "now" in the request body. The endpoint will return a response containing no transaction updates, but only a cursor that will allow you to retrieve all transactions updates associated with that Item going forward, after the time of the API call. Accordingly, you should ensure that your local copy of transactions for an Item is up-to-date at the time you call [**/transactions/sync**](https://plaid.com/docs/api/products/transactions/#transactionssync) with "cursor": "now" for it, or else any transaction updates that occurred between the time that you last pulled fresh data and the time of your [**/transactions/sync**](https://plaid.com/docs/api/products/transactions/#transactionssync) call may be missing.

"cursor": "now" will work exactly like a cursor that was found by starting with "cursor": "" and paginating through all updates, with the only difference being that a transaction created before, but modified after, those requests would be returned as "added" if using "cursor": "now", and "modified" if using "cursor": "".

If you ever want to completely rebuild your local copy of transactions for an Item previously onboarded with "cursor": "now", you may still do so with "cursor": "".

Note that we strongly recommend that this cursor only be used with Items for which you've already used with [**/transactions/get**](https://plaid.com/docs/api/products/transactions/#transactionsget), and not any new Items, which should always be onboarded with "cursor": "".

#### [Test your integration](https://plaid.com/docs/transactions/sync-migration/" \l "test-your-integration)

You can perform basic testing of your integration's business logic in Sandbox, using the [**/sandbox/item/fire\_webhook**](https://plaid.com/docs/api/sandbox/#sandboxitemfire_webhook) endpoint to simulate SYNC\_UPDATES\_AVAILABLE. If this testing succeeds, you should then test your integration with internal test accounts in Development or Production before releasing it to your full Production userbase.

#### [Example code](https://plaid.com/docs/transactions/sync-migration/" \l "example-code)

For a full working example of a Plaid-powered app using [**/transactions/sync**](https://plaid.com/docs/api/products/transactions/#transactionssync), see [Plaid Pattern](https://github.com/plaid/pattern/tree/master/server).

# Introduction to Liabilities

#### Access data for student loans, mortgages, and credit cards using the Liabilities product.

#### [API Reference](https://plaid.com/docs/api/products/liabilities/)

[View Liabilities requests, responses, and example code](https://plaid.com/docs/api/products/liabilities/)

**[View Liabilities API](https://plaid.com/docs/api/products/liabilities/)**

#### [Quickstart](https://plaid.com/docs/quickstart/)

[Learn about Plaid's key concepts and run starter code](https://plaid.com/docs/quickstart/)

**[Get started](https://plaid.com/docs/quickstart/)**

#### [Overview](https://plaid.com/docs/liabilities/" \l "overview)

Plaid's Liabilities product allows you to access information about a user's debts. A common application is personal financial management tools to help customers manage or refinance debt.

#### [Liabilities data](https://plaid.com/docs/liabilities/" \l "liabilities-data)

With Liabilities, you can view account information for credit cards, PayPal credit accounts, student loans, and mortgages in the US. Available information includes balance, next payment date and amount, loan terms such as duration and interest rate, and originator information such as the origination date and initial loan amount. Liabilities data is refreshed approximately once per day, and the latest data can be accessed by calling [**/liabilities/get**](https://plaid.com/docs/api/products/liabilities/#liabilitiesget).

Note that liabilities data does not contain detailed transaction history for credit card accounts. For credit card account transactions, use the [Transactions](https://plaid.com/docs/transactions/) product.

Sample Liabilities data: credit card account

**Copy**

1"credit": [

2 {

3 "account\_id": "dVzbVMLjrxTnLjX4G66XUp5GLklm4oiZy88yK",

4 "aprs": [

5 {

6 "apr\_percentage": 15.24,

7 "apr\_type": "balance\_transfer\_apr",

8 "balance\_subject\_to\_apr": 1562.32,

9 "interest\_charge\_amount": 130.22

10 },

11 {

12 "apr\_percentage": 27.95,

13 "apr\_type": "cash\_apr",

14 "balance\_subject\_to\_apr": 56.22,

15 "interest\_charge\_amount": 14.81

16 },

17 {

18 "apr\_percentage": 12.5,

19 "apr\_type": "purchase\_apr",

20 "balance\_subject\_to\_apr": 157.01,

21 "interest\_charge\_amount": 25.66

22 },

23 {

24 "apr\_percentage": 0,

25 "apr\_type": "special",

26 "balance\_subject\_to\_apr": 1000,

27 "interest\_charge\_amount": 0

28 }

29 ],

30 "is\_overdue": false,

31 "last\_payment\_amount": 168.25,

32 "last\_payment\_date": "2019-05-22",

33 "last\_statement\_issue\_date": "2019-05-28",

34 "last\_statement\_balance": 1708.77,

35 "minimum\_payment\_amount": 20,

36 "next\_payment\_due\_date": "2020-05-28"

37 }

38]

Sample Liabilities data: student loan account

**Copy**

1"student": [

2 {

3 "account\_id": "Pp1Vpkl9w8sajvK6oEEKtr7vZxBnGpf7LxxLE",

4 "account\_number": "4277075694",

5 "disbursement\_dates": [

6 "2002-08-28"

7 ],

8 "expected\_payoff\_date": "2032-07-28",

9 "guarantor": "DEPT OF ED",

10 "interest\_rate\_percentage": 5.25,

11 "is\_overdue": false,

12 "last\_payment\_amount": 138.05,

13 "last\_payment\_date": "2019-04-22",

14 "last\_statement\_issue\_date": "2019-04-28",

15 "loan\_name": "Consolidation",

16 "loan\_status": {

17 "end\_date": "2032-07-28",

18 "type": "repayment"

19 },

20 "minimum\_payment\_amount": 25,

21 "next\_payment\_due\_date": "2019-05-28",

22 "origination\_date": "2002-08-28",

23 "origination\_principal\_amount": 25000,

24 "outstanding\_interest\_amount": 6227.36,

25 "payment\_reference\_number": "4277075694",

26 "pslf\_status": {

27 "estimated\_eligibility\_date": "2021-01-01",

28 "payments\_made": 200,

29 "payments\_remaining": 160

30 },

31 "repayment\_plan": {

32 "description": "Standard Repayment",

33 "type": "standard"

34 },

35 "sequence\_number": "1",

36 "servicer\_address": {

37 "city": "San Matias",

38 "country": "US",

39 "postal\_code": "99415",

40 "region": "CA",

41 "street": "123 Relaxation Road"

42 },

43 "ytd\_interest\_paid": 280.55,

44 "ytd\_principal\_paid": 271.65

45 }

46]

Sample Liabilities data: mortgage account

**Copy**

1"mortgage": [

2 {

3 "account\_id": "BxBXxLj1m4HMXBm9WZJyUg9XLd4rKEhw8Pb1J",

4 "account\_number": "3120194154",

5 "current\_late\_fee": 25.0,

6 "escrow\_balance": 3141.54,

7 "has\_pmi": true,

8 "has\_prepayment\_penalty": true,

9 "interest\_rate": {

10 "percentage": 3.99,

11 "type": "fixed",

12 },

13 "last\_payment\_amount": 3141.54,

14 "last\_payment\_date": "2019-08-01",

15 "loan\_term": "30 year",

16 "loan\_type\_description": "conventional",

17 "maturity\_date": "2045-07-31",

18 "next\_monthly\_payment": 3141.54,

19 "next\_payment\_due\_date": "2019-11-15",

20 "origination\_date": "2015-08-01",

21 "origination\_principal\_amount": 425000,

22 "past\_due\_amount": 2304,

23 "property\_address": {

24 "city": "Malakoff",

25 "country": "US",

26 "postal\_code": "14236",

27 "region": "NY",

28 "street": "2992 Cameron Road",

29 }

30 "ytd\_interest\_paid": 12300.4,

31 "ytd\_principal\_paid": 12340.5

32 }

33]

#### [Payment history](https://plaid.com/docs/liabilities/" \l "payment-history)

The information returned by a [**/liabilities/get**](https://plaid.com/docs/api/products/liabilities/#liabilitiesget) request contains recent payment information, such as the date and amount of the most recent payment. To view further payment history, you can use Plaid's [Transactions](https://plaid.com/docs/transactions/) product.

#### [Liabilities webhooks](https://plaid.com/docs/liabilities/" \l "liabilities-webhooks)

Plaid checks for updated Liabilities data approximately once per day and uses webhooks to inform you of any changes so you can keep your app up to date. For more detail on how to listen and respond to these webhooks, see [Liabilities webhooks](https://plaid.com/docs/api/products/liabilities/#webhooks).

#### [Testing Liabilities](https://plaid.com/docs/liabilities/" \l "testing-liabilities)

Liabilities can be tested in Sandbox or Development without any additional permissions.

Plaid also provides a [GitHub repo](https://github.com/plaid/sandbox-custom-users/) with test data for testing student loan accounts in Sandbox. For more information on configuring custom Sandbox data, see [Configuring the custom user account](https://plaid.com/docs/sandbox/user-custom/#configuring-the-custom-user-account).

#### [Next steps](https://plaid.com/docs/liabilities/" \l "next-steps)

To get started building with Liabilities, see [Add Liabilities to your App](https://plaid.com/docs/liabilities/add-to-app/).

If you're ready to launch to Production, see the Launch checklist.

# Add Liabilities to your app

#### Use Liabilities to fetch data for credit, student loan, and mortgage accounts

In this guide, we'll start from scratch and walk through how to use [**Liabilities**](https://plaid.com/docs/api/products/liabilities/) to retrieve information about credit card, PayPal credit, student loan, and mortgage accounts. If you are already familiar with using Plaid and are set up to make calls to the Plaid API, make sure to note that Liabilities supports the use of [account subtype filtering](https://plaid.com/docs/liabilities/add-to-app/#create-an-item-in-link); you can skip then ahead to [Fetching Liabilities data](https://plaid.com/docs/liabilities/add-to-app/#fetching-liabilities-data).

#### [Get Plaid API keys and complete application profile](https://plaid.com/docs/liabilities/add-to-app/" \l "get-plaid-api-keys-and-complete-application-profile)

If you don't already have one, you'll need to [create a Plaid developer account](https://dashboard.plaid.com/signup). After creating your account, you can find your [API keys](https://dashboard.plaid.com/developers/keys) under the Team Settings menu on the Plaid Dashboard.

You will also need to complete your [application profile](https://dashboard.plaid.com/settings/company/app-branding) on the Dashboard. The information in your profile will be shared with users of your application when they manage their connection on the [Plaid Portal](https://my.plaid.com/). Your application profile must be completed before connecting to certain institutions in Production.

#### [Install and initialize Plaid libraries](https://plaid.com/docs/liabilities/add-to-app/" \l "install-and-initialize-plaid-libraries)

You can use our official server-side client libraries to connect to the Plaid API from your application:

Select group for content switcher

Current librariesLegacy libraries

Terminal

**Curl**

**Select Language**

**Copy**

1## no setup

After you've installed Plaid's client libraries, you can initialize them by passing in your client\_id, secret, and the environment you wish to connect to (Sandbox, Development, or Production). This will make sure the client libraries pass along your client\_id and secret with each request, and you won't need to explicitly include them in any other calls.

In the code samples below, you will need to replace PLAID\_CLIENT\_ID and PLAID\_SECRET with your own keys, which you can obtain from the [Dashboard](https://dashboard.plaid.com/developers/keys). These code samples also demonstrate starting up a server commonly used in each framework (such as Express or Flask).

Select group for content switcher

Current librariesLegacy libraries

**Curl**

**Select Language**

**Copy**

1## no setup

#### [Create an Item in Link](https://plaid.com/docs/liabilities/add-to-app/" \l "create-an-item-in-link)

Plaid Link is a drop-in module that provides a secure, elegant authentication flow for each institution that Plaid supports. Link makes it secure and easy for users to connect their bank accounts to Plaid. Note that these instructions cover Link on the web. For instructions on using Link within mobile apps, see the [Link documentation](https://plaid.com/docs/link/).

Using Link, we will create a Plaid Item, which is a Plaid term for a login at a financial institution. An Item is not the same as a financial institution account, although every account will be associated with an Item. For example, if a user has one login at their bank that allows them to access both their checking account and their savings account, a single Item would be associated with both of those accounts.

First, on the client side of your application, you'll need to set up and configure Link. If you want to customize Link's look and feel, you can do so from the [Dashboard](https://dashboard.plaid.com/link).

When initializing Link, you will need to specify the products you will be using in the product array.

Liabilities supports the use of the account\_filters parameter to [**/link/token/create**](https://plaid.com/docs/api/tokens/#linktokencreate), which you can use to limit the account types that will be shown in Link to either only credit card accounts, only student loan accounts, or only mortgage accounts. For student loans, use the type loan and subtype student. For credit cards, use the type credit and subtype credit card. For mortgages, use the type loan and subtype mortgage.

##### [Create a link\_token](https://plaid.com/docs/liabilities/add-to-app/" \l "create-a-link_token)

Select group for content switcher

Current librariesLegacy libraries

**Curl**

**Select Language**

**Copy**

1curl -X POST https://sandbox.plaid.com/link/token/create \

2-H 'Content-Type: application/json' \

3-d '{

4 "client\_id": "${PLAID\_CLIENT\_ID}",

5 "secret": "${PLAID\_SECRET}",

6 "client\_name": "Plaid Test App",

7 "user": { "client\_user\_id": "${UNIQUE\_USER\_ID}" },

8 "products": ["${PRODUCT}"],

9 "country\_codes": ["US"],

10 "language": "en",

11 "webhook": "https://webhook.example.com",

12 "redirect\_uri": "https://domainname.com/oauth-page.html",

13}'

##### [Install Link dependency](https://plaid.com/docs/liabilities/add-to-app/" \l "install-link-dependency)

index.html

**Curl**

**Select Language**

**Copy**

1<head>

2 <title>Connect a bank</title>

3 <script src="https://cdn.plaid.com/link/v2/stable/link-initialize.js"></script>

4</head>

##### [Configure the client-side Link handler](https://plaid.com/docs/liabilities/add-to-app/" \l "configure-the-client-side-link-handler)

app.js

**Copy**

1const linkHandler = Plaid.create({

2 token: (await $.post('/create\_link\_token')).link\_token,

3 onSuccess: (public\_token, metadata) => {

4 // Send the public\_token to your app server.

5 $.post('/exchange\_public\_token', {

6 public\_token: public\_token,

7 });

8 },

9 onExit: (err, metadata) => {

10 // Optionally capture when your user exited the Link flow.

11 // Storing this information can be helpful for support.

12 },

13 onEvent: (eventName, metadata) => {

14 // Optionally capture Link flow events, streamed through

15 // this callback as your users connect an Item to Plaid.

16 },

17});

18

19linkHandler.open();

#### [Get a persistent access\_token](https://plaid.com/docs/liabilities/add-to-app/" \l "get-a-persistent-access_token)

Next, on the server side, we need to exchange our public\_token for an access\_token and item\_id. The access\_token will allow us to make authenticated calls to the Plaid API. Doing so is as easy as calling the [**/item/public\_token/exchange**](https://plaid.com/docs/api/tokens/#itempublic_tokenexchange) endpoint from our server-side handler. We'll use the client library we configured earlier to make the API call.

Save the access\_token and item\_id in a secure datastore, as they’re used to access Item data and identify webhooks, respectively. The access\_token will remain valid unless you actively chose to expire it via rotation or remove the corresponding Item via [**/item/remove**](https://plaid.com/docs/api/items/#itemremove). The access\_token should be stored securely, and never in client-side code. A public\_token is a one-time use token with a lifetime of 30 minutes, so there is no need to store it.

Select group for content switcher

Current librariesLegacy libraries

**Curl**

**Select Language**

**Copy**

1curl -X POST https://sandbox.plaid.com/item/public\_token/exchange \

2-H 'Content-Type: application/json' \

3-d '{

4 "client\_id": "${PLAID\_CLIENT\_ID}",

5 "secret": "${PLAID\_SECRET}",

6 "public\_token": "public-sandbox-12345678-abcd-1234-abcd-1234567890ab"

7}'

#### [Fetching Liabilities data](https://plaid.com/docs/liabilities/add-to-app/" \l "fetching-liabilities-data)

Now that the authentication step is out of the way, we can begin using authenticated endpoints from the Plaid API. For more detailed information on the schema for account information returned, see [**/liabilities/get**](https://plaid.com/docs/api/products/liabilities/#liabilitiesget).

Select group for content switcher

Current librariesLegacy libraries

**Curl**

**Select Language**

**Copy**

1curl -X POST https://sandbox.plaid.com/liabilities/get \

2-H 'Content-Type: application/json' \

3-d '{

4 "client\_id": "${PLAID\_CLIENT\_ID}"

5 "secret": "${PLAID\_SECRET}"

6 "access\_token": String

7}'

Example response data is below. The data provided is for illustrative purposes; in a Production environment, it would be very unusual for a single institution to provide both credit card and student loan data.

Liabilities sample response

**Copy**

1{

2 "accounts": [

3 {

4 "account\_id": "BxBXxLj1m4HMXBm9WZZmCWVbPjX16EHwv99vp",

5 "balances": {

6 "available": 100,

7 "current": 110,

8 "iso\_currency\_code": "USD",

9 "limit": null,

10 "unofficial\_currency\_code": null

11 },

12 "mask": "0000",

13 "name": "Plaid Checking",

14 "official\_name": "Plaid Gold Standard 0% Interest Checking",

15 "subtype": "checking",

16 "type": "depository"

17 },

18 {

19 "account\_id": "dVzbVMLjrxTnLjX4G66XUp5GLklm4oiZy88yK",

20 "balances": {

21 "available": null,

22 "current": 410,

23 "iso\_currency\_code": "USD",

24 "limit": 2000,

25 "unofficial\_currency\_code": null

26 },

27 "mask": "3333",

28 "name": "Plaid Credit Card",

29 "official\_name": "Plaid Diamond 12.5% APR Interest Credit Card",

30 "subtype": "credit card",

31 "type": "credit"

32 },

33 {

34 "account\_id": "Pp1Vpkl9w8sajvK6oEEKtr7vZxBnGpf7LxxLE",

35 "balances": {

36 "available": null,

37 "current": 65262,

38 "iso\_currency\_code": "USD",

39 "limit": null,

40 "unofficial\_currency\_code": null

41 },

42 "mask": "7777",

43 "name": "Plaid Student Loan",

44 "official\_name": null,

45 "subtype": "student",

46 "type": "loan"

47 },

48 {

49 "account\_id": "BxBXxLj1m4HMXBm9WZJyUg9XLd4rKEhw8Pb1J",

50 "balances": {

51 "available": null,

52 "current": 56302.06,

53 "iso\_currency\_code": "USD",

54 "limit": null,

55 "unofficial\_currency\_code": null

56 },

57 "mask": "8888",

58 "name": "Plaid Mortgage",

59 "official\_name": null,

60 "subtype": "mortgage",

61 "type": "loan"

62 }

63 ],

64 "item": {

65 "available\_products": ["balance", "credit\_details", "investments"],

66 "billed\_products": [

67 "assets",

68 "auth",

69 "identity",

70 "liabilities",

71 "transactions"

72 ],

73 "consent\_expiration\_time": null,

74 "error": null,

75 "institution\_id": "ins\_3",

76 "item\_id": "eVBnVMp7zdTJLkRNr33Rs6zr7KNJqBFL9DrE6",

77 "webhook": "https://www.genericwebhookurl.com/webhook"

78 },

79 "liabilities": {

80 "credit": [

81 {

82 "account\_id": "dVzbVMLjrxTnLjX4G66XUp5GLklm4oiZy88yK",

83 "aprs": [

84 {

85 "apr\_percentage": 15.24,

86 "apr\_type": "balance\_transfer\_apr",

87 "balance\_subject\_to\_apr": 1562.32,

88 "interest\_charge\_amount": 130.22

89 },

90 {

91 "apr\_percentage": 27.95,

92 "apr\_type": "cash\_apr",

93 "balance\_subject\_to\_apr": 56.22,

94 "interest\_charge\_amount": 14.81

95 },

96 {

97 "apr\_percentage": 12.5,

98 "apr\_type": "purchase\_apr",

99 "balance\_subject\_to\_apr": 157.01,

100 "interest\_charge\_amount": 25.66

101 },

102 {

103 "apr\_percentage": 0,

104 "apr\_type": "special",

105 "balance\_subject\_to\_apr": 1000,

106 "interest\_charge\_amount": 0

107 }

108 ],

109 "is\_overdue": false,

110 "last\_payment\_amount": 168.25,

111 "last\_payment\_date": "2019-05-22",

112 "last\_statement\_issue\_date": "2019-05-28",

113 "last\_statement\_balance": 1708.77,

114 "minimum\_payment\_amount": 20,

115 "next\_payment\_due\_date": "2020-05-28"

116 }

117 ],

118 "mortgage": [

119 {

120 "account\_id": "BxBXxLj1m4HMXBm9WZJyUg9XLd4rKEhw8Pb1J",

121 "account\_number": "3120194154",

122 "current\_late\_fee": 25.0,

123 "escrow\_balance": 3141.54,

124 "has\_pmi": true,

125 "has\_prepayment\_penalty": true,

126 "interest\_rate": {

127 "percentage": 3.99,

128 "type": "fixed",

129 },

130 "last\_payment\_amount": 3141.54,

131 "last\_payment\_date": "2019-08-01",

132 "loan\_term": "30 year",

133 "loan\_type\_description": "conventional",

134 "maturity\_date": "2045-07-31",

135 "next\_monthly\_payment": 3141.54,

136 "next\_payment\_due\_date": "2019-11-15",

137 "origination\_date": "2015-08-01",

138 "origination\_principal\_amount": 425000,

139 "past\_due\_amount": 2304,

140 "property\_address": {

141 "city": "Malakoff",

142 "country": "US",

143 "postal\_code": "14236",

144 "region": "NY",

145 "street": "2992 Cameron Road",

146 }

147 "ytd\_interest\_paid": 12300.4,

148 "ytd\_principal\_paid": 12340.5

149 }

150 ],

151 "student": [

152 {

153 "account\_id": "Pp1Vpkl9w8sajvK6oEEKtr7vZxBnGpf7LxxLE",

154 "account\_number": "4277075694",

155 "disbursement\_dates": ["2002-08-28"],

156 "expected\_payoff\_date": "2032-07-28",

157 "guarantor": "DEPT OF ED",

158 "interest\_rate\_percentage": 5.25,

159 "is\_overdue": false,

160 "last\_payment\_amount": 138.05,

161 "last\_payment\_date": "2019-04-22",

162 "last\_statement\_issue\_date": "2019-04-28",

163 "loan\_name": "Consolidation",

164 "loan\_status": {

165 "end\_date": "2032-07-28",

166 "type": "repayment"

167 },

168 "minimum\_payment\_amount": 25,

169 "next\_payment\_due\_date": "2019-05-28",

170 "origination\_date": "2002-08-28",

171 "origination\_principal\_amount": 25000,

172 "outstanding\_interest\_amount": 6227.36,

173 "payment\_reference\_number": "4277075694",

174 "pslf\_status": {

175 "estimated\_eligibility\_date": "2021-01-01",

176 "payments\_made": 200,

177 "payments\_remaining": 160

178 },

179 "repayment\_plan": {

180 "description": "Standard Repayment",

181 "type": "standard"

182 },

183 "sequence\_number": "1",

184 "servicer\_address": {

185 "city": "San Matias",

186 "country": "US",

187 "postal\_code": "99415",

188 "region": "CA",

189 "street": "123 Relaxation Road"

190 },

191 "ytd\_interest\_paid": 280.55,

192 "ytd\_principal\_paid": 271.65

193 }

194 ]

195 },

196 "request\_id": "dTnnm60WgKGLnKL"

197}

#### [Next steps](https://plaid.com/docs/liabilities/add-to-app/" \l "next-steps)

If you're ready to launch to Production, see the Launch checklist.

