



## Paxos APIs : Implementation Guide : Fiat-Gateway

Paxos Trust Company is now offering, via its public-facing API offering, a partner solution that enables end-customers to easily swap between dollars and stablecoins issued by Paxos, including both Paxos Standard (PAX) and Binance USD (BUSD). For example, a crypto-native platform could use Paxos' Fiat Gateway to accept dollar deposits and convert them into PAX for use on the platform; in essence enabling their customers to fund their accounts with USD.

Until now, institutions and individual customers were only able to create PAX via our web application, or by purchasing it on an exchange using other crypto assets. We want to make accessing PAX easier and faster through our suite of public API's. These API endpoints will enable any approved partner to integrate the frictionless functionality of PAX directly onto their platform, via a seamless customer experience consistent with the brand. This means that every crypto service provider who doesn't have access to traditional fiat gateways can still offer their users a way to fund their accounts using fiat currency.

The fiat-gateway solution is just the beginning of Paxos' public-facing API mission; to provide our current and future Paxos product offerings and functionality as an infrastructure layer that can be used by other institutions to help bridge the gap between the traditional financial infrastructure and digital assets, and thus, enable our partners to power their businesses in a secure and trustworthy way through the use of our APIs.

### Getting Started

1. [Create](#) a Paxos account or [Sign In](#) using your existing Paxos credentials.
2. Choose an Integration Type:
  - a. **First-Party Integration