# Launch in Production checklist

#### Check off these recommended steps before launching your app in Production

Below is a list of recommended steps to take before launching your Plaid integration in production. While they might not all be required for the minimal operation of your application, the steps below will help to make your Plaid integration more robust, secure, efficient, and maintainable.

For a similar list of steps in an easy-to-follow, PDF guide format, see the [Plaid implementation handbook](https://plaid.com/documents/plaid-implementation-handbook.pdf).

#### [Production setup](https://plaid.com/docs/launch-checklist/" \l "production-setup)

You must migrate to Production to use more than 100 live (non-Sandbox) Items. Note that Items created on the Sandbox or Development environments cannot be migrated to Production.

If you haven't already done so, [request Production access](https://dashboard.plaid.com/overview/production).

Complete your [application profile](https://dashboard.plaid.com/settings/company/app-branding) and [company profile](https://dashboard.plaid.com/settings/company/profile), which are required to access certain institutions that use OAuth-based connections.

Complete your [security questionnaire](https://dashboard.plaid.com/overview/questionnaire-start), which is required to access certain US institutions that use OAuth-based connections.

For European Union countries and the UK, there is a separate compliance process required to access financial institutions in Production. If your Plaid integration will support financial institutions in Europe (including the UK), and your business is not based in Europe, [file a support ticket](https://dashboard.plaid.com/support/new/product-and-development/product-troubleshooting/request-product-access) to request Production access for Europe. Allow at least one week for your request to be processed.

Configure both Plaid Link and your server-side code to use the Production environment by setting the appropriate value when configuring your client object or (if not using a client library) your HTTP request. For language-specific details, see the GitHub page for your client library, or [API host information](https://plaid.com/docs/api/#api-host) if not using a library.

Ensure you are using your Production secret, which can be found on the [dashboard](http://dashboard.plaid.com/team/api).

If migrating from Sandbox, ensure that you remove any usage of Sandbox-specific functionality, such as the user\_good test user or calls to /sandbox/ endpoints.

Add teammates to the dashboard to give other users access. For more details, see [Plaid teams](https://plaid.com/docs/account/teams/).

#### [Link setup](https://plaid.com/docs/launch-checklist/" \l "link-setup)

Follow the steps below to assist users in completing the Link flow, help ensure compliance with Plaid's policies, and avoid being billed for unneeded products.

[Implement OAuth support](https://plaid.com/docs/link/oauth/). Some institutions in both Europe and the US require the use of OAuth-based connections. While OAuth will typically "just work" on desktop, as long as you have completed your [application profile](https://dashboard.plaid.com/settings/company/app-branding), [company profile](https://dashboard.plaid.com/settings/company/profile), and [security questionnaire](https://dashboard.plaid.com/overview/questionnaire-start), you may need to implement client-side redirect logic for users on mobile devices.

Make sure your OAuth integration works with all the [test cases](https://plaid.com/docs/link/oauth/#testing-oauth) recommended in the OAuth documentation.

To avoid unnecessary billing and user confusion, implement logic for preventing [duplicate Items](https://plaid.com/docs/link/duplicate-items/).

Configure Link customizations within the [dashboard](http://dashboard.plaid.com/link) and ensure that the countries and languages selected for the customizations match the settings provided to [**/link/token/create**](https://plaid.com/docs/api/tokens/#linktokencreate). For more details, see [Link customization](https://plaid.com/docs/link/customization/).

Ensure the client\_name parameter of [**/link/token/create**](https://plaid.com/docs/api/tokens/#linktokencreate) is set to display your company's name as you'd like it to appear within Link.

Ensure that the products parameter of [**/link/token/create**](https://plaid.com/docs/api/tokens/#linktokencreate) includes only the products you intend to use. The products listed here will influence which institutions and accounts appear in Link (only institutions and account types that support all specified products will appear in Link) and will trigger a billing event for listed products upon successful token exchange.

If you are using multiple Plaid products, see the recommendations in [Choosing how to initialize products](https://plaid.com/docs/link/initializing-products/) to make sure your Link sessions are optimized for performance, conversion, and billing.

Provide any notice and obtain any consent required for Plaid to process end user information in accordance with Plaid's [End User Privacy Policy](https://plaid.com/legal/#end-user-privacy-policy).

You may find it helpful to know that some of our customers link to Plaid's privacy policy within their own privacy policy, while others surface a separate just-in-time notice and consent page during their onboarding flow. Ultimately, it is up to you to determine how to obtain any legally required consents.

To maximize the number of users who complete the Link flow, review Plaid's recommendations for [optimizing Link conversion](https://plaid.com/docs/link/best-practices/) and implement any practices relevant to your use case, such as providing [pre-Link messaging](https://plaid.com/docs/link/messaging/) or enabling the [returning user experience](https://plaid.com/docs/link/returning-user/).

##### [Callbacks](https://plaid.com/docs/launch-checklist/" \l "callbacks)

Handle callbacks beyond just onSuccess in order to gracefully handle errors and build analytics around Link.

Listen to the [**onExit()**](https://plaid.com/docs/link/web/#onexit) and [**onEvent()**](https://plaid.com/docs/link/web/#onevent) callbacks for error\_type and error\_code in order to implement error handling.

Listen to the [**onEvent()**](https://plaid.com/docs/link/web/#onevent) callback for exit\_status or timestamp in order to implement Link conversion analytics. Alternatively, you can use Plaid's built-in [Link conversion analytics](https://dashboard.plaid.com/link-analytics) in the Dashboard.

#### [Webhook configuration](https://plaid.com/docs/launch-checklist/" \l "webhook-configuration)

Plaid uses webhooks for many common flows. If your integration does any of the following, webhooks are required or strongly recommended:

* (Any product) Your application is calling a Plaid endpoint for an Item repeatedly, over a period of time, not just immediately after Link.
* (Transactions or Investments) You are accessing transactions made after the end user linked their Item.
* (Auth) You are using automated micro-deposits for account verification.
* (Assets) You are creating Asset Reports.
* (Income) You are verifying a user's income.
* (Payment Initiation (UK and Europe)) You are initiating payments.
* (Virtual Accounts (UK and Europe) or Transfer) You are sending or receiving funds.

Make sure Link has been initialized with your URL for receiving webhooks via [**/link/token/create**](https://plaid.com/docs/api/tokens/#linktokencreate).

(For Identity Verification, Transfer or Payment Initiation, as well as [optional Auth micro-deposit events](https://plaid.com/docs/auth/coverage/microdeposit-events/)) Make sure to configure account level webhook URLs via the [Plaid Dashboard](https://dashboard.plaid.com/developers/webhooks).

Make sure you can receive webhooks from [Plaid's webhook IPs](https://plaid.com/docs/api/webhooks/).

Review Plaid's [webhook best practices](https://plaid.com/docs/api/webhooks/#best-practices-for-applications-using-webhooks) to ensure your webhook handling logic is robust to outages and traffic spikes.

#### [Error handling](https://plaid.com/docs/launch-checklist/" \l "error-handling)

Sometimes, Plaid API calls may fail due to intermittent outages or connectivity errors at supported institutions. Implement retry logic or error handling as necessary for product API calls.

#### [Link in update mode](https://plaid.com/docs/launch-checklist/" \l "link-in-update-mode)

Update mode is used to fix Items that have entered an error state (for example, because a user changed their password). If your application needs to access an Item repeatedly over a period of time, rather than just immediately after Link, implementing update mode logic is strongly recommended.

Handle the ITEM\_LOGIN\_REQUIRED and PENDING\_EXPIRATION Item error states by launching update mode to ensure your users retain access to their Items. For more information, see [Updating Items via Link](https://plaid.com/docs/link/update-mode/).

Listen for the [**NEW\_ACCOUNTS\_AVAILABLE**](https://plaid.com/docs/api/items/#new_accounts_available) webhook to learn when an Item has a new account associated with it. To request access to that account, launch Link in [update mode](https://plaid.com/docs/link/update-mode/).

(Optional, Auth and Identity only) Implement [Product Validations](https://plaid.com/docs/link/update-mode/#resolving-access_not_granted-or-no_auth_accounts-errors-via-product-validations) in update mode to prevent customers from accidentally revoking required permissions during the update mode flow.

#### [Storage & logging](https://plaid.com/docs/launch-checklist/" \l "storage--logging)

Log Plaid identifiers and IDs properly to enhance security, when contacting Support about a specific request or callback, and for finding specific entries in the [Activity Log](https://dashboard.plaid.com/activity/logs). For more information, see [Dashboard logs and troubleshooting](https://plaid.com/docs/account/activity/).

[Access tokens](https://plaid.com/docs/quickstart/glossary/#access-token) and [Item IDs](https://plaid.com/docs/quickstart/glossary/#item-id) are the core identifiers that map your users to their financial institutions. Store them securely and associate them with users of your application. Make sure, however, that these identifiers are never exposed client-side. Keep in mind that one user can create multiple Items if they have accounts with multiple financial institutions.

The same storage requirements apply to other types of tokens used instead of access tokens by certain products, such as asset report tokens (used with Assets), payment profile tokens (used with Transfer) and user tokens (used with Income).

Ensure that the following identifiers are securely logged, as they will be needed when contacting Support about a specific request or callback.

link\_session\_id: Included in the onExit, onEvent, and onSuccess callback of a Link integration.

request\_id: Included in all Plaid API responses.

account\_id: Included in all successful Plaid API responses that relate to a specific Item or account.

item\_id: Included in all successful Plaid API responses that relate to a specific Item.

#### [Item management](https://plaid.com/docs/launch-checklist/" \l "item-management)

Delete Items using [**/item/remove**](https://plaid.com/docs/api/items/#itemremove) when they are no longer being used. For example, you may wish to allow users to remove linked Items through your app's account management interface, or you may want to delete Items when a user deletes their account with your service or becomes inactive, or if the Item has been in an error state for an extended period. Deleting unneeded Items is a security best practice and will also prevent you from being charged for these Items when using a subscription-based product such as Transactions.

#### [Multi-app use cases](https://plaid.com/docs/launch-checklist/" \l "multi-app-use-cases)

If you anticipate having multiple apps that use Plaid or having multiple teams at your company using it, consider creating a single shared backend service to handle functions such as logging and token storage in a consolidated place.

### [Product-specific recommendations](https://plaid.com/docs/launch-checklist/" \l "product-specific-recommendations)

#### [Auth](https://plaid.com/docs/launch-checklist/" \l "auth)

If you are using Auth for an account funding use case, see the [Plaid account funding guide](https://plaid.com/documents/plaid-account-funding-guide.pdf) for use case specific recommendations.

If launching in the US, support the [Automated micro-deposit](https://plaid.com/docs/auth/coverage/automated/) and [Same-day micro-deposit](https://plaid.com/docs/auth/coverage/same-day/) flows for maximum institution coverage.

If supporting the Instant Match, Automated micro-deposit, or Same-Day micro-deposit flows, make sure that the country codes parameter provided to [**/link/token/create**](https://plaid.com/docs/api/tokens/#linktokencreate) (or directly to Link, if using a legacy public key implementation) for these flows includes only the US and no other countries.

If supporting the Automated micro-deposit flow, make sure to listen for [Auth webhooks](https://plaid.com/docs/api/products/auth/#webhooks) to know when the transaction is completed.

Make sure you are not displaying account numbers in your app's UI, even if truncated or masked, to avoid user confusion when working with institutions that provide virtualized or temporary account numbers. Always use values from the mask field instead.

#### [Balance](https://plaid.com/docs/launch-checklist/" \l "balance)

If you are using Balance for an account funding use case, see the [Plaid account funding guide](https://plaid.com/documents/plaid-account-funding-guide.pdf) for use case specific recommendations.

If you are using Balance with non-depository accounts, such as credit card or loan accounts, make sure to specify min\_last\_updated\_datetime when calling [**/accounts/balance/get**](https://plaid.com/docs/api/products/balance/#accountsbalanceget) to ensure balance calls to Capital One can succeed.

#### [Identity](https://plaid.com/docs/launch-checklist/" \l "identity)

If you are using Identity for an account funding use case, see the [Plaid account funding guide](https://plaid.com/documents/plaid-account-funding-guide.pdf) for use case specific recommendations.

#### [Transactions](https://plaid.com/docs/launch-checklist/" \l "transactions)

If fetching historical Transactions data using [**/transactions/get**](https://plaid.com/docs/api/products/transactions/#transactionsget) or [**/transactions/sync**](https://plaid.com/docs/api/products/transactions/#transactionssync), make sure your app implements pagination logic. For sample code, see the [integration instructions](https://plaid.com/docs/transactions/add-to-app/#fetching-transaction-data) or the API Reference for the endpoint.

If you use [**/transactions/sync**](https://plaid.com/docs/api/products/transactions/#transactionssync), make sure your app properly handles errors that occur during pagination by restarting the pagination loop. For more details, see [**TRANSACTIONS\_SYNC\_MUTATION\_DURING\_PAGINATION**](https://plaid.com/docs/errors/transactions/#transactions_sync_mutation_during_pagination).

Handle Transactions webhooks, see [Handling Transaction webhooks](https://plaid.com/docs/transactions/webhooks/).

#### [Investments](https://plaid.com/docs/launch-checklist/" \l "investments)

Handle the HOLDINGS-type [**DEFAULT\_UPDATE**](https://plaid.com/docs/api/products/investments/#holdings-default_update) webhook if your app needs to keep holdings, values, and balances up-to-date.

Handle the INVESTMENTS\_TRANSACTIONS-type [**DEFAULT\_UPDATE**](https://plaid.com/docs/api/products/investments/#investments_transactions-default_update) webhook if your app needs to keep Investments transactions up-to-date.

If fetching historical Investments transactions using [**/investments/transactions/get**](https://plaid.com/docs/api/products/investments/#investmentstransactionsget), make sure your app implements pagination logic. For sample code, see the [**/investments/transactions/get**](https://plaid.com/docs/api/products/investments/#investmentstransactionsget) API Reference (pagination examples available in Python and Ruby code samples).

#### [Assets](https://plaid.com/docs/launch-checklist/" \l "assets)

Handle the ASSETS-type [**PRODUCT\_READY**](https://plaid.com/docs/api/products/assets/#product_ready) webhook to know when to call [**/asset\_report/get**](https://plaid.com/docs/api/products/assets/#asset_reportget).

Handle the ASSETS-type [**ERROR**](https://plaid.com/docs/api/products/assets/#error) webhook to gracefully handle errors from failures in [**/asset\_report/create**](https://plaid.com/docs/api/products/assets/#asset_reportcreate).

#### [Payment Initiation](https://plaid.com/docs/launch-checklist/" \l "payment-initiation)

Handle the [**PAYMENT\_STATUS\_UPDATE**](https://plaid.com/docs/api/products/payment-initiation/#payment_status_update) webhook to keep updated on payment status information.

#### [Virtual Accounts](https://plaid.com/docs/launch-checklist/" \l "virtual-accounts)

Handle the [**WALLET\_TRANSACTION\_STATUS\_UPDATE**](https://plaid.com/docs/api/products/virtual-accounts/#wallet_transaction_status_update) webhook to keep updated on transaction status information.

#### [Income](https://plaid.com/docs/launch-checklist/" \l "income)

See the [Plaid income verification solution guide](https://plaid.com/documents/plaid-income-verification-solution-guide.pdf) for use case specific recommendations.

Handle the INCOME-type [**income\_verification**](https://plaid.com/docs/api/products/income/#income_verification) webhook to know when the verification is complete and endpoints that show income or employment related data are ready to be called.

#### [Identity Verification](https://plaid.com/docs/launch-checklist/" \l "identity-verification)

See the [Plaid Identity Verification and Monitor solution guide](https://plaid.com/documents/plaid-idv-monitor-solution-guide.pdf) for use case specific recommendations.

Make sure to update your template IDs when moving to Production; the Sandbox and Production environments have different template IDs.

If you've set any of your Acceptable Risk Levels to "high" in your Risk Rules in order to prevent checks from failing during testing, make sure to change them back to the levels you plan to use in Production.

If enabling Selfie Check, make sure that you are [requesting the required camera permissions](https://plaid.com/docs/identity-verification/#mobile-support) on mobile.

When performing your matching logic, ensure make sure to take into account that no\_data is different from no\_match. no\_data means that the list issuer didn’t supply data against which to match. no\_match means that data was provided by the list issuer and it did not match the information provided by your customer.

#### [Monitor](https://plaid.com/docs/launch-checklist/" \l "monitor)

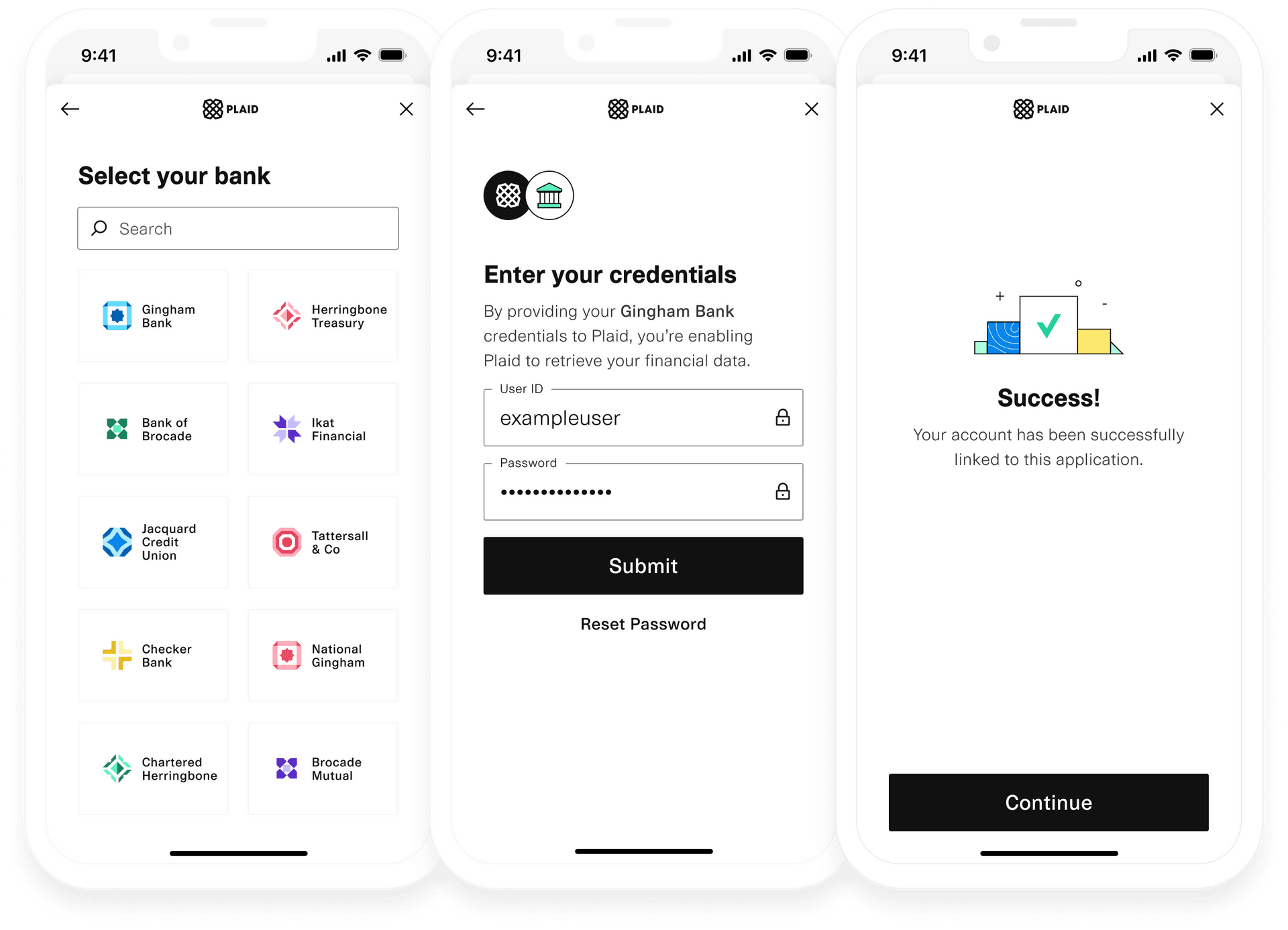
See the [Plaid Identity Verification and Monitor solution guide](https://plaid.com/documents/plaid-idv-monitor-solution-guide.pdf) for use case specific recommendations.

Make sure to implement logic for both onboarding (adding users to a program when they are created) and offboarding (moving users to a program with rescans disabled when they close their account).

Ensure that the Monitor review queue is checked regularly. For more details, see [Building a reviewer workflow](https://plaid.com/docs/monitor/#preparing-for-ongoing-screening) and [preparing for ongoing screening](https://plaid.com/docs/monitor/#preparing-for-ongoing-screening).

# Link overview

#### Use Link to connect to your users' financial accounts with the Plaid API



#### [Introduction to Link](https://plaid.com/docs/link/" \l "introduction-to-link)

Plaid Link is the client-side component that your users will interact with in order to link their accounts to Plaid and allow you to access their accounts via the Plaid API.

Plaid Link will handle credential validation, multi-factor authentication, and error handling for each institution that Plaid supports. Link is supported via SDKs for all modern browsers and Platforms, including [web](https://plaid.com/docs/link/web/), [iOS](https://plaid.com/docs/link/ios/), [Android](https://plaid.com/docs/link/android/), as well as via [React Native](https://plaid.com/docs/link/react-native/) and [mobile webviews](https://plaid.com/docs/link/webview/), along with a community-supported wrapper for [Flutter](https://github.com/jorgefspereira/plaid_flutter). To try Link, see [Plaid Link Demo](https://plaid.com/demo/).

Link is the only available method for connecting accounts in Development and Production and is mandatory for all Plaid integrations. In the Sandbox test environment, Link can optionally be bypassed for testing purposes via [**/sandbox/public\_token/create**](https://plaid.com/docs/api/sandbox/#sandboxpublic_tokencreate).

Want to build a Plaid-powered application using Link directly in your browser? The [Plaid Basics interactive tutorial](https://plaid.com/tutorials/basics/) can walk you through the process.

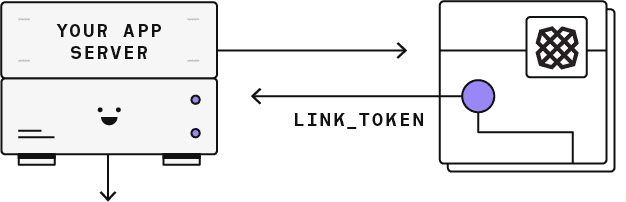
#### [Link flow overview](https://plaid.com/docs/link/" \l "link-flow-overview)

The diagram below shows a model of how Link is used to obtain a public\_token, which can then be exchanged for an access\_token, which is used to authenticate requests to the Plaid API.

**The Plaid flow** begins when your user wants to connect their bank account to your app.



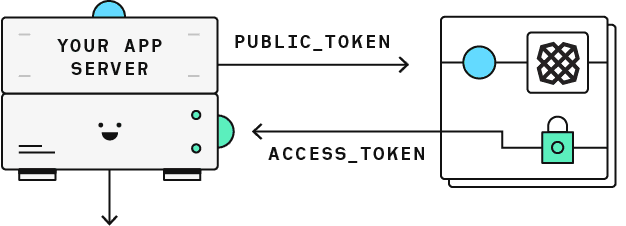
**1**Call [**/link/token/create**](https://plaid.com/docs/api/tokens/#linktokencreate) to create a link\_token and pass the temporary token to your app's client.



**2**Use the link\_token to open Link for your user. In the [**onSuccess** callback](https://plaid.com/docs/link/web/#onsuccess), Link will provide a temporary public\_token.



**3**Call [**/item/public\_token/exchange**](https://plaid.com/docs/api/tokens/#itempublic_tokenexchange) to exchange the public\_token for a permanent access\_token and item\_id for the new Item.



**4**Store the access\_token and use it to make product requests for your user's Item.



In code, this flow is initiated by creating a link\_token and using it to initialize Link. The link\_token can be configured with the Plaid products you will be using and the countries you will need to support. Once the user has logged in via Link, Link will issue a callback containing a public\_token, which can be exchanged for an access\_token via [**/item/public\_token/exchange**](https://plaid.com/docs/api/tokens/#itempublic_tokenexchange).

#### [Initializing Link](https://plaid.com/docs/link/" \l "initializing-link)

Link is initialized by passing the link\_token to Link. The exact implementation details for passing the link\_token will vary by platform. For detailed instructions, see the page for your specific platform: [web](https://plaid.com/docs/link/web/), [iOS](https://plaid.com/docs/link/ios/), [Android](https://plaid.com/docs/link/android/), [React Native](https://plaid.com/docs/link/react-native/) or [mobile webview](https://plaid.com/docs/link/webview/).

For recommendations on configuring the link\_token for your use case, see [Choosing how to initialize products](https://plaid.com/docs/link/initializing-products/).

#### [Supporting OAuth](https://plaid.com/docs/link/" \l "supporting-oauth)

Some institutions use an OAuth authentication flow, in which Plaid Link redirects the end user to their bank's website or mobile app to authenticate. To learn how to connect to an institution that uses OAuth, see the [OAuth guide](https://plaid.com/docs/link/oauth/).

#### [Customizing Link](https://plaid.com/docs/link/" \l "customizing-link)

You can customize parts of Link's flow, including some text elements, the institution select view, and the background color, and enable additional features like the Account Select view straight from the [Dashboard](https://dashboard.plaid.com/link). You can preview your changes in realtime and then publish them instantly once you're ready to go live. For more details, see [Link customization](https://plaid.com/docs/link/customization/).

To help you take advantage of the options available for customizing and configuring Link, Plaid offers a [best practices guide](https://plaid.com/docs/link/best-practices/) with recommendations for how to initialize and configure Link within your app.

Link's appearance will also automatically change if the institution selected is not in a healthy state. For more details, see [Institution status in Link](https://plaid.com/docs/link/institution-status/).

#### [Returning user flows](https://plaid.com/docs/link/" \l "returning-user-flows)

The returning user flow allows you to enable a faster Plaid Link experience for your users who already use Plaid. To learn more, see [Returning user experience](https://plaid.com/docs/link/returning-user/).

#### [Error-handling flows](https://plaid.com/docs/link/" \l "error-handling-flows)

If your application will access an Item on a recurring basis, rather than just once, it should support [update mode](https://plaid.com/docs/link/update-mode/). Update mode allows you to refresh an Item if it enters an error state, such as when a user changes their password or MFA information. For more information, see [Updating an Item](https://plaid.com/docs/link/update-mode/).

It's also recommended to have special handling for when a user attempts to link the same Item twice. Requesting access tokens for duplicate Items can lead to higher bills and end-user confusion. To learn more, see [preventing duplicate Items](https://plaid.com/docs/link/duplicate-items/).

Occasionally, Link itself can enter an error state if the user takes over 30 minutes to complete the Link process. For information on handling this flow, see [Handling invalid Link tokens](https://plaid.com/docs/link/handle-invalid-link-token/).

#### [Optimizing Link conversion](https://plaid.com/docs/link/" \l "optimizing-link-conversion)

How you configure Link can have a huge impact on the percentage of users who successfully complete the Link flow. To ensure you're maximizing conversion, see [Best practices for Link conversion](https://plaid.com/docs/link/best-practices/).

#### [Link token migration](https://plaid.com/docs/link/" \l "link-token-migration)

Prior to July 2020, Link was initialized using a public key instead of a link\_token. If you are an existing Plaid developer using a Plaid public key, it is recommended that you migrate to a link\_token-based implementation as soon as possible, using the instructions in the [Link token migration guide](https://plaid.com/docs/link/link-token-migration-guide/). For those who are not yet ready to migrate, instructions on maintaining a legacy integration can be found in the [legacy integration guide](https://plaid.com/docs/link/maintain-legacy-integration/).

#### [Troubleshooting](https://plaid.com/docs/link/" \l "troubleshooting)

Since all your users will go through Link, it's important to build as robust an integration as possible. For details on dealing with common problems, see the [Troubleshooting](https://plaid.com/docs/link/troubleshooting/) section.

#### [Link updates](https://plaid.com/docs/link/" \l "link-updates)

Plaid periodically updates Link to add new functionality and improve conversion. These changes will be automatically deployed. Any test suites and business logic in your app should be robust to the possibility of changes to the user-facing Link flow.

Users of Plaid's SDKs for React, React Native, iOS, and Android should regularly update to ensure support for the latest client platforms and Plaid functionality.

# Link Web SDK

#### Reference for integrating with the Link JavaScript SDK and React SDK

Prefer to learn with code examples? Check out our [GitHub repo](https://github.com/plaid/tiny-quickstart) with working example Link implementations for both [JavaScript](https://github.com/plaid/tiny-quickstart/tree/main/vanilla_js) and [React](https://github.com/plaid/tiny-quickstart/tree/main/react).

#### [Installation](https://plaid.com/docs/link/web/" \l "installation)

Select group for content switcher

JavaScriptReact

Include the Plaid Link initialize script on each page of your site. It must always be loaded directly from https://cdn.plaid.com, rather than included in a bundle or hosted yourself. Unlike Plaid's other SDKs, the JavaScript web SDK is not versioned; cdn.plaid.com will automatically provide the latest available SDK.

index.html

**Copy**

1<script src="https://cdn.plaid.com/link/v2/stable/link-initialize.js"></script>

To get started with Plaid Link for React, clone the [GitHub repository](https://github.com/plaid/react-plaid-link) and review the example application and README, which provide reference implementations.

Next, you'll need to install the react-plaid-link package.

With npm:

**Copy**

1npm install --save react-plaid-link

With yarn:

**Copy**

1yarn add react-plaid-link

Then import the necessary components and types:

**Copy**

1import {

2 usePlaidLink,

3 PlaidLinkOptions,

4 PlaidLinkOnSuccess,

5} from 'react-plaid-link';

#### [CSP directives](https://plaid.com/docs/link/web/" \l "csp-directives)

If you are using a Content Security Policy (CSP), use the following directives to allow Link traffic:

**Curl**

**Select Language**

**Copy**

1default-src https://cdn.plaid.com/

2script-src https://cdn.plaid.com/link/v2/stable/link-initialize.js

3frame-src https://cdn.plaid.com/

4connect-src https://production.plaid.com/

If using Sandbox or Development instead of production, make sure to update the connect-src directive to point to the appropriate server (https://sandbox.plaid.com or https://development.plaid.com).

#### [Create](https://plaid.com/docs/link/web/" \l "create)

Plaid.create accepts one argument, a configuration Object, and returns an Object with three functions, [**open**](https://plaid.com/docs/link/web/#open), [**exit**](https://plaid.com/docs/link/web/#exit), and [**destroy**](https://plaid.com/docs/link/web/#destroy). Calling open will display the Consent Pane view, calling exit will close Link, and calling destroy will clean up the iframe.  
When using the React SDK, this method is called usePlaidLink and returns an object with four values, [**open**](https://plaid.com/docs/link/web/#open), [**exit**](https://plaid.com/docs/link/web/#exit), ready, and error. The values open and exit behave as described above. ready is a passthrough for onLoad and will be true when Link is ready to be opened. error is populated only if Plaid fails to load the Link JavaScript. There is no separate method to destroy Link in the React SDK, as unmount will automatically destroy the Link instance.  
**Note:** Control whether or not your Link integration uses the Account Select view from the [Dashboard](https://dashboard.plaid.com/signin?redirect=%2Flink%2Faccount-select).

[**token**](https://plaid.com/docs/link/web/#link-web-create-token)

string

Specify a link\_token to authenticate your app with Link. This is a short lived, one-time use token that should be unique for each Link session. In addition to the primary flow, a link\_token can be configured to launch Link in [update mode](https://plaid.com/docs/link/update-mode/). See the [/link/token/create](https://plaid.com/docs/api/tokens#create-a-link_token) endpoint for a full list of configurations.

[**onSuccess**](https://plaid.com/docs/link/web/#link-web-create-onSuccess)

callback

A function that is called when a user successfully links an Item. The function should expect two arguments, the public\_token and a metadata object. See [onSuccess](https://plaid.com/docs/link/web/#onsuccess).

[**onExit**](https://plaid.com/docs/link/web/#link-web-create-onExit)

callback

A function that is called when a user exits Link without successfully linking an Item, or when an error occurs during Link initialization. The function should expect two arguments, a nullable error object and a metadata object. See [onExit](https://plaid.com/docs/link/web/#onexit).

[**onEvent**](https://plaid.com/docs/link/web/#link-web-create-onEvent)

callback

A function that is called when a user reaches certain points in the Link flow. The function should expect two arguments, an eventName string and a metadata object. See [onEvent](https://plaid.com/docs/link/web/#onevent).

[**onLoad**](https://plaid.com/docs/link/web/#link-web-create-onLoad)

callback

A function that is called when the Link module has finished loading. Calls to plaidLinkHandler.open() prior to the onLoad callback will be delayed until the module is fully loaded.

[**receivedRedirectUri**](https://plaid.com/docs/link/web/#link-web-create-receivedRedirectUri)

string

A receivedRedirectUri is required to support OAuth authentication flows when re-launching Link on a mobile device.

[**key**](https://plaid.com/docs/link/web/#link-web-create-key)

deprecatedstring

The public\_key is no longer used for new implementations of Link. If your integration is still using a public\_key, see the [migration guide](https://plaid.com/docs/link/link-token-migration-guide) to upgrade to using a link\_token. See the [maintenance guide](https://plaid.com/docs/link/maintain-legacy-integration) to troubleshoot any public\_key issues.

Create example

**JavaScript**

**Select Language**

**Copy**

1const handler = Plaid.create({

2 token: 'GENERATED\_LINK\_TOKEN',

3 onSuccess: (public\_token, metadata) => {},

4 onLoad: () => {},

5 onExit: (err, metadata) => {},

6 onEvent: (eventName, metadata) => {},

7});

#### [onSuccess](https://plaid.com/docs/link/web/" \l "onsuccess)

The onSuccess callback is called when a user successfully links an Item. It takes two arguments: the public\_token and a metadata object.

Collapse all

[**public\_token**](https://plaid.com/docs/link/web/#link-web-onsuccess-public-token)

string

Displayed once a user has successfully linked their Item.

[**metadata**](https://plaid.com/docs/link/web/#link-web-onsuccess-metadata)

object

Displayed once a user has successfully linked their Item.

Hide object

[**institution**](https://plaid.com/docs/link/web/#link-web-onsuccess-metadata-institution)

nullableobject

An institution object. If the Item was created via Same-Day micro-deposit verification, will be null.

Hide object

[**name**](https://plaid.com/docs/link/web/#link-web-onsuccess-metadata-institution-name)

string

The full institution name, such as 'Wells Fargo'

[**institution\_id**](https://plaid.com/docs/link/web/#link-web-onsuccess-metadata-institution-institution-id)

string

The Plaid institution identifier

[**accounts**](https://plaid.com/docs/link/web/#link-web-onsuccess-metadata-accounts)

object

A list of accounts attached to the connected Item. If Account Select is enabled via the developer dashboard, accounts will only include selected accounts.

Hide object

[**id**](https://plaid.com/docs/link/web/#link-web-onsuccess-metadata-accounts-id)

string

The Plaid account\_id

[**name**](https://plaid.com/docs/link/web/#link-web-onsuccess-metadata-accounts-name)

string

The official account name

[**mask**](https://plaid.com/docs/link/web/#link-web-onsuccess-metadata-accounts-mask)

nullablestring

The last 2-4 alphanumeric characters of an account's official account number. Note that the mask may be non-unique between an Item's accounts. It may also not match the mask that the bank displays to the user.

[**type**](https://plaid.com/docs/link/web/#link-web-onsuccess-metadata-accounts-type)

string

The account type. See the [Account schema](https://plaid.com/docs/api/accounts#account-type-schema) for a full list of possible values

[**subtype**](https://plaid.com/docs/link/web/#link-web-onsuccess-metadata-accounts-subtype)

string

The account subtype. See the [Account schema](https://plaid.com/docs/api/accounts#account-type-schema) for a full list of possible values

[**verification\_status**](https://plaid.com/docs/link/web/#link-web-onsuccess-metadata-accounts-verification-status)

nullablestring

Indicates an Item's micro-deposit-based verification status. Possible values are:  
pending\_automatic\_verification: The Item is pending automatic verification  
pending\_manual\_verification: The Item is pending manual micro-deposit verification. Items remain in this state until the user successfully verifies the deposit.  
automatically\_verified: The Item has successfully been automatically verified  
manually\_verified: The Item has successfully been manually verified  
verification\_expired: Plaid was unable to automatically verify the deposit within 7 calendar days and will no longer attempt to validate the Item. Users may retry by submitting their information again through Link.  
verification\_failed: The Item failed manual micro-deposit verification because the user exhausted all 3 verification attempts. Users may retry by submitting their information again through Link.  
null: micro-deposit-based verification is not being used for the Item.

[**class\_type**](https://plaid.com/docs/link/web/#link-web-onsuccess-metadata-accounts-class-type)

nullablestring

If micro-deposit verification is being used, indicates whether the account being verified is a business or personal account.

[**account**](https://plaid.com/docs/link/web/#link-web-onsuccess-metadata-account)

deprecatednullableobject

Deprecated. Use accounts instead.

[**link\_session\_id**](https://plaid.com/docs/link/web/#link-web-onsuccess-metadata-link-session-id)

string

A unique identifier associated with a user's actions and events through the Link flow. Include this identifier when opening a support ticket for faster turnaround.

[**transfer\_status**](https://plaid.com/docs/link/web/#link-web-onsuccess-metadata-transfer-status)

nullablestring

The status of a transfer. Returned only when [Transfer UI](https://plaid.com/docs/transfer/using-transfer-ui) is implemented.

* COMPLETE – The transfer was completed.
* INCOMPLETE – The transfer could not be completed. For help, see [Troubleshooting transfers](https://plaid.com/docs/transfer/using-transfer-ui#troubleshooting-transfers).

Possible values: COMPLETE, INCOMPLETE

onSuccess example

**JavaScript**

**Select Language**

**Copy**

1const handler = Plaid.create({

2 ...,

3 onSuccess: (public\_token, metadata) => {

4 fetch('//yourserver.com/get\_access\_token', {

5 method: 'POST',

6 headers: {

7 'Content-Type': 'application/json',

8 },

9 body: json.Stringify({

10 public\_token: public\_token,

11 }),

12 });

13 }

14});

Metadata schema

**Copy**

1{

2 institution: {

3 name: 'Wells Fargo',

4 institution\_id: 'ins\_4'

5 },

6 accounts: [

7 {

8 id: 'ygPnJweommTWNr9doD6ZfGR6GGVQy7fyREmWy',

9 name: 'Plaid Checking',

10 mask: '0000',

11 type: 'depository',

12 subtype: 'checking',

13 verification\_status: ''

14 },

15 {

16 id: '9ebEyJAl33FRrZNLBG8ECxD9xxpwWnuRNZ1V4',

17 name: 'Plaid Saving',

18 mask: '1111',

19 type: 'depository',

20 subtype: 'savings'

21 }

22 ...

23 ],

24 link\_session\_id: '79e772be-547d-4c9c-8b76-4ac4ed4c441a'

25}

#### [onExit](https://plaid.com/docs/link/web/" \l "onexit)

The onExit callback is called when a user exits Link without successfully linking an Item, or when an error occurs during Link initialization. onExit takes two arguments, a nullable error object and a metadata object. The metadata parameter is always present, though some values may be null. Note that onExit will not be called when Link is destroyed in some other way than closing Link, such as the user hitting the browser back button or closing the browser tab on which the Link session is present.

Collapse all

[**error**](https://plaid.com/docs/link/web/#link-web-onexit-error)

nullableobject

A nullable object that contains the error type, code, and message of the error that was last encountered by the user. If no error was encountered, error will be null.

Hide object

[**error\_type**](https://plaid.com/docs/link/web/#link-web-onexit-error-error-type)

String

A broad categorization of the error.

[**error\_code**](https://plaid.com/docs/link/web/#link-web-onexit-error-error-code)

String

The particular error code. Each error\_type has a specific set of error\_codes.

[**error\_message**](https://plaid.com/docs/link/web/#link-web-onexit-error-error-message)

String

A developer-friendly representation of the error code.

[**display\_message**](https://plaid.com/docs/link/web/#link-web-onexit-error-display-message)

nullableString

A user-friendly representation of the error code. null if the error is not related to user action. This may change over time and is not safe for programmatic use.

[**metadata**](https://plaid.com/docs/link/web/#link-web-onexit-metadata)

object

Displayed if a user exits Link without successfully linking an Item.

Hide object

[**institution**](https://plaid.com/docs/link/web/#link-web-onexit-metadata-institution)

nullableobject

An institution object. If the Item was created via Same-Day micro-deposit verification, will be null.

Hide object

[**name**](https://plaid.com/docs/link/web/#link-web-onexit-metadata-institution-name)

string

The full institution name, such as Wells Fargo

[**institution\_id**](https://plaid.com/docs/link/web/#link-web-onexit-metadata-institution-institution-id)

string

The Plaid institution identifier

[**status**](https://plaid.com/docs/link/web/#link-web-onexit-metadata-status)

string

The point at which the user exited the Link flow. One of the following values.

Hide object

[**requires\_questions**](https://plaid.com/docs/link/web/#link-web-onexit-metadata-status-requires-questions)

User prompted to answer security questions

[**requires\_selections**](https://plaid.com/docs/link/web/#link-web-onexit-metadata-status-requires-selections)

User prompted to answer multiple choice question(s)

[**requires\_code**](https://plaid.com/docs/link/web/#link-web-onexit-metadata-status-requires-code)

User prompted to provide a one-time passcode

[**choose\_device**](https://plaid.com/docs/link/web/#link-web-onexit-metadata-status-choose-device)

User prompted to select a device on which to receive a one-time passcode

[**requires\_credentials**](https://plaid.com/docs/link/web/#link-web-onexit-metadata-status-requires-credentials)

User prompted to provide credentials for the selected financial institution or has not yet selected a financial institution

[**requires\_account\_selection**](https://plaid.com/docs/link/web/#link-web-onexit-metadata-status-requires-account-selection)

User prompted to select one or more financial accounts to share

[**requires\_oauth**](https://plaid.com/docs/link/web/#link-web-onexit-metadata-status-requires-oauth)

User prompted to enter an OAuth flow

[**institution\_not\_found**](https://plaid.com/docs/link/web/#link-web-onexit-metadata-status-institution-not-found)

User exited the Link flow after unsuccessfully (no results returned) searching for a financial institution

[**institution\_not\_supported**](https://plaid.com/docs/link/web/#link-web-onexit-metadata-status-institution-not-supported)

User exited the Link flow after discovering their selected institution is no longer supported by Plaid

[**link\_session\_id**](https://plaid.com/docs/link/web/#link-web-onexit-metadata-link-session-id)

string

A unique identifier associated with a user's actions and events through the Link flow. Include this identifier when opening a support ticket for faster turnaround.

[**request\_id**](https://plaid.com/docs/link/web/#link-web-onexit-metadata-request-id)

string

The request ID for the last request made by Link. This can be shared with Plaid Support to expedite investigation.

onExit example

**JavaScript**

**Select Language**

**Copy**

1const handler = Plaid.create({

2 ...,

3 onExit: (error, metadata) => {

4 // Save data from the onExit handler

5 supportHandler.report({

6 error: error,

7 institution: metadata.institution,

8 link\_session\_id: metadata.link\_session\_id,

9 plaid\_request\_id: metadata.request\_id,

10 status: metadata.status,

11 });

12 },

13});

Error schema

**Copy**

1{

2 error\_type: 'ITEM\_ERROR',

3 error\_code: 'INVALID\_CREDENTIALS',

4 error\_message: 'the credentials were not correct',

5 display\_message: 'The credentials were not correct.',

6}

Metadata schema

**Copy**

1{

2 institution: {

3 name: 'Wells Fargo',

4 institution\_id: 'ins\_4'

5 },

6 status: 'requires\_credentials',

7 link\_session\_id: '36e201e0-2280-46f0-a6ee-6d417b450438',

8 request\_id: '8C7jNbDScC24THu'

9}

#### [onEvent](https://plaid.com/docs/link/web/" \l "onevent)

The onEvent callback is called at certain points in the Link flow. It takes two arguments, an eventName string and a metadata object.  
The metadata parameter is always present, though some values may be null. Note that new eventNames, metadata keys, or view names may be added without notice.  
The onEvent callback is not guaranteed to fire exactly at the time of a user action in Link. In general, the OPEN event will fire in real time; subsequent events will fire at the end of the Link flow, along with the onSuccess or onExit callback. If you need to determine the exact time when an event happened, use the timestamp in the metadata.  
The following callback events are stable, which means that they will not be deprecated or changed: OPEN, EXIT, HANDOFF, SELECT\_INSTITUTION, ERROR, BANK\_INCOME\_INSIGHTS\_COMPLETED, IDENTITY\_VERIFICATION\_PASS\_SESSION, IDENTITY\_VERIFICATION\_FAIL\_SESSION. The remaining callback events are informational and subject to change.

Collapse all

[**eventName**](https://plaid.com/docs/link/web/#link-web-onevent-eventName)

string

A string representing the event that has just occurred in the Link flow.

Hide object

[**BANK\_INCOME\_INSIGHTS\_COMPLETED**](https://plaid.com/docs/link/web/#link-web-onevent-eventName-BANK-INCOME-INSIGHTS-COMPLETED)

The user has completed the Assets and Bank Income Insights flow.

[**CLOSE\_OAUTH**](https://plaid.com/docs/link/web/#link-web-onevent-eventName-CLOSE-OAUTH)

The user closed the third-party website or mobile app without completing the OAuth flow.

[**CONNECT\_NEW\_INSTITUTION**](https://plaid.com/docs/link/web/#link-web-onevent-eventName-CONNECT-NEW-INSTITUTION)

The user has chosen to link a new institution instead of linking a saved institution. This event is only emitted in the Link Remember Me flow.

[**ERROR**](https://plaid.com/docs/link/web/#link-web-onevent-eventName-ERROR)

A recoverable error occurred in the Link flow, see the error\_code metadata.

[**EXIT**](https://plaid.com/docs/link/web/#link-web-onevent-eventName-EXIT)

The user has exited without completing the Link flow and the [onExit](https://plaid.com/docs/link/web/#onexit) callback is fired.

[**FAIL\_OAUTH**](https://plaid.com/docs/link/web/#link-web-onevent-eventName-FAIL-OAUTH)

The user encountered an error while completing the third-party's OAuth login flow.

[**HANDOFF**](https://plaid.com/docs/link/web/#link-web-onevent-eventName-HANDOFF)

The user has exited Link after successfully linking an Item.

[**IDENTITY\_VERIFICATION\_START\_STEP**](https://plaid.com/docs/link/web/#link-web-onevent-eventName-IDENTITY-VERIFICATION-START-STEP)

The user has started a step of the Identity Verification flow. The step is indicated by view\_name.

[**IDENTITY\_VERIFICATION\_PASS\_STEP**](https://plaid.com/docs/link/web/#link-web-onevent-eventName-IDENTITY-VERIFICATION-PASS-STEP)

The user has passed a step of the Identity Verification flow. The step is indicated by view\_name.

[**IDENTITY\_VERIFICATION\_FAIL\_STEP**](https://plaid.com/docs/link/web/#link-web-onevent-eventName-IDENTITY-VERIFICATION-FAIL-STEP)

The user has failed a step of the Identity Verification flow. The step is indicated by view\_name.

[**IDENTITY\_VERIFICATION\_PENDING\_REVIEW\_STEP**](https://plaid.com/docs/link/web/#link-web-onevent-eventName-IDENTITY-VERIFICATION-PENDING-REVIEW-STEP)

The user has reached the pending review state.

[**IDENTITY\_VERIFICATION\_CREATE\_SESSION**](https://plaid.com/docs/link/web/#link-web-onevent-eventName-IDENTITY-VERIFICATION-CREATE-SESSION)

The user has started a new Identity Verification session.

[**IDENTITY\_VERIFICATION\_RESUME\_SESSION**](https://plaid.com/docs/link/web/#link-web-onevent-eventName-IDENTITY-VERIFICATION-RESUME-SESSION)

The user has resumed an existing Identity Verification session.

[**IDENTITY\_VERIFICATION\_PASS\_SESSION**](https://plaid.com/docs/link/web/#link-web-onevent-eventName-IDENTITY-VERIFICATION-PASS-SESSION)

The user has successfully completed their Identity Verification session.

[**IDENTITY\_VERIFICATION\_FAIL\_SESSION**](https://plaid.com/docs/link/web/#link-web-onevent-eventName-IDENTITY-VERIFICATION-FAIL-SESSION)

The user has failed their Identity Verification session.

[**IDENTITY\_VERIFICATION\_PENDING\_REVIEW\_SESSION**](https://plaid.com/docs/link/web/#link-web-onevent-eventName-IDENTITY-VERIFICATION-PENDING-REVIEW-SESSION)

The user has completed their Identity Verification session, which is now in a pending review state.

[**IDENTITY\_VERIFICATION\_OPEN\_UI**](https://plaid.com/docs/link/web/#link-web-onevent-eventName-IDENTITY-VERIFICATION-OPEN-UI)

The user has opened the UI of their Identity Verification session.

[**IDENTITY\_VERIFICATION\_RESUME\_UI**](https://plaid.com/docs/link/web/#link-web-onevent-eventName-IDENTITY-VERIFICATION-RESUME-UI)

The user has resumed the UI of their Identity Verification session.

[**IDENTITY\_VERIFICATION\_CLOSE\_UI**](https://plaid.com/docs/link/web/#link-web-onevent-eventName-IDENTITY-VERIFICATION-CLOSE-UI)

The user has closed the UI of their Identity Verification session.

[**MATCHED\_SELECT\_INSTITUTION**](https://plaid.com/docs/link/web/#link-web-onevent-eventName-MATCHED-SELECT-INSTITUTION)

The user selected an institution that was presented as a matched institution. This event can be emitted either during the Returning User Experience flow or if the institution's routing\_number was provided when calling /link/token/create. To distinguish between the two scenarios, see metadata.match\_reason.

[**MATCHED\_SELECT\_VERIFY\_METHOD**](https://plaid.com/docs/link/web/#link-web-onevent-eventName-MATCHED-SELECT-VERIFY-METHOD)

The user selected a verification method for a matched institution. This event is emitted during the Returning User Experience flow.

[**OPEN**](https://plaid.com/docs/link/web/#link-web-onevent-eventName-OPEN)

The user has opened Link.

[**OPEN\_MY\_PLAID**](https://plaid.com/docs/link/web/#link-web-onevent-eventName-OPEN-MY-PLAID)

The user has opened my.plaid.com. This event is only sent when Link is initialized with Assets as a product

[**OPEN\_OAUTH**](https://plaid.com/docs/link/web/#link-web-onevent-eventName-OPEN-OAUTH)

The user has navigated to a third-party website or mobile app in order to complete the OAuth login flow.

[**SEARCH\_INSTITUTION**](https://plaid.com/docs/link/web/#link-web-onevent-eventName-SEARCH-INSTITUTION)

The user has searched for an institution.

[**SELECT\_AUTH\_TYPE**](https://plaid.com/docs/link/web/#link-web-onevent-eventName-SELECT-AUTH-TYPE)

The user has chosen whether to Link instantly or manually (i.e., with micro-deposits). This event emits the selection metadata to indicate the user's selection.

[**SELECT\_BRAND**](https://plaid.com/docs/link/web/#link-web-onevent-eventName-SELECT-BRAND)

The user selected a brand, e.g. Bank of America. The SELECT\_BRAND event is only emitted for large financial institutions with multiple online banking portals.

[**SELECT\_DEGRADED\_INSTITUTION**](https://plaid.com/docs/link/web/#link-web-onevent-eventName-SELECT-DEGRADED-INSTITUTION)

The user selected an institution with a DEGRADED health status and was shown a corresponding message.

[**SELECT\_DOWN\_INSTITUTION**](https://plaid.com/docs/link/web/#link-web-onevent-eventName-SELECT-DOWN-INSTITUTION)

The user selected an institution with a DOWN health status and was shown a corresponding message.

[**SELECT\_FILTERED\_INSTITUTION**](https://plaid.com/docs/link/web/#link-web-onevent-eventName-SELECT-FILTERED-INSTITUTION)

The user selected an institution Plaid does not support all requested products for and was shown a corresponding message.

[**SELECT\_INSTITUTION**](https://plaid.com/docs/link/web/#link-web-onevent-eventName-SELECT-INSTITUTION)

The user selected an institution.

[**SKIP\_SUBMIT\_PHONE**](https://plaid.com/docs/link/web/#link-web-onevent-eventName-SKIP-SUBMIT-PHONE)

The user has opted to not provide their phone number to Plaid. This event is only emitted in the Link Remember Me flow.

[**SUBMIT\_ACCOUNT\_NUMBER**](https://plaid.com/docs/link/web/#link-web-onevent-eventName-SUBMIT-ACCOUNT-NUMBER)

The user has submitted an account number. This event emits the account\_number\_mask metadata to indicate the mask of the account number the user provided.

[**SUBMIT\_CREDENTIALS**](https://plaid.com/docs/link/web/#link-web-onevent-eventName-SUBMIT-CREDENTIALS)

The user has submitted credentials.

[**SUBMIT\_DOCUMENTS**](https://plaid.com/docs/link/web/#link-web-onevent-eventName-SUBMIT-DOCUMENTS)

The user is being prompted to submit documents for an Income verification flow.

[**SUBMIT\_DOCUMENTS\_ERROR**](https://plaid.com/docs/link/web/#link-web-onevent-eventName-SUBMIT-DOCUMENTS-ERROR)

The user encountered an error when submitting documents for an Income verification flow.

[**SUBMIT\_DOCUMENTS\_SUCCESS**](https://plaid.com/docs/link/web/#link-web-onevent-eventName-SUBMIT-DOCUMENTS-SUCCESS)

The user has successfully submitted documents for an Income verification flow.

[**SUBMIT\_MFA**](https://plaid.com/docs/link/web/#link-web-onevent-eventName-SUBMIT-MFA)

The user has submitted MFA.

[**SUBMIT\_PHONE**](https://plaid.com/docs/link/web/#link-web-onevent-eventName-SUBMIT-PHONE)

The user has submitted their phone number. This event is only emitted in the Link Remember Me flow.

[**SUBMIT\_ROUTING\_NUMBER**](https://plaid.com/docs/link/web/#link-web-onevent-eventName-SUBMIT-ROUTING-NUMBER)

The user has submitted routing number. This event emits the routing\_number metadata to indicate user's routing number.

[**TRANSITION\_VIEW**](https://plaid.com/docs/link/web/#link-web-onevent-eventName-TRANSITION-VIEW)

The TRANSITION\_VIEW event indicates that the user has moved from one view to the next.

[**VERIFY\_PHONE**](https://plaid.com/docs/link/web/#link-web-onevent-eventName-VERIFY-PHONE)

The user has successfully verified their phone number using OTP. This event is only emitted in the Link Remember Me flow.

[**VIEW\_DATA\_TYPES**](https://plaid.com/docs/link/web/#link-web-onevent-eventName-VIEW-DATA-TYPES)

The user has viewed data types on the data transparency consent pane.

[**metadata**](https://plaid.com/docs/link/web/#link-web-onevent-metadata)

object

An object containing information about the event.

Hide object

[**account\_number\_mask**](https://plaid.com/docs/link/web/#link-web-onevent-metadata-account-number-mask)

nullablestring

The account number mask extracted from the user-provided account number. If the user-inputted account number is four digits long, account\_number\_mask is empty. Emitted by SUBMIT\_ACCOUNT\_NUMBER.

[**error\_type**](https://plaid.com/docs/link/web/#link-web-onevent-metadata-error-type)

nullablestring

The error type that the user encountered. Emitted by: ERROR, EXIT.

[**error\_code**](https://plaid.com/docs/link/web/#link-web-onevent-metadata-error-code)

nullablestring

The error code that the user encountered. Emitted by ERROR, EXIT.

[**error\_message**](https://plaid.com/docs/link/web/#link-web-onevent-metadata-error-message)

nullablestring

The error message that the user encountered. Emitted by: ERROR, EXIT.

[**exit\_status**](https://plaid.com/docs/link/web/#link-web-onevent-metadata-exit-status)

nullablestring

The status key indicates the point at which the user exited the Link flow. Emitted by: EXIT

[**institution\_id**](https://plaid.com/docs/link/web/#link-web-onevent-metadata-institution-id)

nullablestring

The ID of the selected institution. Emitted by: all events.

[**institution\_name**](https://plaid.com/docs/link/web/#link-web-onevent-metadata-institution-name)

nullablestring

The name of the selected institution. Emitted by: all events.

[**institution\_search\_query**](https://plaid.com/docs/link/web/#link-web-onevent-metadata-institution-search-query)

nullablestring

The query used to search for institutions. Emitted by: SEARCH\_INSTITUTION.

[**is\_update\_mode**](https://plaid.com/docs/link/web/#link-web-onevent-metadata-is-update-mode)

nullablestring

Indicates if the current Link session is an update mode session. Emitted by: OPEN.

[**match\_reason**](https://plaid.com/docs/link/web/#link-web-onevent-metadata-match-reason)

nullablestring

The reason this institution was matched. This will be either returning\_user or routing\_number if emitted by: MATCHED\_SELECT\_INSTITUTION. Otherwise, this will be SAVED\_INSTITUTION or AUTO\_SELECT\_SAVED\_INSTITUTION if emitted by: SELECT\_INSTITUTION.

[**routing\_number**](https://plaid.com/docs/link/web/#link-web-onevent-metadata-routing-number)

nullablestring

The routing number submitted by user at the micro-deposits routing number pane. Emitted by SUBMIT\_ROUTING\_NUMBER.

[**mfa\_type**](https://plaid.com/docs/link/web/#link-web-onevent-metadata-mfa-type)

nullablestring

If set, the user has encountered one of the following MFA types: code, device, questions, selections. Emitted by: SUBMIT\_MFA and TRANSITION\_VIEW when view\_name is MFA

[**view\_name**](https://plaid.com/docs/link/web/#link-web-onevent-metadata-view-name)

nullablestring

The name of the view that is being transitioned to. Emitted by: TRANSITION\_VIEW.

Hide object

[**ACCEPT\_TOS**](https://plaid.com/docs/link/web/#link-web-onevent-metadata-view-name-ACCEPT-TOS)

The view showing Terms of Service in the identity verification flow.

[**CONNECTED**](https://plaid.com/docs/link/web/#link-web-onevent-metadata-view-name-CONNECTED)

The user has connected their account.

[**CONSENT**](https://plaid.com/docs/link/web/#link-web-onevent-metadata-view-name-CONSENT)

We ask the user to consent to the privacy policy.

[**CREDENTIAL**](https://plaid.com/docs/link/web/#link-web-onevent-metadata-view-name-CREDENTIAL)

Asking the user for their account credentials.

[**DATA\_TRANSPARENCY**](https://plaid.com/docs/link/web/#link-web-onevent-metadata-view-name-DATA-TRANSPARENCY)

We disclose the data types being shared.

[**DATA\_TRANSPARENCY\_CONSENT**](https://plaid.com/docs/link/web/#link-web-onevent-metadata-view-name-DATA-TRANSPARENCY-CONSENT)

We ask the user to consent to the privacy policy and disclose data types being shared.

[**DOCUMENTARY\_VERIFICATION**](https://plaid.com/docs/link/web/#link-web-onevent-metadata-view-name-DOCUMENTARY-VERIFICATION)

The view requesting document verification in the identity verification flow (configured via "Fallback Settings" in the "Rulesets" section of the template editor).

[**ERROR**](https://plaid.com/docs/link/web/#link-web-onevent-metadata-view-name-ERROR)

An error has occurred.

[**EXIT**](https://plaid.com/docs/link/web/#link-web-onevent-metadata-view-name-EXIT)

Confirming if the user wishes to close Link.

[**KYC\_CHECK**](https://plaid.com/docs/link/web/#link-web-onevent-metadata-view-name-KYC-CHECK)

The view representing the "know your customer" step in the identity verification flow.

[**LOADING**](https://plaid.com/docs/link/web/#link-web-onevent-metadata-view-name-LOADING)

Link is making a request to our servers.

[**MATCHED\_CONSENT**](https://plaid.com/docs/link/web/#link-web-onevent-metadata-view-name-MATCHED-CONSENT)

We ask the matched user to consent to the privacy policy and SMS terms.

[**MATCHED\_CREDENTIAL**](https://plaid.com/docs/link/web/#link-web-onevent-metadata-view-name-MATCHED-CREDENTIAL)

We ask the matched user for their account credentials to a matched institution.

[**MATCHED\_MFA**](https://plaid.com/docs/link/web/#link-web-onevent-metadata-view-name-MATCHED-MFA)

We ask the matched user for MFA authentication to verify their identity.

[**MFA**](https://plaid.com/docs/link/web/#link-web-onevent-metadata-view-name-MFA)

The user is asked by the institution for additional MFA authentication.

[**NUMBERS**](https://plaid.com/docs/link/web/#link-web-onevent-metadata-view-name-NUMBERS)

The user is asked to insert their account and routing numbers.

[**OAUTH**](https://plaid.com/docs/link/web/#link-web-onevent-metadata-view-name-OAUTH)

The user is informed they will authenticate with the financial institution via OAuth.

[**RECAPTCHA**](https://plaid.com/docs/link/web/#link-web-onevent-metadata-view-name-RECAPTCHA)

The user was presented with a Google reCAPTCHA to verify they are human.

[**RISK\_CHECK**](https://plaid.com/docs/link/web/#link-web-onevent-metadata-view-name-RISK-CHECK)

The risk check step in the identity verification flow (configured via "Risk Rules" in the "Rulesets" section of the template editor).

[**SCREENING**](https://plaid.com/docs/link/web/#link-web-onevent-metadata-view-name-SCREENING)

The watchlist screening step in the identity verification flow.

[**SELECT\_ACCOUNT**](https://plaid.com/docs/link/web/#link-web-onevent-metadata-view-name-SELECT-ACCOUNT)

We ask the user to choose an account.

[**SELECT\_AUTH\_TYPE**](https://plaid.com/docs/link/web/#link-web-onevent-metadata-view-name-SELECT-AUTH-TYPE)

The user is asked to choose whether to Link instantly or manually (i.e., with micro-deposits).

[**SELECT\_BRAND**](https://plaid.com/docs/link/web/#link-web-onevent-metadata-view-name-SELECT-BRAND)

The user is asked to select a brand, e.g. Bank of America. The brand selection interface occurs before the institution select pane and is only provided for large financial institutions with multiple online banking portals.

[**SELECT\_INSTITUTION**](https://plaid.com/docs/link/web/#link-web-onevent-metadata-view-name-SELECT-INSTITUTION)

We ask the user to choose their institution.

[**SELECT\_SAVED\_ACCOUNT**](https://plaid.com/docs/link/web/#link-web-onevent-metadata-view-name-SELECT-SAVED-ACCOUNT)

The user is asked to select their saved accounts and/or new accounts for linking in the Link Remember Me flow.

[**SELECT\_SAVED\_INSTITUTION**](https://plaid.com/docs/link/web/#link-web-onevent-metadata-view-name-SELECT-SAVED-INSTITUTION)

The user is asked to pick a saved institution or link a new one in the Link Remember Me flow.

[**SELFIE\_CHECK**](https://plaid.com/docs/link/web/#link-web-onevent-metadata-view-name-SELFIE-CHECK)

The view in the identity verification flow which uses the camera to confirm there is real user that matches their ID documents.

[**SUBMIT\_PHONE**](https://plaid.com/docs/link/web/#link-web-onevent-metadata-view-name-SUBMIT-PHONE)

The user is asked for their phone number in the Link Remember Me flow.

[**UPLOAD\_DOCUMENTS**](https://plaid.com/docs/link/web/#link-web-onevent-metadata-view-name-UPLOAD-DOCUMENTS)

The user is asked to upload documents (for Income verification).

[**VERIFY\_PHONE**](https://plaid.com/docs/link/web/#link-web-onevent-metadata-view-name-VERIFY-PHONE)

The user is asked to verify their phone OTP in the Link Remember Me flow.

[**VERIFY\_SMS**](https://plaid.com/docs/link/web/#link-web-onevent-metadata-view-name-VERIFY-SMS)

The SMS verification step in the identity verification flow.

[**request\_id**](https://plaid.com/docs/link/web/#link-web-onevent-metadata-request-id)

string

The request ID for the last request made by Link. This can be shared with Plaid Support to expedite investigation. Emitted by: all events.

[**link\_session\_id**](https://plaid.com/docs/link/web/#link-web-onevent-metadata-link-session-id)

string

The link\_session\_id is a unique identifier for a single session of Link. It's always available and will stay constant throughout the flow. Emitted by: all events.

[**timestamp**](https://plaid.com/docs/link/web/#link-web-onevent-metadata-timestamp)

string

An ISO 8601 representation of when the event occurred. For example 2017-09-14T14:42:19.350Z. Emitted by: all events.

[**selection**](https://plaid.com/docs/link/web/#link-web-onevent-metadata-selection)

nullablestring

Either the verification method for a matched institution selected by the user or the Auth Type Select flow type selected by the user. If selection is used to describe selected verification method, then possible values are phoneotp or password; if selection is used to describe the selected Auth Type Select flow, then possible values are flow\_type\_manual or flow\_type\_instant. Emitted by: MATCHED\_SELECT\_VERIFY\_METHOD and SELECT\_AUTH\_TYPE.

onEvent example

**JavaScript**

**Select Language**

**Copy**

1const handler = Plaid.create({

2 ...,

3 onEvent: (eventName, metadata) => {

4 // send event and metadata to self-hosted analytics

5 analytics.send(eventName, metadata);

6 },

7});

Metadata schema

**Copy**

1{

2 error\_type: 'ITEM\_ERROR',

3 error\_code: 'INVALID\_CREDENTIALS',

4 error\_message: 'the credentials were not correct',

5 exit\_status: null,

6 institution\_id: 'ins\_4',

7 institution\_name: 'Wells Fargo',

8 institution\_search\_query: 'wellsf',

9 mfa\_type: null,

10 view\_name: 'ERROR'

11 request\_id: 'm8MDnv9okwxFNBV',

12 link\_session\_id: '30571e9b-d6c6-42ee-a7cf-c34768a8f62d',

13 timestamp: '2017-09-14T14:42:19.350Z',

14 selection: null,

15}

#### [open()](https://plaid.com/docs/link/web/" \l "open)

Calling open will display the Consent Pane view to your user, starting the Link flow. Once open is called, you will begin receiving events via the [**onEvent** callback](https://plaid.com/docs/link/web/#onevent).

open example

**JavaScript**

**Select Language**

**Copy**

1const handler = Plaid.create({ ... });

2

3// Open Link

4handler.open();

#### [exit()](https://plaid.com/docs/link/web/" \l "exit)

The exit function allows you to programmatically close Link. Calling exit will trigger either the [**onExit**](https://plaid.com/docs/link/web/#onexit) or [**onSuccess**](https://plaid.com/docs/link/web/#onsuccess) callbacks.  
The exit function takes a single, optional argument, a configuration Object.

[**force**](https://plaid.com/docs/link/web/#link-web-exit-force)

boolean

If true, Link will exit immediately. If false, or the option is not provided, an exit confirmation screen may be presented to the user.

Graceful exit example

**JavaScript**

**Select Language**

**Copy**

1const handler = Plaid.create({ ... });

2

3// Graceful exit - Link may display a confirmation screen

4// depending on how far the user is in the flow

5handler.exit();

Forced exit example

**JavaScript**

**Select Language**

**Copy**

1const handler = Plaid.create({ ... });

2

3// Force exit - Link exits immediately

4handler.exit({ force: true });

#### [destroy()](https://plaid.com/docs/link/web/" \l "destroy)

The destroy function allows you to destroy the Link handler instance, properly removing any DOM artifacts that were created by it. Use destroy() when creating new replacement Link handler instances in the onExit callback.

Destroy example

**JavaScript**

**Select Language**

**Copy**

1const handler = Plaid.create({ ... })

2

3// Destroy and clean up the Link handler & iFrame

4handler.destroy();

#### [OAuth](https://plaid.com/docs/link/web/" \l "oauth)

Using Plaid Link with an OAuth flow requires some additional setup instructions. For details, see the [OAuth Guide](https://plaid.com/docs/link/oauth/).

#### [Supported browsers](https://plaid.com/docs/link/web/" \l "supported-browsers)

Plaid officially supports Link on the latest versions of Chrome, Firefox, Safari, and Edge. Browsers are supported on Windows, Mac, Linux, iOS, and Android. Previous browser versions are also supported, as long as they are actively maintained; Plaid does not support browser versions that are no longer receiving patch updates, or that have been assigned official end of life (EOL) or end of support (EOS) status.

Ad-blocking software is not officially supported with Link web, and some ad-blockers have known to cause conflicts with Link.

#### [Example code in Plaid Pattern](https://plaid.com/docs/link/web/" \l "example-code-in-plaid-pattern)

For a real-life example of using Plaid Link for React, see [LaunchLink.tsx](https://github.com/plaid/pattern/blob/master/client/src/components/LaunchLink.tsx). This file illustrates the code for implementation of Plaid Link for React for the Node-based [Plaid Pattern](https://github.com/plaid/pattern) sample app.

# OAuth Guide

#### Configure Link to connect to institutions via OAuth

Prefer to learn by watching? Video guides are available for this topic.

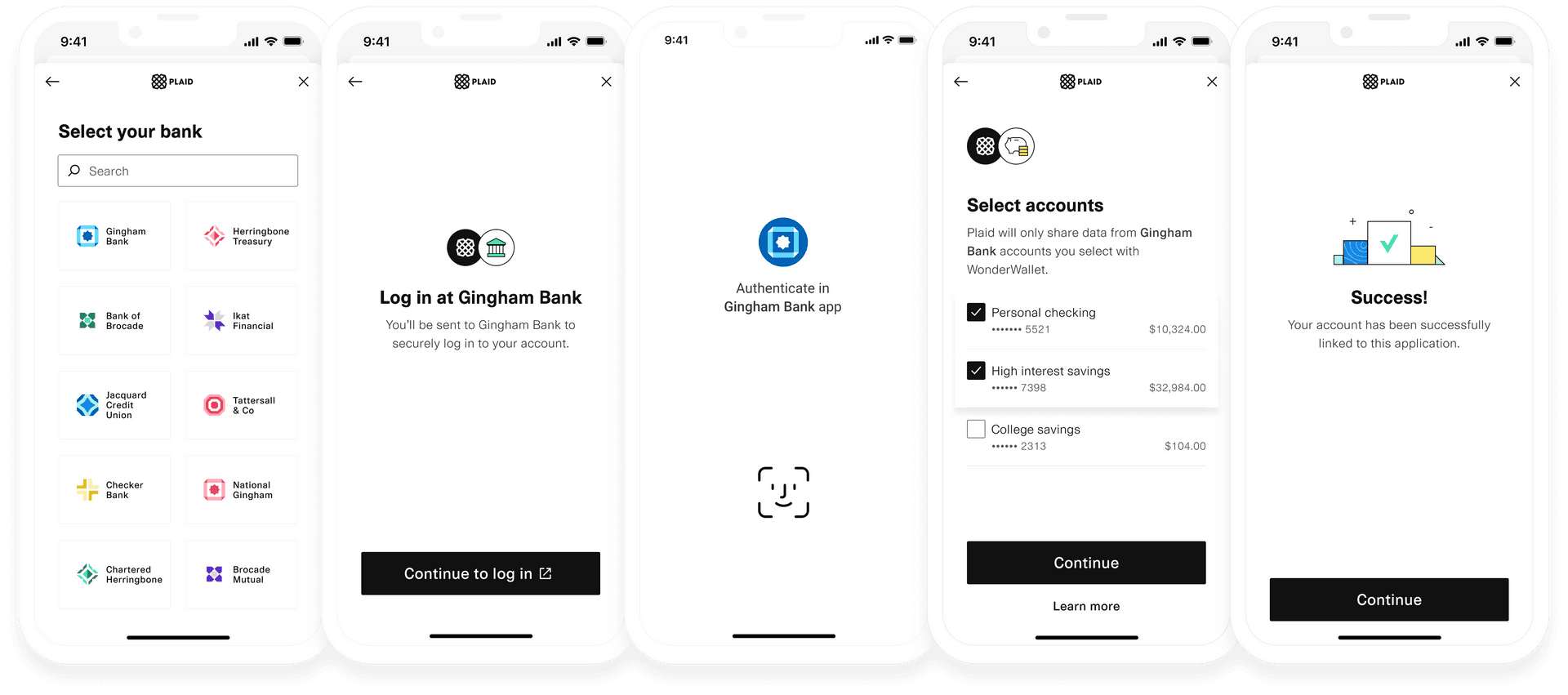
* [iOS Video Tutorial: OAuth section](https://youtu.be/9fgmW38usTo&t=2608s)
* [OAuth overview and guide for web](https://youtu.be/E0GwNBFVGik)
* [OAuth video guide for Android](https://youtu.be/oM7vL49I5tc)

#### [Introduction to OAuth](https://plaid.com/docs/link/oauth/" \l "introduction-to-oauth)

OAuth support is required in all Plaid integrations that connect to financial institutions. Without OAuth support, your end users will not be able to connect accounts from institutions that require OAuth, which includes several of the largest banks in the US. OAuth setup can be skipped only if your Plaid integration is limited to products that do not connect to financial institutions (Enrich, Identity Verification, Monitor, and Document Income).

OAuth is an industry-standard framework for authorization. With OAuth, end users can grant third parties access to their data without sharing their credentials directly with the third party.

Typically, end users authenticate and permission data directly within Plaid Link when connecting their financial accounts to third party applications. With OAuth, however, end users temporarily leave Link to authenticate and authorize data sharing using the institution's website or mobile app instead. Afterward, they're redirected back to Link to complete the Link flow and return control to the third party application.

A typical OAuth flow with Plaid Link

In addition, Plaid integrations with OAuth have several benefits over the traditional, non-OAuth experience in Link, such as:

* **Familiar and trustworthy experiences** With OAuth, end users authenticate via the bank's website or mobile app, a familiar experience that can help with conversion.
* **Streamlined login experiences** Some OAuth-enabled institutions (e.g., Chase) provide an "App-to-App" experience for end users if the end user has the institution's mobile app installed on their device. App-to-App can provide alternative authentication methods to end users (e.g., Touch ID or Face ID) that can help simplify and accelerate the authentication process.
* **Greater connection uptime** You can generally expect greater connection uptime with OAuth-enabled institutions, which means fewer connection errors for end users when using Plaid Link.
* **Longer-lived connections** Items at OAuth-enabled institutions generally remain connected longer. This typically results in fewer re-authentication errors (e.g., ITEM\_LOGIN\_REQUIRED).
* **Improved MFA (multi/second-factor) support** OAuth-enabled institutions can support end user accounts that may be currently unsupported due to the end user's MFA settings.

#### [OAuth support and compatibility](https://plaid.com/docs/link/oauth/" \l "oauth-support-and-compatibility)

OAuth is supported on all platforms on which Link is supported.

Beginning January 1, 2024, Chase will no longer support in-process webview traffic via OAuth. The process described in this guide for webview integrations will walk you through creating compliant, out-of-process webview integrations. To learn more about how to convert existing in-process webview integrations, see the [in-process webview deprecation notice](https://plaid.docsend.com/view/h3qdupjusiwyjvv5).

Plaid supports the OAuth2 protocol. For a list of the largest Plaid-supported institutions that use OAuth in the US, consult the [OAuth institutions page](https://dashboard.plaid.com/settings/compliance/us-oauth-institutions). For a full list of institutions, call [**/institutions/get**](https://plaid.com/docs/api/institutions/#institutionsget) endpoint with your desired country\_codes and the oauth option set to true. Note that for an institution where a migration to OAuth is in progress, some Items may use OAuth, while other Items at the same institution may not.

#### [Prerequisites for adding OAuth support](https://plaid.com/docs/link/oauth/" \l "prerequisites-for-adding-oauth-support)

##### [Ensure you have implemented Link tokens](https://plaid.com/docs/link/oauth/" \l "ensure-you-have-implemented-link-tokens)

OAuth requires the use of Link tokens. If you are using a legacy implementation with public keys rather than Link tokens, see the [Link token migration guide](https://plaid.com/docs/link/link-token-migration-guide/).

##### [Request Production access from Plaid](https://plaid.com/docs/link/oauth/" \l "request-production-access-from-plaid)

[Production access](https://dashboard.plaid.com/overview/production) is a prerequisite for supporting OAuth. Plaid will contact you once your account has been enabled for Production.

In the US and Canada, OAuth requires Production access; you cannot connect to US OAuth institutions in the Development environment without having received both OAuth approval and Production approval. You can, however, test OAuth in the Sandbox environment, using Sandbox-only institutions, without needing Production approval.

##### [Complete the registration requirements](https://plaid.com/docs/link/oauth/" \l "complete-the-registration-requirements)

Before implementing support for OAuth institutions, be sure to [complete the registration requirements](https://dashboard.plaid.com/settings/compliance/us-oauth-institutions) in the Plaid Dashboard.

* **Application display information** – This is public information that end users of your application will see when managing connections between your application and their bank accounts, including during OAuth flows. This information helps end users understand what your application is and why it is requesting access, which can improve conversion. In addition, some US institutions require your profile to be completed and will not allow apps with an empty profile to access their OAuth implementations.
* **Company information** – Information about your company. This information is not shared with end users of your application and is only accessible to Plaid, members of your team, and financial institutions you register with.

If you later need to update your company information or application display information, you can do so at any time via the Dashboard.

* **Plaid Master Services Agreement** – (US/CA only) Your latest contract with Plaid. If this is marked as incomplete, please reach out to your account manager or [contact support](https://dashboard.plaid.com/support/new/product-and-development/account-administration/oauth-registration) for an updated version.
* **Plaid security questionnaire** – (US/CA only) You must complete a questionnaire about your company's risk and security practices before accessing certain bank APIs. Because it may take some time for Plaid to review, it is recommended that you submit this questionnaire as early as possible in the integration process. If your Plaid integration process is otherwise complete but your security questionnaire has not yet been approved, contact your account manager or [submit a Support ticket](https://dashboard.plaid.com/support/new/product-and-development/account-administration/oauth-registration).

#### [Implementing OAuth support](https://plaid.com/docs/link/oauth/" \l "implementing-oauth-support)

##### [Required steps](https://plaid.com/docs/link/oauth/" \l "required-steps)

1. Wait for OAuth approval from Plaid, which can be tracked on the [OAuth institution page](https://dashboard.plaid.com/settings/compliance/us-oauth-institutions).
2. Understand [institution-specific behaviors](https://plaid.com/docs/link/oauth/#institution-specific-behaviors) and, if necessary, update your app to support them.

##### [Additional required steps for mobile implementations](https://plaid.com/docs/link/oauth/" \l "additional-required-steps-for-mobile-implementations)

1. [Create and register a redirect URI](https://plaid.com/docs/link/oauth/#create-and-register-a-redirect-uri)
2. [Generate a Link token and configure it with your redirect URI](https://plaid.com/docs/link/oauth/#configure-your-link-token-with-your-redirect-uri)
3. [(Mobile web and webview integrations only) Support sessions launched in embedded browsers by reinitializing Link at your redirect URI](https://plaid.com/docs/link/oauth/#reinitializing-link)

##### [Recommended, optional steps](https://plaid.com/docs/link/oauth/" \l "recommended-optional-steps)

1. [Handle Link OAuth events](https://plaid.com/docs/link/oauth/#handling-link-events)
2. [Listen for consent expiration webhooks](https://plaid.com/docs/link/oauth/#refreshing-item-consent)
3. [Manage consent revocation](https://plaid.com/docs/link/oauth/#managing-consent-revocation)
4. [(US/CA only) Enable OAuth and migrate users](https://plaid.com/docs/link/oauth/#enabling-oauth-connections-and-migrating-users)
5. [(Europe only) Enable QR Code authentication](https://plaid.com/docs/link/oauth/#qr-code-authentication)

#### [Create and register a redirect URI](https://plaid.com/docs/link/oauth/" \l "create-and-register-a-redirect-uri)

After successfully completing the OAuth flow via their bank's website or app, you'll need to redirect the end user back to your application. This is accomplished with a redirect URI that you'll need to set up and configure accordingly depending on your client platform.

**Constructing valid redirect URIs**

Redirect URIs must use HTTPS. The only exception is on Sandbox, where, for testing purposes, redirect URIs pointing to localhost are allowed over HTTP. Custom URI schemes are not supported in any environment. Subdomain wildcards are supported using a \* character. For example, adding https://\*.example.com/oauth.html to the allowlist permits https://oauth1.example.com/oauth.html, https://oauth2.example.com/oauth.html, etc. Subdomain wildcards can only be used for domains that you control and are not allowed for domains on the [Public Suffix List](https://publicsuffix.org/list/). For example, https://\*.co.uk/oauth.html is not a valid subdomain wildcard. Redirect URIs do not support hash routing, so your URI cannot contain a '#' symbol.

Note: Do not enter a wildcard (\*) when specifying a redirect\_uri in the call to [**/link/token/create**](https://plaid.com/docs/api/tokens/#linktokencreate). Wildcards are reserved for the allowlist on the dashboard only.

##### [Desktop web, mobile web, React, or Webview](https://plaid.com/docs/link/oauth/" \l "desktop-web-mobile-web-react-or-webview)

For desktop web, mobile web, or React, the redirect URI is typically the address of a blank web page you'll need to create and host. This web page will be used to allow the end user to resume and complete the Link flow after completing the OAuth flow on their bank's website or app. https://example.com/oauth-page.html is an example of a typical redirect URI. After creating your redirect URI, add it to the [Allowed redirect URIs](https://dashboard.plaid.com/developers/api).

**Redirect URIs and desktop & mobile web**

OAuth flows will function properly on web even if you don't set up a redirect URI. Desktop web and mobile web integrations will always try to open the OAuth bank's website in a new pop-up window if possible (or in a new tab on mobile web), regardless of whether a redirect\_uri is provided. However, not providing a redirect URI will prevent mobile web users from using your integration through a webview browser (a browser launched via Mail, Facebook, Google Maps, etc.) because those browsers often do not support pop-ups. To provide the best experience for end users on mobile web, always specify a redirect URI and reinitialize Link.  
Setting a redirect\_uri is still required for Link web SDK integrations within a mobile application (e.g within a webview) because those integrations still use the redirect OAuth flow.

##### [iOS SDK, React Native (iOS)](https://plaid.com/docs/link/oauth/" \l "ios-sdk-react-native-ios)

To maintain support for Chase OAuth flows, all integrations must upgrade to version 4.1.0 of the Link iOS SDK or version 9.0.1 of the React Native SDK (released January 2023) by January 1, 2024.

For iOS SDK or React Native (iOS), the redirect URI is typically the address of a blank web page you'll need to create and host. You'll need to configure an [Apple App Association File](https://developer.apple.com/documentation/security/password_autofill/setting_up_an_app_s_associated_domains) to associate your redirect URI with your application. To enable [App-to-App authentication flows](https://plaid.com/docs/link/oauth/#app-to-app-authentication), you'll need to register the redirect URI as an app link. Custom URI schemes are not supported; a proper universal link **must** be used.

##### [Android SDK, React Native (Android)](https://plaid.com/docs/link/oauth/" \l "android-sdk-react-native-android)

Register your Android package by adding the Android package name(s) to the [Allowed Android package names](https://dashboard.plaid.com/developers/api) list. Plaid will automatically create your redirect URI and its contents based on your package name. When specifying a redirect URI in the following steps, you will use [**android\_package\_name**](https://plaid.com/docs/api/tokens/#link-token-create-request-android-package-name).

#### [Configure your Link token with your redirect URI](https://plaid.com/docs/link/oauth/" \l "configure-your-link-token-with-your-redirect-uri)

OAuth support requires use of Link tokens. The legacy client-side settings oauthRedirectUri and oauthNonce are ignored and will not be read. If you are still using a legacy public key, see the [Link token migration guide](https://plaid.com/docs/link/link-token-migration-guide/).

You'll need to specify your redirect URI via the redirect\_uri field when generating a Link token with [**/link/token/create**](https://plaid.com/docs/api/tokens/#linktokencreate) (on Android, use the android\_package\_name parameter to provide your Android package name instead). Use this Link token to initialize Link.

If you're using Link in [update mode](https://plaid.com/docs/link/update-mode/), ensure you specify your redirect URI via the redirect\_uri field (on Android, use the android\_package\_name parameter to provide your package name instead).

Do not use query parameters when specifying the redirect\_uri. Make sure to specify the user.client\_user\_id.

Select group for content switcher

Current librariesLegacy libraries

Generating a Link token

**Curl**

**Select Language**

**Copy**

1curl -X POST https://production.plaid.com/link/token/create \

2-H 'Content-Type: application/json' \

3-d '{

4 "client\_id": "${PLAID\_CLIENT\_ID}",

5 "secret": "${PLAID\_SECRET}",

6 "client\_name": "Plaid Test App",

7 "user": { "client\_user\_id": "${UNIQUE\_USER\_ID}" },

8 "products": ["transactions"],

9 "country\_codes": ["US"],

10 "language": "en",

11 "webhook": "https://sample.webhook.com",

12 "redirect\_uri": "https://example.com/callback"

13}'

##### [Using OAuth within an iFrame](https://plaid.com/docs/link/oauth/" \l "using-oauth-within-an-iframe)

Launching Link from within an iFrame is not recommended. Link conversion for OAuth institutions is typically up to 15 percentage points higher when using Plaid's SDKs than when using iFrames. If Link is launched from within an iFrame, you'll be unable to maintain user state. Page rendering, sizing, and data exchange may also be suboptimal.

#### [Reinitializing Link](https://plaid.com/docs/link/oauth/" \l "reinitializing-link)

After completing the OAuth flow, the end user will be redirected to your redirect URI (e.g., https://example.com/oauth-page.html). This is where they'll resume and complete the Link flow and return to your application. To do this, you'll need to reinitialize Link at your redirect URI.

Depending on your client platform, Link may require additional configuration to work with OAuth. Detailed instructions for each platform are provided below.

| **CLIENT PLATFORM** | **LINK REINITIALIZATION REQUIRED?** |
| --- | --- |
| [Desktop web](https://plaid.com/docs/link/oauth/#desktop-web-mobile-web-or-react) | No |
| [Mobile web](https://plaid.com/docs/link/oauth/#desktop-web-mobile-web-or-react) | Not required, but recommended in order to maximize Link conversion |
| [Webview](https://plaid.com/docs/link/oauth/#webview) | Yes |
| [iOS SDK](https://plaid.com/docs/link/oauth/#ios) | No |
| [React Native (iOS)](https://plaid.com/docs/link/oauth/#react-native-on-ios) | No |
| Android SDK (version 3.2.3 or later required) | No, but [app package registration required](https://plaid.com/docs/link/oauth/#android-sdk-react-native-android) |
| React Native (Android) | No, but [app package registration required](https://plaid.com/docs/link/oauth/#android-sdk-react-native-android) |

##### [Desktop web, mobile web, or React](https://plaid.com/docs/link/oauth/" \l "desktop-web-mobile-web-or-react)

A reference implementation for OAuth in React can be found in the [Plaid React GitHub](https://github.com/plaid/react-plaid-link#oauth--opening-link-without-a-button-click). If you are looking for a demonstration of a real-life app that incorporates the implementation of OAuth in React see [Plaid Pattern](https://github.com/plaid/pattern), a Node-based example app.

Desktop and mobile web sessions do not require Link reinitialization by default.

However, not supporting Link reinitialization will prevent mobile web users from using your integration through a webview (an embedded browser launched via Mail, Facebook, Google Maps, etc.). For these sessions, you'll need to launch Link twice, once before the OAuth redirect (i.e., the first Link initialization) and once after the OAuth redirect (i.e., Link reinitialization). The Link reinitialization should occur at your redirect URI.

When reinitializing Link, configure it using the same Link token you used when [initializing Link the first time](https://plaid.com/docs/link/oauth/#configure-your-link-token-with-your-redirect-uri). It is up to you to determine the best way to provide the correct link\_token upon redirect. As an example, the code sample below demonstrates the use of a browser's local storage to retrieve the Link token from the first Link initialization.

Select group for content switcher

ReactVanilla JS

/oauth-page: Reinitializing Link with receivedRedirectUri

**Copy**

1import React, { useEffect } from 'react';

2import { usePlaidLink } from 'react-plaid-link';

3

4const OAuthLink = () => {

5 // The Link token from the first Link initialization

6 const linkToken = localStorage.getItem('link\_token');

7

8 const onSuccess = React.useCallback((public\_token: string) => {

9 // send public\_token to server, retrieve access\_token and item\_id

10 // return to "https://example.com" upon completion

11 });

12

13 const onExit = (err, metadata) => {

14 // handle error...

15 };

16

17 const config: Parameters<typeof usePlaidLink>[0] = {

18 token: linkToken!,

19 // pass in the received redirect URI, which contains an OAuth state ID parameter that is required to

20 // re-initialize Link

21 receivedRedirectUri: window.location.href,

22 onSuccess,

23 onExit,

24 };

25

26 const { open, ready, error } = usePlaidLink(config);

27

28 // automatically reinitialize Link

29 useEffect(() => {

30 if (ready) {

31 open();

32 }

33 }, [ready, open]);

34

35 return <></>;

36};

37

38export default OAuthLink;

oauth-page.html: Reinitialize Link with receivedRedirectUri

**Copy**

1<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/2.2.3/jquery.min.js"></script>

2 <script src="https://cdn.plaid.com/link/v2/stable/link-initialize.js"></script>

3 <script>

4 (function ($) {

5 var linkToken = localStorage.getItem("link\_token");

6 var handler = Plaid.create({

7 token: linkToken,

8 // pass in the received redirect URI, which contains an OAuth state ID parameter that is required to

9 // re-initialize Link

10 receivedRedirectUri: window.location.href,

11 onSuccess: function (public\_token) {

12 $.post(

13 "/api/set\_access\_token",

14 { public\_token: public\_token },

15 function (data) {

16 location.href = "https://example.com";

17 }

18 );

19 },

20 });

21 handler.open();

22 })(jQuery);

23 </script>

In addition, when reinitializing Link, you should configure it with the receivedRedirectUri field and pass in the full received redirect URI, as demonstrated in the code sample. The received redirect URI is your redirect URI appended with an OAuth state ID parameter. The OAuth state ID parameter will allow you to persist user state when reinitializing Link, allowing the end user to resume the Link flow where they left off. No extra configuration or setup is needed to generate the received redirect URI. The received redirect URI is programmatically generated for you by Plaid after the end user authenticates on their bank's website or mobile app. You can retrieve it using window.location.href.

The received redirect URI must not contain any extra query parameters or fragments other than what is provided upon redirect. The standard Link callback onSuccess will be triggered as usual once the user completes the Link flow.

An example received redirect URI

**Copy**

1https://example.com/oauth-page.html?oauth\_state\_id=9d5feadd-a873-43eb-97ba-422f35ce849b`

###### Optional methods for retrieving the initial Link token

If Link is reinitialized in the same browser session as the first Link initialization, you can store the Link token in a cookie or local storage in the browser for easy access when reinitializing Link. For example, the Plaid Quickstart uses localStorage.setItem to store the token.

If Link is reinitialized in a different browser session than the first Link initialization, you can store a mapping of the Link token associated with the user (server-side). Upon opening the second browser session, authenticate the user, fetch the corresponding Link token from the server, and use it to reinitialize Link.

##### [Webview](https://plaid.com/docs/link/oauth/" \l "webview)

Beginning January 1, 2024, all webview-based integrations will need to extend the webview handler for redirects in order to support Chase OAuth. This can be accomplished with code samples for [iOS](https://github.com/plaid/plaid-link-examples/blob/master/webviews/wkwebview/wkwebview/LinkViewController.swift#L56-L72) and [Android](https://github.com/plaid/plaid-link-examples/blob/master/webviews/android/LinkWebview/app/src/main/java/com/example/linkwebview/MainActivity.kt#L89-L156) For more details, see [Extending webview instances to support certain institutions](https://plaid.com/docs/link/oauth/#extending-webview-instances-to-support-certain-institutions).

For webview, you'll need to launch Link twice, once before the OAuth redirect (i.e., the first Link initialization) and once after the OAuth redirect (i.e., Link reinitialization). The Link reinitialization should occur at your redirect URI.

For the initial Link instance, first generate a Link token as described in [Configure your Link token with your redirect URI](https://plaid.com/docs/link/oauth/#configure-your-link-token-with-your-redirect-uri), then set Link's token parameter to this Link token. After the end user successfully completes the OAuth flow via their bank's website or app, they'll be redirected to your redirect URI, where you'll reinitialize Link.

Initializing Link (first Link initialization)

**Copy**

1https://cdn.plaid.com/link/v2/stable/link.html?isWebview=true

2&token=GENERATED\_LINK\_TOKEN

When reinitializing Link, use the same Link token you generated when you first initialized Link. It is up to you to determine the best way to provide the correct link\_token upon redirect.

In addition, when reinitializing Link, you should configure it with the receivedRedirectUri field. The received redirect URI is your redirect URI appended with an OAuth state ID parameter. The OAuth state ID parameter will allow you to persist user state when reinitializing Link, allowing the end user to resume the Link flow where they left off. No extra configuration or setup is needed to generate the received redirect URI. The received redirect URI is programmatically generated after the end user authenticates on their bank's website or mobile app. The received redirect URI must not contain any extra query parameters or fragments other than what is provided upon redirect. Note that any unsafe ASCII characters in the receivedRedirectUri in the webview query string must be URL-encoded; for improved readability, the example below is shown prior to URL encoding.

Reinitializing Link

**Copy**

1https://cdn.plaid.com/link/v2/stable/link.html?isWebview=true

2&token=SAME\_GENERATED\_LINK\_TOKEN&receivedRedirectUri=https://example.com/oauth-page?oauth\_state\_id=9d5feadd-a873-43eb-97ba-422f35ce849b

###### Extending webview instances to support certain institutions

Some institutions require further modifications to work with mobile webviews. This applies to USAA on Android and Chase App-to-App on Android. Beginning on January 1, 2024, this also applies to all Chase webview integrations. For these institutions, you'll need to extend your Webview instance to override the handler for redirects. [An example function for Android](https://github.com/plaid/plaid-link-examples/blob/master/webviews/android/LinkWebview/app/src/main/java/com/example/linkwebview/MainActivity.kt#L89-L156) and [an example function for iOS](https://github.com/plaid/plaid-link-examples/blob/master/webviews/wkwebview/wkwebview/LinkViewController.swift#L56-L72) can be found within Plaid's Link examples on GitHub and can be copied for use with your app.

On Android, you will also need to support Android App links to have a valid working redirect\_uri to provide to [**/link/token/create**](https://plaid.com/docs/api/tokens/#linktokencreate). This requires creating a ./well\_known/assetlinks.json and an IntentFilter on the Android app.

##### [iOS](https://plaid.com/docs/link/oauth/" \l "ios)

To maintain support for Chase OAuth flows, all integrations must upgrade to version 4.1.0 of the [Link iOS SDK](https://github.com/plaid/plaid-link-ios) (released January 2023) by January 1, 2024.

For the initial Link instance, first generate a Link token as described in [Configure your Link token with your redirect URI](https://plaid.com/docs/link/oauth/#configure-your-link-token-with-your-redirect-uri). Your redirect\_uri must also be added to the Plaid Dashboard and should be configured as a universal link (not a custom URI) using an Apple App Association File.

Initializing Link

**Swift**

**Select Language**

**Copy**

1// With custom configuration

2let linkToken = "<#GENERATED\_LINK\_TOKEN#>"

3let onSuccess: (LinkSuccess) -> Void = { (success) in

4 // Read success.publicToken here

5 // Log/handle success.metadata here

6}

7let linkConfiguration = LinkTokenConfiguration(linkToken: linkToken, onSuccess: onSuccess)

8let handlerResult = Plaid.create(linkConfiguration)

9

10switch handlerResult {

11case .success(let handler):

12 self.handler = handler

13 handler.open(presentUsing: .viewController(self))

14case .failure(let error):

15 // Log and handle the error here.

16}

OAuth on iOS devices can occur fully within the integrating application (In-App OAuth), or it can include a transition from your application to the bank's app ([App-to-App OAuth](https://plaid.com/docs/link/oauth/#app-to-app-authentication)). App-to-App OAuth is initiated by the bank itself and is not controlled by the iOS SDK. In order to ensure your users can return to your application, you must support App-to-App OAuth.

##### [App-to-App OAuth requirements](https://plaid.com/docs/link/oauth/" \l "app-to-app-oauth-requirements)

During App-to-App OAuth, the end user is directed from your application to the bank's app to authenticate. To return the end user back to your application after they authenticate, your redirect URI must be a universal link. Once the user returns to your app, UIKit will invoke a method within your application and provide the redirect URI that triggered this return to the app.

App-to-App behavior can be [tested in the Sandbox environment](https://plaid.com/docs/link/oauth/#testing-oauth). If App-to-App does not function as intended, validate that the redirect URI used to configure Link is a valid universal link and that your application has the associated-domains entitlement for that URI.

##### [React Native on iOS](https://plaid.com/docs/link/oauth/" \l "react-native-on-ios)

To maintain support for Chase OAuth flows, all integrations must upgrade to version 9.0.1 of the [React Native SDK](https://github.com/plaid/react-native-plaid-link-sdk) (released January 2023) by January 1, 2024.

An example React Native client implementation for OAuth on iOS can be found in the [Tiny Quickstart](https://github.com/plaid/tiny-quickstart/tree/main/react_native/TinyQuickstartReactNative).

React Native on iOS uses universal links for OAuth. You will need to [create and register a redirect URI](https://plaid.com/docs/link/oauth/#create-and-register-a-redirect-uri) and [configure your Link token with your redirect URI](https://plaid.com/docs/link/oauth/#configure-your-link-token-with-your-redirect-uri) in order for OAuth to work correctly.

##### [Android SDK and Android on React Native](https://plaid.com/docs/link/oauth/" \l "android-sdk-and-android-on-react-native)

Example code for implementing OAuth on Android can be found on GitHub in the [Android SDK](https://github.com/plaid/plaid-link-android). An example React Native client implementation for OAuth on Android can be found in the [Tiny Quickstart](https://github.com/plaid/tiny-quickstart/tree/main/react_native/TinyQuickstartReactNative).

When using the Android SDK or Android on React Native, you must generate a Link token by calling [**/link/token/create**](https://plaid.com/docs/api/tokens/#linktokencreate) and passing in an android\_package\_name. Your package name (e.g., com.example.testapp) must also be added to the Plaid Dashboard. Do not pass a redirect\_uri into the [**/link/token/create**](https://plaid.com/docs/api/tokens/#linktokencreate) call. Proceed to initialize Link with the generated token.

Select group for content switcher

Current librariesLegacy libraries

Passing in an Android package name

**Copy**

1String clientUserId = "user-id";

2

3LinkTokenCreateRequestUser user = new LinkTokenCreateRequestUser()

4 .clientUserId(clientUserId)

5 .legalName("legal name")

6 .phoneNumber("4155558888")

7 .emailAddress("email@address.com");

8

9LinkTokenCreateRequest request = new LinkTokenCreateRequest()

10 .user(user)

11 .clientName("Plaid Test App")

12 .products(Arrays.asList(Products.AUTH))

13 .countryCodes(Arrays.asList(CountryCode.US))

14 .language("en")

15 .webhook("https://example.com/webhook")

16 .linkCustomizationName("default")

17 .androidPackageName("com.plaid.example")

18

19Response<LinkTokenCreateResponse> response = client()

20 .linkTokenCreate(request)

21 .execute();

22

23String linkToken = response.body().getLinkToken();

#### [Testing OAuth](https://plaid.com/docs/link/oauth/" \l "testing-oauth)

You can test OAuth on Sandbox even if Plaid has not yet enabled OAuth flows for your account. To test out the OAuth flow in the Sandbox environment, you can use Platypus OAuth Bank (ins\_127287), Platypus OAuth App2App Bank (ins\_132241) for testing App2App flows on mobile, or to emulate European flows, Flexible Platypus Open Banking (ins\_117181). These dummy institutions direct you to a Plaid sample OAuth flow that is similar to what you would see for a bank’s OAuth flow. When prompted, you can enter anything as credentials (including leaving the input fields blank) to proceed through the sample OAuth flow. Note that institution-specific OAuth flows cannot be tested in Sandbox; OAuth panes for Platypus institutions will be shown instead.

To ensure your OAuth integration works across all platforms, test it in the following scenarios before deployment:

* On each client platform that is available to your users (e.g. desktop, iOS app, iOS mobile web, Android app, Android mobile web)
* With the OAuth institution app installed, for institutions that support [app-to-app authentication](https://plaid.com/docs/link/oauth/#app-to-app-authentication) (i.e. Chase). For more details on testing app-to-app in Sandbox, see [App-to-App authentication](https://plaid.com/docs/link/oauth/#app-to-app-authentication).
* Without the OAuth institution app installed
* In [update mode](https://plaid.com/docs/link/update-mode/), by using [**/sandbox/item/reset\_login**](https://plaid.com/docs/api/sandbox/#sandboxitemreset_login) in the Sandbox environment

All environments, including Sandbox, use the pop-up flow on desktop and mobile web, unless the page is accessed through a mobile webview. To test the redirect flow, you can use your browser's developer tools to simulate running your application in a webview browser, such as Chrome WebView. On Chrome, for example, select the "Toggle Device Toolbar" option from within Chrome's Developer Tools and create a new virtual device configuration with a WebView user agent.

Example user agent

**Copy**

1Mozilla/5.0 (Linux; Android 5.1.1; Nexus 5 Build/LMY48B; wv) AppleWebKit/537.36 (KHTML, like Gecko) Version/4.0 Chrome/43.0.2357.65 Mobile Safari/537.36

In the Sandbox environment, all product-specific permissions will be granted in the OAuth flow, regardless of whether you check the boxes to share account and routing number or identity information.

#### [Troubleshooting common OAuth problems](https://plaid.com/docs/link/oauth/" \l "troubleshooting-common-oauth-problems)

For guides to troubleshooting common OAuth issues, see [Link troubleshooting](https://plaid.com/docs/link/troubleshooting/#oauth-not-working).

#### [Enabling OAuth connections and migrating users](https://plaid.com/docs/link/oauth/" \l "enabling-oauth-connections-and-migrating-users)

Your integration will automatically convert to using OAuth connections for a given institution on the date that Plaid has indicated to you. You may also have the option to migrate to OAuth earlier; if this option is available, a button to enable OAuth for the institution will appear on the [OAuth institution page](https://dashboard.plaid.com/settings/compliance/us-oauth-institutions).

Your existing Items with the institution being migrated will automatically be moved into the ITEM\_LOGIN\_REQUIRED state. This will happen gradually over a migration period (by default, 90 days, but the duration may vary depending on the institution), starting on or shortly after the OAuth enablement date. Completing the [update mode](https://plaid.com/docs/link/update-mode/) flow for these Items will convert them to use OAuth connections. To avoid any disruption in connectivity, you can also prompt your users to complete the update mode flow as soon as you have been enabled for OAuth with their institution.

#### [Institution-specific behaviors](https://plaid.com/docs/link/oauth/" \l "institution-specific-behaviors)

Some institutions have unique behaviors when used with OAuth connections. Note that these behaviors are standard for any connection using these institutions' APIs and not specific to their integration with Plaid.

The behaviors listed below are the ones most likely to require changes to your application's business logic; for a more exhaustive list of institution-specific OAuth details, see the "Bank-specific documentation" section of the [OAuth integration PDF guide](https://dashboard.plaid.com/oauth-guide).

##### [Chase](https://plaid.com/docs/link/oauth/" \l "chase)

When used with Auth, Chase will return "tokenized" routing and account numbers, which can be used for ACH transactions but are not the user's actual account and routing numbers. The mask, however, will continue to be based on the actual account number. For this reason, when displaying account numbers from Chase to the user to help them identify their account in your UI, always use the mask rather than displaying the account number. If a user [revokes their permissions](https://plaid.com/docs/link/oauth/#managing-consent-revocation) to your app, the tokenized numbers will no longer work. For more details, see the [API reference](https://plaid.com/docs/api/products/auth/#auth-get-response-numbers-ach).

[Update mode](https://plaid.com/docs/link/update-mode/#using-update-mode-to-request-new-accounts) cannot be used to remove accounts or permissions from a Chase Item. If your end user wishes to revoke or limit Plaid's access to a Chase account, they must do so via the Chase Security Center within the Chase online banking portal. This occurs even across Items; if a Chase Item is deleted and a new Item is created for your app using the same credentials, the previously selected permissions will persist to the new Item.

Existing Chase OAuth Items will be invalidated when a new public token is created using the same credentials. For more details, see [Preventing duplicate Items](https://plaid.com/docs/link/duplicate-items/#preventing-duplicate-items).

##### [Capital One](https://plaid.com/docs/link/oauth/" \l "capital-one)

When calling [**/accounts/balance/get**](https://plaid.com/docs/api/products/balance/#accountsbalanceget) for a Capital One non-depository account, such as a credit card or loan account, you will need to specify how fresh you require the balance data to be. If balance data meeting your requirements is not available, the call will fail and you will not be billed. For more details, see the [API Reference](https://plaid.com/docs/api/products/balance/#accounts-balance-get-request-options-min-last-updated-datetime). For similar reasons, [**/transactions/refresh**](https://plaid.com/docs/api/products/transactions/#transactionsrefresh) will result in a PRODUCT\_NOT\_SUPPORTED error if used on a Capital One Item.

##### [Charles Schwab](https://plaid.com/docs/link/oauth/" \l "charles-schwab)

Existing Schwab OAuth Items will be invalidated when a new public token is created using the same credentials. For more details, see [Preventing duplicate Items](https://plaid.com/docs/link/duplicate-items/#preventing-duplicate-items).

#### [Handling Link events](https://plaid.com/docs/link/oauth/" \l "handling-link-events)

OAuth flows have a different sequence of [Link events](https://plaid.com/docs/link/web/#onevent) than non-OAuth flows. If you are using Link events to measure conversion metrics for completing the Link process, you may need to handle these events differently when using OAuth.

In addition, the flow itself may be different if you are initiating OAuth with a redirect URI or displaying the OAuth screen in a separate pop-up window.

The events fired for a non-OAuth flow might look something like this:

Typical non-OAuth event flow

**Copy**

1OPEN (view\_name = CONSENT)

2TRANSITION\_VIEW view\_name = SELECT\_INSTITUTION)

3SELECT\_INSTITUTION

4TRANSITION\_VIEW (view\_name = CREDENTIAL)

5SUBMIT\_CREDENTIALS

6TRANSITION\_VIEW (view\_name = LOADING)

7TRANSITION\_VIEW (view\_name = MFA, mfa\_type = code)

8SUBMIT\_MFA (mfa\_type = code)

9TRANSITION\_VIEW (view\_name = LOADING)

10TRANSITION\_VIEW (view\_name = CONNECTED)

11HANDOFF

12onSuccess

The events fired for a typical OAuth flow may look more like the following:

Typical OAuth event flow when using a redirect\_uri

**Copy**

1OPEN (view\_name = CONSENT)

2TRANSITION\_VIEW (view\_name = SELECT\_INSTITUTION)

3SELECT\_INSTITUTION

4TRANSITION\_VIEW (view\_name = OAUTH)

5OPEN\_OAUTH

6...

7(The user completes the OAuth flow at their bank)

8...

9TRANSITION\_VIEW (view\_name = CONNECTED)

10HANDOFF

11onSuccess

Link does not issue the SUBMIT\_CREDENTIALS event when a user authenticates with an institution that requires OAuth. Link issues the OPEN\_OAUTH event when a user chooses to be redirected to the institution’s OAuth portal. It is recommended to track this event instead of SUBMIT\_CREDENTIALS.

In most situations, if Plaid encounters an error from the bank's OAuth flow, or if the user chooses to not grant access via their OAuth login attempt, the user will return to your application and Link will fire an EXIT event with a requires\_oauth exit status. You may also see an ERROR event, depending on the type of error that Plaid encountered.

Link event flow after returning with an error

**Copy**

1SELECT\_INSTITUTION

2OPEN (view\_name = null)

3ERROR (error\_code = INTERNAL\_SERVER\_ERROR)

4TRANSITION\_VIEW (view\_name = ERROR)

5TRANSITION\_VIEW (view\_name = EXIT)

6EXIT (exit\_status = requires\_oauth)

If the user closes the bank's OAuth window without completing the OAuth flow, Link will fire a CLOSE\_OAUTH event. This only happens in situations where the OAuth flow appears in a pop-up window.

If the OAuth flow times out while waiting for the user to sign in, Link will fire a FAIL\_OAUTH event. This only happens in situations where the OAuth flow appears in a pop-up window.

Once you receive the onSuccess callback from an OAuth flow, the integration steps going forward are the same as for non-OAuth flows.

#### [Refreshing Item consent](https://plaid.com/docs/link/oauth/" \l "refreshing-item-consent)

Some institutions require users to periodically re-affirm their consent to avoid Item expiration. When using OAuth, end users may need to refresh their access-consent after a certain amount of time, such as one year or three months. This is particularly common in Europe, where access-consent typically expires after 90 days. Expiring access-consent is less common in the US, and consent periods are typically longer. For more details on which US institutions expire access-consent and how frequently, see the "Bank-specific documentation" section of the [OAuth integration PDF guide](https://dashboard.plaid.com/oauth-guide).

To determine when a user will need to re-authenticate, make a request to the [**/item/get**](https://plaid.com/docs/api/items/#itemget) endpoint and note the consent\_expiration\_time field. Plaid will also send a [**PENDING\_EXPIRATION**](https://plaid.com/docs/api/items/#pending_expiration) webhook one week before a user’s access-consent is set to expire. In order to continue receiving data for that user without interruption, ensure they re-authenticate via [update mode](https://plaid.com/docs/link/update-mode/) prior to that date. When using Link in update mode, be sure to specify your redirect URI via the redirect\_uri field as described in [Configure your Link token with your redirect URI](https://plaid.com/docs/link/oauth/#configure-your-link-token-with-your-redirect-uri).

If consent is not refreshed before the Item expires, the Item will enter the ITEM\_LOGIN\_REQUIRED error state. Sending the Item through update mode, as described above, will resolve the error.

For a real-life example of handling the PENDING\_EXPIRATION webhook and update mode, see [handleItemWebhook.js](https://github.com/plaid/pattern/blob/master/server/webhookHandlers/handleItemWebhook.js#L69-L86), [linkTokens.js](https://github.com/plaid/pattern/blob/master/server/routes/linkTokens.js#L30-L35) and [launchLink.tsx](https://github.com/plaid/pattern/blob/master/client/src/components/LaunchLink.tsx#L42). These files illustrate the code for handling of webhooks and update mode for Plaid Link for React for the Node-based [Plaid Pattern](https://github.com/plaid/pattern) sample app.

#### [Managing consent revocation](https://plaid.com/docs/link/oauth/" \l "managing-consent-revocation)

Many institutions that support OAuth provide a means for the end user to revoke consent via their website. If an end user revokes consent, the Item will enter an ITEM\_LOGIN\_REQUIRED state after approximately 24-48 hours.

A user may also revoke consent for a single account, without revoking consent for the entire Item. Accounts in this situation are treated the same as accounts that have been closed: no webhook will be fired, you will stop receiving transactions associated with the account, and Plaid will stop returning the account in API responses. Access can be re-authorized via the [update mode](https://plaid.com/docs/link/update-mode/) flow.

When a user revokes access to an Item or an account, we recommend giving them the opportunity to either verify that they intended to remove it or to indicate that the revocation was unintentional. If the user did not mean to revoke access, they can re-authorize access by going through the standard [update mode](https://plaid.com/docs/link/update-mode/) flow. If the user verifies that their action was intentional or does not respond after a few days, delete the associated data, and, in the case of a removed Item, use [**/item/remove**](https://plaid.com/docs/api/items/#itemremove) to delete the Item altogether.

When using Link in update mode (for either case described in this section), be sure to specify your redirect URI via the redirect\_uri field as described in [Configure your Link token with your redirect URI](https://plaid.com/docs/link/oauth/#configure-your-link-token-with-your-redirect-uri).

#### [Partial consent](https://plaid.com/docs/link/oauth/" \l "partial-consent)

OAuth can provide the ability for end users to configure granular permissions on their Items. For example, an end user may allow access to a checking account but not a credit card account behind the same login, or may allow an institution to share only certain account information, such as identity data but not transaction history.

Before handing the user off to the institution's OAuth flow, Plaid will provide guidance within Link recommending which permissions the user needs to grant based on which products you have initialized Link with. In Production and Development, this guidance will be tailored to use the same wording that the institution's OAuth flow uses. This guidance will appear both on initial link and during the [update mode flow](https://plaid.com/docs/link/update-mode/). For more details on the exact appearance and messaging used, see the [OAuth Guide](https://dashboard.plaid.com/settings/compliance/us-oauth-institutions) within the Dashboard.

If an end user chooses not to share data that is required by your Link token's products or required\_if\_supported\_products configuration, or does not share access to any accounts, Link will show an error, and they will be prompted to restart the Link flow.

Note that if your app calls [**/link/token/create**](https://plaid.com/docs/api/tokens/#linktokencreate) using an account\_filter parameter to limit the account types that can be used with Link, the filter will only be applied after the OAuth flow has been completed and will not affect the permission selection interface within the OAuth flow.

If you do not have the product-specific OAuth permissions required to use a specific endpoint with an Item, you will receive an [**ACCESS\_NOT\_GRANTED**](https://plaid.com/docs/errors/item/#access_not_granted) error. If you are missing permissions for an account on the Item, you may receive a product-specific error indicating that a compatible account could not be found, such as [**NO\_AUTH\_ACCOUNTS**](https://plaid.com/docs/errors/item/#no_auth_accounts-or-no-depository-accounts), [**NO\_INVESTMENT\_ACCOUNTS**](https://plaid.com/docs/errors/item/#no_investment_accounts), or [**NO\_LIABILITY\_ACCOUNTS**](https://plaid.com/docs/errors/item/#no_liability_accounts).

If your app later needs to request access to a product or account that was not originally granted for that Item during Link, you can send the user to the [update mode](https://plaid.com/docs/link/update-mode/) flow to authorize additional permissions. When using Link in update mode, be sure to specify your redirect URI via the redirect\_uri field as described in [Configure your Link token with your redirect URI](https://plaid.com/docs/link/oauth/#configure-your-link-token-with-your-redirect-uri).

#### [App-to-App authentication](https://plaid.com/docs/link/oauth/" \l "app-to-app-authentication)

Some banks (i.e. Chase) support an App-to-App experience if the user is authenticating on their mobile device and has the bank's app installed. Instead of logging in via the bank's site, the bank's app will be launched instead, from which the user will be able to log in (including via TouchID or Face ID) before being redirected back to your app. Support for App-to-App should be automatic once you have implemented support for OAuth on mobile with any of Plaid's mobile SDKs. Note that on iOS, this requires configuring an Apple App Association file to associate your redirect URI with your app, as described under [Create and register a redirect URI](https://plaid.com/docs/link/oauth/#create-and-register-a-redirect-uri). If using webviews, App-to-App support is not automatic; for an App-to-App experience, it is strongly recommended to use a Plaid mobile SDK instead.

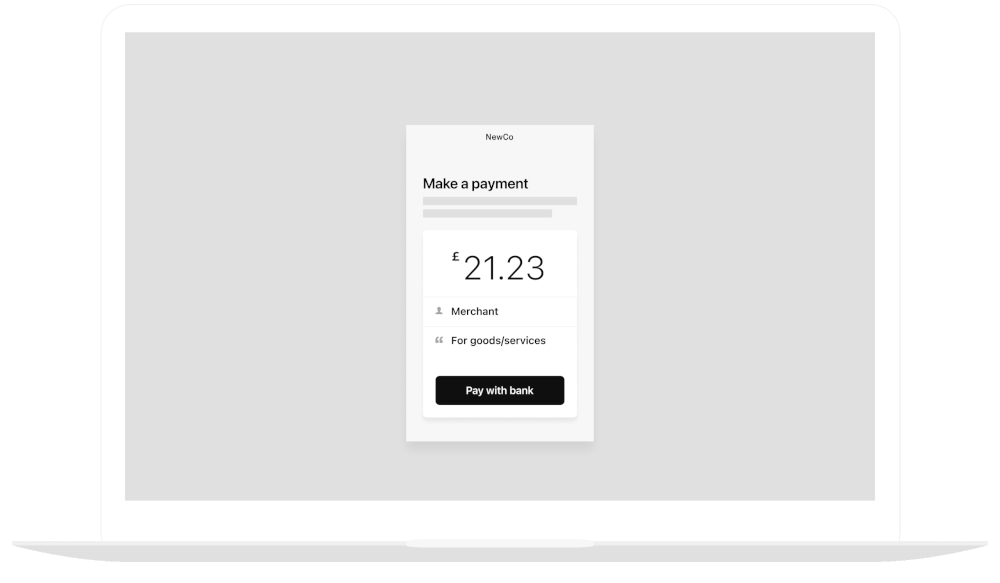
The full App-to-App flow cannot currently be tested in the Sandbox environment. In Sandbox, all OAuth flows will appear as a webview on top of your current application. It is recommended to test iOS App-to-App flows on physical devices, as some customers have reported issues with testing Universal Links flows on iOS simulators.

Chase is currently the only bank that supports App-to-App authentication on Plaid.

#### [QR code authentication](https://plaid.com/docs/link/oauth/" \l "qr-code-authentication)

For many European institutions, Plaid supports the ability for an end user to authenticate via their bank's mobile app – even if the user's journey begins in a desktop-based web session – in order to optimize for conversion. After the user selects an institution, they will be presented with the choice to scan a QR code and authenticate in the bank’s mobile app or to continue on desktop. When the user scans the QR code, they will be redirected to the bank’s app (or website, if the user does not have the app installed). After the user completes the OAuth flow, they will be redirected to a Plaid-owned page instructing them to return to their desktop to complete the flow.

To enable QR authentication, contact your Plaid account manager or [file a Support ticket](https://dashboard.plaid.com/support/new/product-and-development/product-troubleshooting/request-product-access). No changes to your Link OAuth implementation are required to enable this flow.



To test out the QR code flow in the Sandbox environment, you can use Flexible Platypus Open Banking (ins\_117181). When you launch Link with this institution selected in Sandbox, the QR code authentication flow will be triggered. The Sandbox institution does not direct you to a real bank's mobile app, but allows you to grant, deny, or simulate errors from the placeholder OAuth page instead.

##### [Supported institutions for QR code authentication](https://plaid.com/docs/link/oauth/" \l "supported-institutions-for-qr-code-authentication)

| **INSTITUTION NAME** | **INSTITUTION ID** |
| --- | --- |
| Bank of Scotland - Personal | ins\_118274 |
| Bank of Scotland - Business | ins\_118276 |
| Barclays (UK) - Mobile Banking: Business | ins\_118512 |
| Barclays (UK) - Mobile Banking: Personal | ins\_118511 |
| Barclays (UK) - Mobile Banking: Wealth Management | ins\_118513 |
| First Direct | ins\_81 |
| Halifax | ins\_117246 |
| HSBC (UK) - Business | ins\_118277 |
| HSBC (UK) - Personal | ins\_55 |
| Lloyds Bank - Business and Commercial | ins\_118275 |
| Lloyds Bank - Personal | ins\_61 |
| Monzo | ins\_117243 |
| Nationwide Building Society | ins\_60 |
| NatWest - Current Accounts | ins\_115643 |
| Revolut | ins\_63 |
| Royal Bank of Scotland - Current Accounts | ins\_115642 |
| Santander (UK) - Personal and Business | ins\_62 |
| Starling | ins\_117520 |
| Tesco (UK) | ins\_118393 |
| TSB | ins\_86 |
| Ulster Bank (UK) | ins\_117734 |

# Introduction to Link Remember Me

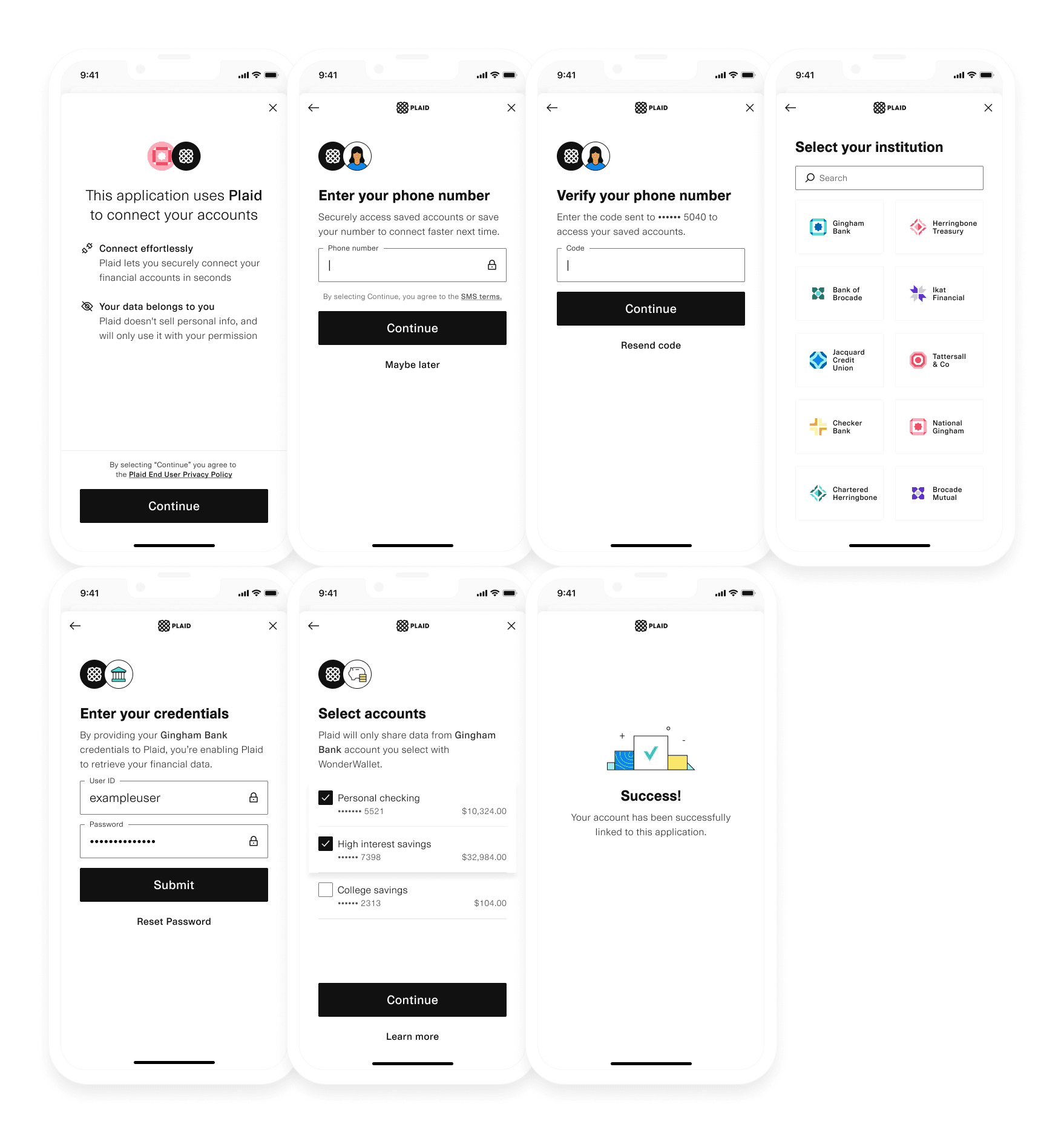
#### A faster and easier Link experience for existing Plaid users

Link Remember Me allows users to fast track account connections across thousands of Plaid-powered apps and services.

#### [Overview](https://plaid.com/docs/link/remember-me/" \l "overview)

Link Remember Me enables users to associate their phone number with the accounts they're connecting to a financial app or service. Once users have opted-in to being 'remembered' by Plaid, they'll be able to quickly connect those same accounts to other financial apps and services in the future using a one-time password.

After the Consent screen in Link, users can input their phone number and verify it using a one-time password sent to their device. Next, the user will proceed to select an institution and connect their account(s). Once account verification is completed, Plaid will associate the institution and accounts with the user's phone number.



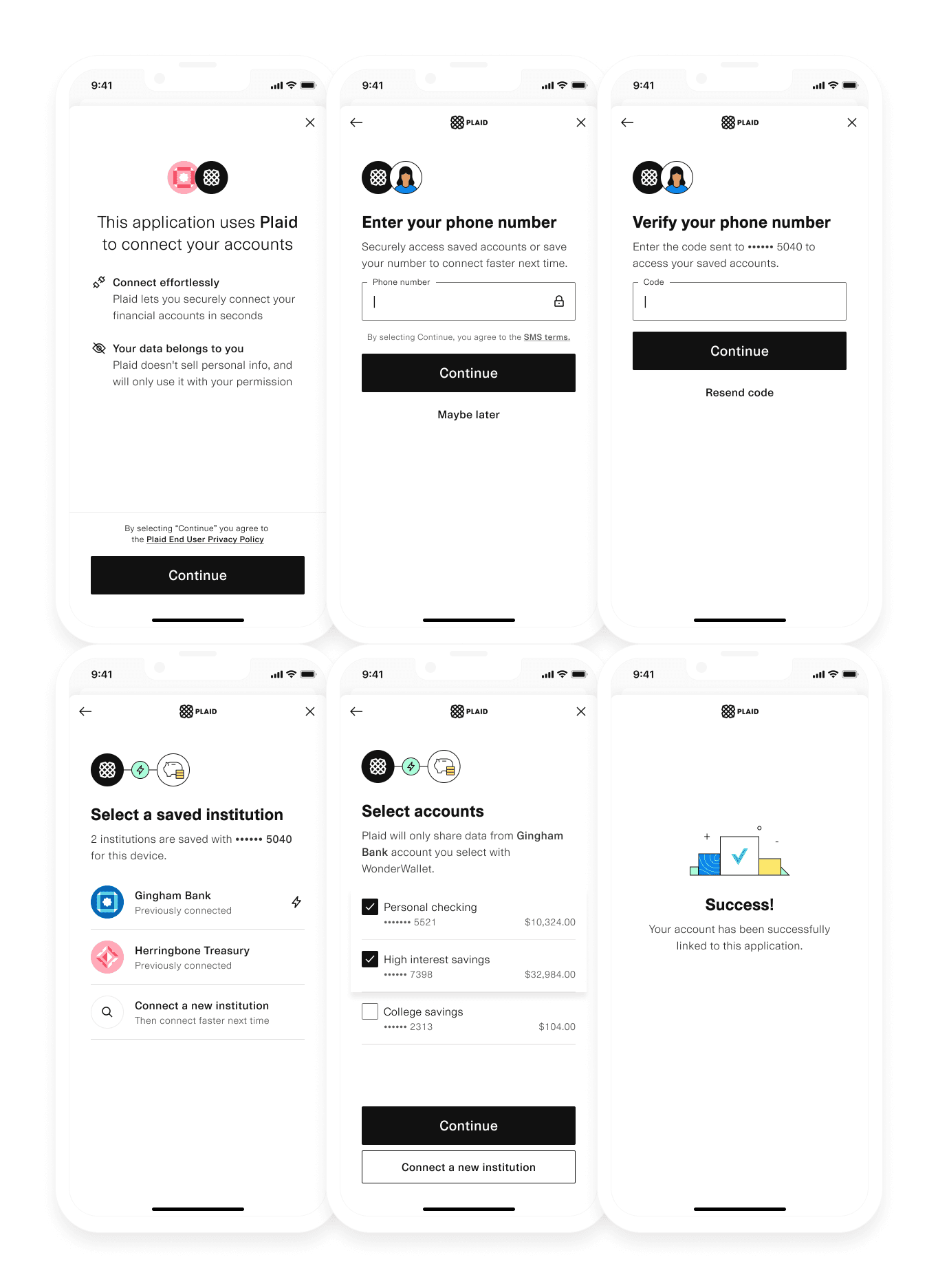
##### [Link Remember Me User Experience](https://plaid.com/docs/link/remember-me/" \l "link-remember-me-user-experience)

When users want to connect their saved institutions and accounts to additional Plaid-powered apps or services, Plaid runs security checks to detect that they are a returning user with the same phone number and device used when opting-in to Link Remember Me.

A returning user can access their saved institutions and accounts with Plaid the next time they open Link. The user can select from a list of previously connected institutions or connect a new institution after verifying their identity through OTP.

Some previously linked institutions do not require users to authenticate again using credentials for a frictionless experience, designated by the lightning bolt in the institution selection pane. This is based on a variety of factors such as whether the user's device is trusted and the institution supports OTP authentication. A previously connected institution without a lightning bolt will require authentication using credentials with the financial institution.

If there is only one previously linked institution that exists and the institution is eligible for the streamlined experience using OTP, the institution will be automatically selected —- enabling the user to go directly to [selecting which accounts](https://plaid.com/docs/link/customization/#account-select) to share with an app or service.



Link Remember Me is enabled by default for customers in the US and Canada. The majority of Link sessions will include Link Remember Me by the end of 2023.

#### [Testing Link Remember Me in Sandbox](https://plaid.com/docs/link/remember-me/" \l "testing-link-remember-me-in-sandbox)

Link Remember Me can only be tested in the Sandbox or Production environments, not Development. To enable a Sandbox Link session for Link Remember Me, use the [Link customization name](https://plaid.com/docs/api/tokens/#link-token-create-request-link-customization-name) REMEMBER\_ME\_SANDBOX when creating a Link token.

Real phone numbers do not work in the Sandbox. Instead, the Sandbox has been seeded with a test user whose phone numbers may be used to trigger different scenarios. To explore each scenario, enter the corresponding phone number and correct OTP. For all scenarios, the correct OTP is 123456.

Returning User: A user who has previously enrolled in Link Remember Me by confirming their identity through OTP and successfully linking an Item.

| **LINK REMEMBER ME SANDBOX SCENARIOS** | **SEEDED PHONE NUMBER** |
| --- | --- |
| New User | 415-555-0010 |
| Verified Returning User | 415-555-0011 |
| Verified Returning User: linked new account | 415-555-0012 |
| Verified Returning User: linked OAuth institution | 415-555-0013 |
| Verified Returning User + New Device | 415-555-0014 |
| Verified Returning User: automatic account selection | 415-555-0015 |

#### [Tracking Link Remember Me events](https://plaid.com/docs/link/remember-me/" \l "tracking-link-remember-me-events)

Link emits events to indicate whether or not users opt-in to being 'remembered' by Plaid to connect their accounts faster in the future. The SUBMIT\_PHONE event indicates that a user has provided their phone number to be remembered by Plaid. The VERIFY\_PHONE event indicates that a user has verified their phone number using OTP. The SKIP\_SUBMIT\_PHONE event indicates that a user has opted to not provide their phone number to be remembered by Plaid.

Additionally, new values have been added to the view\_name field in the TRANSITION\_VIEW event for Link Remember Me panes, as well as the match\_reason field in the SELECT\_INSTITUTION event.

The match\_reason field distinguishes the different institution selection options if a saved institution was selected: AUTO\_SELECT\_SAVED\_INSTITUTION indicates that the SELECT\_SAVED\_INSTITUTION pane was skipped. SAVED\_INSTITUTION indicates that the user selected a saved institution.

If the user chose to link a new institution instead, the CONNECT\_NEW\_INSTITUTION event is fired.

##### [Error events](https://plaid.com/docs/link/remember-me/" \l "error-events)

Link will emit the ERROR event when the user submits an invalid phone number or an invalid OTP. The error\_code will be INVALID\_PHONE\_NUMBER or INVALID\_OTP, respectively.

For more details, see [Link SDK documentation](https://plaid.com/docs/link/web/). For more information on tracking Link conversion in general, see [Improving Link conversion](https://plaid.com/docs/link/best-practices/#improving-link-conversion).

# Preventing duplicate Items

#### Prevent unnecessary billing and confusing application behavior

Prefer to learn by watching? A [video guide](https://youtu.be/TczdBk5z9eM) is available for this topic.

#### [How duplicate Items are created](https://plaid.com/docs/link/duplicate-items/" \l "how-duplicate-items-are-created)

An [Item](https://plaid.com/docs/quickstart/glossary/#item) represents a login at a financial institution. This login typically happens when an end user links an account (or several accounts) using [Plaid Link](https://plaid.com/docs/link/). A duplicate Item will be created if the end user logs into the same institution using the same credentials again using Plaid Link (in the same application), and if an access token is requested for the Item.

Duplicate Items can occur for multiple reasons. For example, a duplicate Item can occur if a user accidentally links the same account more than once, because they do not realize they already linked an account, or because their linked account is no longer working. Duplicate Items can also occur if a user intentionally links multiple Items for abusive purposes (for example, as part of an attempt to receive multiple sign-up bonuses or to evade a ban).

Duplicate Items can create confusing or unwanted behavior in your application and could result in being potentially billed for multiple Items. We recommend building safeguards in your application to help prevent end users from creating duplicate Items. In this article, we'll describe a few ways to prevent and detect Item duplication.

#### [Preventing duplicate Items](https://plaid.com/docs/link/duplicate-items/" \l "preventing-duplicate-items)

##### [Use the onSuccess callback metadata](https://plaid.com/docs/link/duplicate-items/" \l "use-the-onsuccess-callback-metadata)

The [**onSuccess**](https://plaid.com/docs/link/web/#onsuccess) callback is called when a user successfully links an Item using Plaid Link. This callback provides metadata that you can use to prevent duplicate Items from being created. One approach is to require a user login prior to launching Plaid Link so that you can retrieve existing Items associated with the user.

Then, before requesting an access\_token, examine and compare the onSuccess callback metadata to the user's existing Items. You can compare a combination of the accounts’ institution\_id, account name, and account mask to determine whether an end user has previously linked an account to your application. Do not exchange a public token for an access token if you detect a duplicate Item.

While the mask value is usually the same as the last 4 digits of the account number, this is not the case at all institutions. Never detect duplicate Items by attempting to match a mask with an account number.

Sample onSuccess metadata schema

**Copy**

1{

2 institution: {

3 name: 'Wells Fargo',

4 institution\_id: 'ins\_4'

5 },

6 accounts: [

7 {

8 id: 'ygPnJweommTWNr9doD6ZfGR6GGVQy7fyREmWy',

9 name: 'Plaid Checking',

10 mask: '0000',

11 type: 'depository',

12 subtype: 'checking',

13 verification\_status: ''

14 },

15 {

16 id: '9ebEyJAl33FRrZNLBG8ECxD9xxpwWnuRNZ1V4',

17 name: 'Plaid Saving',

18 mask: '1111',

19 type: 'depository',

20 subtype: 'savings'

21 }

22 ...

23 ],

24 link\_session\_id: '79e772be-547d-4c9c-8b76-4ac4ed4c441a'

25}

For Items linked via [OAuth](https://plaid.com/docs/link/oauth/#introduction-to-oauth) or [Account Selection v2](https://plaid.com/docs/link/account-select-v2-migration-guide/), note that duplicate Items are not always identical. For example, an Item may have a checking account linked, while its duplicate may have only a savings account linked. While this scenario is rare, if it becomes a problem for your use case, you can switch to using the institution\_id field on a per-user basis to prevent duplicate Items and enforce that each user of your app not link multiple Items with the same institution. However, because this is a legitimate use case for some applications, it is not recommended as the primary means of detecting duplicate Items.

**Duplicate Items with certain OAuth institutions**

For Chase and Charles Schwab, existing Items will be invalidated if an end user adds a second Item using the same credentials, at the point when the user has completed the institution's OAuth flow. For this reason, the onSuccess metadata approach cannot be used to prevent duplicate Items at these institutions. Instead, use alternative means of preventing duplicate Items, such as providing pre-Link messaging. If a duplicate Item is created, delete the old Item using [**/item/remove**](https://plaid.com/docs/api/items/#itemremove).

If the user completes the OAuth flow at either of these institutions but does not successfully complete the Link flow, the existing Item will still be invalidated and eventually enter the ITEM\_LOGIN\_REQUIRED state, even though a new Item was never successfully created. If this occurs, the old Item can repaired using [update mode](https://plaid.com/docs/link/update-mode/). If the user did successfully create a new Item, do not use [update mode](https://plaid.com/docs/link/update-mode/) on the old Item, as it will invalidate the new Item. Instead, delete the old Item using [**/item/remove**](https://plaid.com/docs/api/items/#itemremove).

##### [Implement Pre-Link messaging](https://plaid.com/docs/link/duplicate-items/" \l "implement-pre-link-messaging)

A lightweight but effective method for preventing accidental duplicate Items from being created is by providing relevant messaging and information to end users before they engage with Plaid Link in your application. For example, displaying a list of accounts they've already connected to your application can help prevent end users from inadvertently linking the same accounts again.

##### [Use Link's update mode](https://plaid.com/docs/link/duplicate-items/" \l "use-links-update-mode)

From time to time, an Item may become unhealthy due to entering the ITEM\_LOGIN\_REQUIRED state. Or, the Item may still be healthy, but you may want your end user to authorize additional accounts or permissions associated with the Item. When either of these scenarios happens, use [Link in update mode](https://plaid.com/docs/link/update-mode/) to refresh the Item instead of creating a new Item.

#### [Example implementation: Preventing accidental duplicate Items](https://plaid.com/docs/link/duplicate-items/" \l "example-implementation-preventing-accidental-duplicate-items)

For an example that demonstrates how to prevent accidental duplicate Items, see the [Plaid Pattern](https://github.com/plaid/pattern) sample app. Plaid Pattern implements simple server-side logic to check whether the user ID and institution ID pair already exist in the application database. If the pair exists, an access token will not be requested for this Item, thereby preventing a duplicate. In addition, a message is displayed to the end user informing them that they've already linked this Item. The relevant code can be found in [/server/routes/items.js](https://github.com/plaid/pattern/blob/master/server/routes/items.js#L41-L49) and [/server/db/queries/items.js](https://github.com/plaid/pattern/blob/master/server/db/queries/items.js#L73-L88).

#### [Identifying existing duplicate Items](https://plaid.com/docs/link/duplicate-items/" \l "identifying-existing-duplicate-items)

To identify existing duplicate Items, use the same matching logic as described above, but retrieve this data via the [**/accounts/get**](https://plaid.com/docs/api/accounts/#accountsget) endpoint instead of the onSuccess callback. Occasionally, the mask or name fields may be null, in which case you can compare institution\_id and client\_user\_id as a fallback. After identifying duplicate Items, use the [**/item/remove**](https://plaid.com/docs/api/items/#itemremove) endpoint to delete an Item.

If you are using Auth via [**/auth/get**](https://plaid.com/docs/api/products/auth/#authget), existing duplicate Items may be detected by using the account and routing number fields and looking for duplicates. Note that this method will not work for Chase Items, as Chase uses tokenized account numbers. If you wish to detect existing duplicate Items at Chase, you can use the [**persistent\_account\_id**](https://plaid.com/docs/api/accounts/#accounts-get-response-accounts-persistent-account-id) field, which will be the same across duplicate instances of a Chase Item. This field is in beta; to request access, contact Support or your Account Manager.

#### [Detecting and preventing duplicate Items across user accounts](https://plaid.com/docs/link/duplicate-items/" \l "detecting-and-preventing-duplicate-items-across-user-accounts)

When attempting to detect or prevent duplicate Items, your approach should depend on the type of duplicate Item you are trying to prevent. For example, you can often scope your search to either a single user's accounts, or to all accounts known to your application. If you are attempting to prevent accidental duplicate Items, you should scope the search to a single user; if you are attempting to detect or prevent abuse, it may make more sense to expand your search to all accounts that have been linked to your app.

Note that there can be legitimate use cases for the same financial institution account to be linked across multiple different user accounts (for example, family members who share a joint bank account, or employees of the same company who share access to a business account); depending on your use case, you may want to incorporate this data into a broader abuse detection framework rather than blocking all duplicate accounts.

# Link token migration guide

#### How to migrate your application from the public key to a Link token

## **[Introduction](https://plaid.com/docs/link/link-token-migration-guide/" \l "introduction)**

Plaid has introduced a new link\_token, which replaces the static public\_key. This is an improvement that provides better error validation, more advanced security, and enables Link event logs to be surfaced in the Plaid Dashboard. While Plaid does not plan to discontinue support for existing public\_key integrations, it is recommended that you upgrade your integration to use a link\_token, as it enables enhanced capabilities, and future Plaid development and features will be based on the link\_token infrastructure. Link tokens are also required for new OAuth integrations.

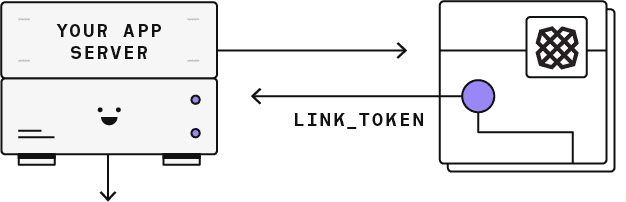
If you are unable to migrate to Link tokens at this time, see [maintaining a public key integration](https://plaid.com/docs/link/maintain-legacy-integration/) for instructions on working with legacy public keys.

This guide covers the client and server-side changes required to implement the new link\_token. Here's an overview of the updates before we dive into the details.

**The Plaid flow** begins when your user wants to connect their bank account to your app.



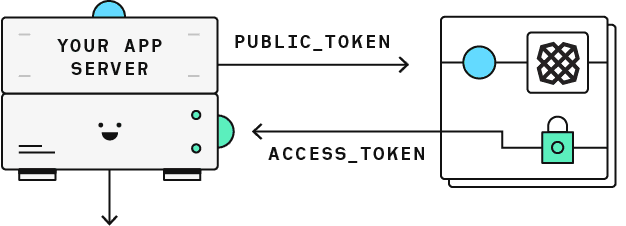
**1**Call [**/link/token/create**](https://plaid.com/docs/api/tokens/#linktokencreate) to create a link\_token and pass the temporary token to your app's client.



**2**Use the link\_token to open Link for your user. In the [**onSuccess** callback](https://plaid.com/docs/link/web/#onsuccess), Link will provide a temporary public\_token.



**3**Call [**/item/public\_token/exchange**](https://plaid.com/docs/api/tokens/#itempublic_tokenexchange) to exchange the public\_token for a permanent access\_token and item\_id for the new Item.



**4**Store the access\_token and use it to make product requests for your user's Item.



#### [What's new](https://plaid.com/docs/link/link-token-migration-guide/" \l "whats-new)

* Link will now be initialized with a new link\_token. The link\_token replaces all previous methods of initializing Link, including the public\_key (for initial Link), the public\_token (for update mode), and the payment\_token (for Payment Initiation).
* The new endpoint to create the link\_token is [**/link/token/create**](https://plaid.com/docs/api/tokens/#linktokencreate).
* The [**INVALID\_LINK\_TOKEN**](https://plaid.com/docs/errors/invalid-input/#invalid_link_token) error code is now available to gracefully handle invalidated tokens.
* Link events from sessions created with the new link\_token will be surfaced in the [Logs](https://dashboard.plaid.com/activity/logs) section of the Dashboard. However, Link events from sessions created with the public\_key will not.

#### [Link tokens](https://plaid.com/docs/link/link-token-migration-guide/" \l "link-tokens)

The link\_token is a new type of token that is created by your app's server and passed to your app's client to initialize Link. The Link configuration parameters that were previously set within Link itself are now set via parameters passed to [**/link/token/create**](https://plaid.com/docs/api/tokens/#linktokencreate) and conveyed to Link via the link\_token. If these configurations are still set client-side when using the link\_token, they will not have any effect on Link behavior.

## **[Update your integration](https://plaid.com/docs/link/link-token-migration-guide/" \l "update-your-integration)**

The overall process for updating your integration is:

1. Update your server to create a link\_token.
2. Update your client to pass the link\_token and handle INVALID\_LINK\_TOKEN errors.
3. Ensure you have updated all Link entry points, including those for update mode.
4. Test your integration.
5. Update any callsites that use the public\_key for authentication to use the client\_id and secret instead, then re-test those callsites.
6. Disable the public\_key.

Detailed instructions for each step can be found below.

#### [Update your server](https://plaid.com/docs/link/link-token-migration-guide/" \l "update-your-server)

Add a new authenticated endpoint to your app's server to create a link\_token by calling [**/link/token/create**](https://plaid.com/docs/api/tokens/#linktokencreate).

Select group for content switcher

Current librariesLegacy libraries

**Curl**

**Select Language**

**Copy**

1curl -X POST https://sandbox.plaid.com/link/token/create \

2-H 'Content-Type: application/json' \

3-d '{

4 "client\_id": "${PLAID\_CLIENT\_ID}",

5 "secret": "${PLAID\_SECRET}",

6 "client\_name": "Plaid Test App",

7 "user": { "client\_user\_id": "${UNIQUE\_USER\_ID}" },

8 "products": ["${PRODUCT}"],

9 "country\_codes": ["US"],

10 "language": "en",

11 "webhook": "https://webhook.example.com",

12 "redirect\_uri": "https://domainname.com/oauth-page.html",

13}'

Many of the parameters to [**/link/token/create**](https://plaid.com/docs/api/tokens/#linktokencreate) are the same as parameters previously set in Link. Aside from the change to snake\_case from camelCase, there are a few substantive differences, summarized below. Note that these bullets are only a summary; for the full [**/link/token/create**](https://plaid.com/docs/api/tokens/#linktokencreate) endpoint definition, see the [API Reference](https://plaid.com/docs/api/tokens/#linktokencreate).

* A new required parameter, user.id, has been added. This should be a unique identifier, such as the user ID of the end user in your application. It should not contain personally identifiable information, such as a phone number or email address.
* language and country\_codes, which were previously optional, are now required.
* accountSubtypes has been replaced by the account\_filters parameter, and its syntax has changed.

In addition, there are a few differences relevant specifically to European integrations:

* The oauthNonce parameter is no longer used, since it is effectively replaced by user.id.
* For the Payment Initiation product, The paymentToken is no longer used, and [**/payment\_initiation/payment/token/create**](https://plaid.com/docs/link/maintain-legacy-integration/#creating-a-payment-token) has been deprecated. Instead, the payment\_id should be provided to [**/link/token/create**](https://plaid.com/docs/api/tokens/#linktokencreate) via the payment\_initiation.payment\_id parameter. Initializing Link with the returned link\_token will launch the Payment Initiation flow.

##### [Authenticate your app](https://plaid.com/docs/link/link-token-migration-guide/" \l "authenticate-your-app)

The endpoint used to create a link\_token should only be available to users that are logged in to your app. Once your user is logged in, pass an identifier that uniquely identifies your user into the user.client\_user\_id field. The value of this field should not be personally identifiable information such as an email address or phone number. Using user.client\_user\_id will allow for easier debugging in the [Dashboard logs](https://dashboard.plaid.com/activity/logs). You will be able to search for Link logs that belong to one of your end users.

As this update involves an additional API call when adding an Item, create a link\_token when your user initially visits your app to avoid adding latency to your Link flow.

#### [Update your client](https://plaid.com/docs/link/link-token-migration-guide/" \l "update-your-client)

For each of your web and mobile apps, use the new endpoint you created to fetch a link\_token, then pass it into one of Plaid's Link SDKs to initialize Link. You can then safely remove the public\_key and other client-side configs that are now configured in the [**/link/token/create**](https://plaid.com/docs/api/tokens/#linktokencreate) request.

If the token expires or the user enters too many invalid credentials, the link\_token can become invalidated. If it does get into an invalid state, Link will exit with an [**INVALID\_LINK\_TOKEN**](https://plaid.com/docs/errors/invalid-input/#invalid_link_token) error code. By recognizing when this error occurs in the onExit callback, you can generate a fresh link\_token for the next time your user opens Link.

##### [Update Link web](https://plaid.com/docs/link/link-token-migration-guide/" \l "update-link-web)

The code below demonstrates code that passes the new link\_token to Link. For more in-depth coverage on how to integrate with Link web, see the [Link web docs](https://plaid.com/docs/link/web/).

Note that in the error handling section, in order to handle an invalid link\_token for Link in the browser, you will need to gracefully clean up the old iframe before reinitializing Link. To do this, use the destroy() method and reinitialize Link with a new link\_token in the onExit callback.

Initialize Link with a Link Token (Web)

**Copy**

1<button id="link-button">Link Account</button>

2<script src="https://cdn.plaid.com/link/v2/stable/link-initialize.js"></script>

3<script type="text/javascript">

4(async function() {

5 const fetchLinkToken = async () => {

6 const response = await fetch('/create\_link\_token', { method: 'POST' });

7 const responseJSON = await response.json();

8 return responseJSON.link\_token;

9 };

10

11 const configs = {

12 // 1. Pass a new link\_token to Link.

13 token: await fetchLinkToken(),

14 onSuccess: async function(public\_token, metadata) {

15 // 2a. Send the public\_token to your app server.

16 // The onSuccess function is called when the user has successfully

17 // authenticated and selected an account to use.

18 await fetch('/exchange\_public\_token', {

19 method: 'POST',

20 body: JSON.stringify({ public\_token: public\_token }),

21 });

22 },

23 onExit: async function(err, metadata) {

24 // 2b. Gracefully handle the invalid link token error. A link token

25 // can become invalidated if it expires, has already been used

26 // for a link session, or is associated with too many invalid logins.

27 if (err != null && err.error\_code === 'INVALID\_LINK\_TOKEN') {

28 linkHandler.destroy();

29 linkHandler = Plaid.create({

30 ...configs,

31 token: await fetchLinkToken(),

32 });

33 }

34 if (err != null) {

35 // Handle any other types of errors.

36 }

37 // metadata contains information about the institution that the

38 // user selected and the most recent API request IDs.

39 // Storing this information can be helpful for support.

40 },

41 };

42

43 var linkHandler = Plaid.create(configs);

44

45 document.getElementById('link-button').onclick = function() {

46 linkHandler.open();

47 };

48})();

49</script>

##### [Update Link iOS](https://plaid.com/docs/link/link-token-migration-guide/" \l "update-link-ios)

The iOS SDK now provides an initWithLinkToken method on both the PLKConfiguration and the PLKPlaidLinkViewController classes that should allow you to easily initialize Link with a link\_token.

The code below shows how to initialize Link with the link\_token in iOS. For more in-depth coverage on how to integrate with Link iOS, see the [iOS docs](https://plaid.com/docs/link/ios/).

Initialize Link with a Link Token (iOS)

**Swift**

**Select Language**

**Copy**

1let linkConfiguration = PLKConfiguration(linkToken: "GENERATED\_LINK\_TOKEN")

2let linkViewDelegate = self

3let linkViewController = PLKPlaidLinkViewController(

4 linkToken: "GENERATED\_LINK\_TOKEN",

5 configuration: linkConfiguration,

6 delegate: linkViewDelegate,

7)

8if (UI\_USER\_INTERFACE\_IDIOM() == .pad) {

9 linkViewController.modalPresentationStyle = .formSheet;

10}

11present(linkViewController, animated: true)

##### [Update Link Android](https://plaid.com/docs/link/link-token-migration-guide/" \l "update-link-android)

The Android SDK exposes a new class called LinkTokenConfiguration. This class accepts the link\_token and should be passed into the openPlaidLink method.

The code below demonstrates how to use the LinkTokenConfiguration class to open Link. For more in depth coverage on the Android SDK, see the [Android docs](https://plaid.com/docs/link/android/).

Initialize Link with a Link Token (Android)

**Kotlin**

**Select Language**

**Copy**

1import android.os.Bundle

2import android.util.Log

3import androidx.appcompat.app.AppCompatActivity

4

5import com.plaid.link.Plaid

6import com.plaid.link.linkTokenConfiguration

7import com.plaid.link.openPlaidLink

8import com.plaid.link.configuration.AccountSubtype

9import com.plaid.link.configuration.LinkLogLevel

10import com.plaid.link.configuration.PlaidEnvironment

11import com.plaid.link.configuration.PlaidProduct

12import com.plaid.link.event.LinkEvent

13import java.util.Locale

14

15class MainActivity : AppCompatActivity() {

16

17 override fun onCreate(savedInstanceState: Bundle?) {

18 super.onCreate(savedInstanceState)

19

20 // Optional

21 Plaid.setLinkEventListener { event -> Log.i("Event", event.toString()) }

22

23 // Open Link – put this inside of a Button / Fab click listener

24 this@MAINACTIVITY.openPlaidLink(

25 linkTokenConfiguration {

26 // required

27 token = "GENERATED\_LINK\_TOKEN"

28

29 // optional

30 logLevel = LinkLogLevel.WARN // Defaults to ASSERT

31 extraParams = mapOf() // Map of additional configs

32 }

33 );

34 }

35}

#### [Update Link update mode flows](https://plaid.com/docs/link/link-token-migration-guide/" \l "update-link-update-mode-flows)

With the introduction of the link\_token, [**/item/public\_token/create**](https://plaid.com/docs/api/tokens/#itempublic_tokencreate) is deprecated, and Link's update mode is initialized by passing in a link\_token rather than a public\_token. You can obtain this link\_token by calling [**/link/token/create**](https://plaid.com/docs/api/tokens/#linktokencreate) and providing the user.id of the user whose Item is being updated, along with the access\_token for the Item. Make sure to update any update mode flow entry points in addition to updating primary Link flows. For more details and complete sample code, see [Updating Items via Link](https://plaid.com/docs/link/update-mode/).

Initializing Link with a link\_token for update mode

**Copy**

1// Initialize Link with the token parameter

2// set to the generated link\_token for the Item

3const linkHandler = Plaid.create({

4 token: 'GENERATED\_LINK\_TOKEN',

5 onSuccess: (public\_token, metadata) => {

6 // You do not need to repeat the /item/public\_token/exchange

7 // process when a user uses Link in update mode.

8 // The Item's access\_token has not changed.

9 },

10 // ...

11});

#### [Test in Sandbox](https://plaid.com/docs/link/link-token-migration-guide/" \l "test-in-sandbox)

Once you have updated both your app's client and server, it's time to test that your integration works. The best way to test is by using the test credentials in the Sandbox:

**Copy**

1username: user\_good

2password: pass\_good

Test your error handling flow for INVALID\_LINK\_TOKEN by using the Sandbox test credentials to force an error:

**Copy**

1username: user\_custom

2password: { "force\_error": "INVALID\_LINK\_TOKEN" }

You can also verify that you have updated correctly by viewing Link event logs in the [Plaid Dashboard](https://dashboard.plaid.com/activity/logs).

To test your update mode implementation, use the [**/sandbox/item/reset\_login**](https://plaid.com/docs/api/sandbox/#sandboxitemreset_login) endpoint to force an Item into a state that requires an update, then walk through the test steps above.

#### [Update API endpoints](https://plaid.com/docs/link/link-token-migration-guide/" \l "update-api-endpoints)

In order to completely migrate off of the public\_key, there are a few Plaid API endpoints that should replace the public\_key with the client\_id and secret: [**/institutions/search**](https://plaid.com/docs/api/institutions/#institutionssearch), [**/institutions/get\_by\_id**](https://plaid.com/docs/api/institutions/#institutionsget_by_id), and [**/sandbox/public\_token/create**](https://plaid.com/docs/api/sandbox/#sandboxpublic_tokencreate).

Because the client\_id and secret are now used to authenticate the endpoints above, they should only be called from your server. The ability to call them from the client has been removed in the latest client library updates.

#### [Disable the public key](https://plaid.com/docs/link/link-token-migration-guide/" \l "disable-the-public-key)

After completing all of the above steps, you can now confidently disable the public\_key via the [Plaid Dashboard](https://dashboard.plaid.com/developers/keys). This can be done on a per-environment basis, with different settings for Sandbox, Development, and Production, to help you test your migration. The public\_key can also be disabled separately for Link and for the API. Disabling it for Link means you will be required to use the link\_token to initialize Link. Disabling it for the API means that you will be required to use the client\_id and secret to call [**/institutions/search**](https://plaid.com/docs/api/institutions/#institutionssearch), [**/institutions/get\_by\_id**](https://plaid.com/docs/api/institutions/#institutionsget_by_id), and[**/sandbox/public\_token/create**](https://plaid.com/docs/api/sandbox/#sandboxpublic_tokencreate).

A screenshot of a application

Description automatically generated

It is recommended you disable the public\_key as soon as possible to ensure that you do not accidentally add legacy public\_key-dependent code to your application.

#### [Conclusion](https://plaid.com/docs/link/link-token-migration-guide/" \l "conclusion)

Congratulations on upgrading to the new link\_token! If you require any help migrating to the link\_token integration, please [contact Plaid Support](https://dashboard.plaid.com/support/new/product-and-development/developer-lifecycle/link) for assistance.

QUICK START EXAMPLE: [plaid/tiny-quickstart: A minimal quickstart demonstrating Plaid Link, Balances, and OAuth (github.com)](https://github.com/plaid/tiny-quickstart)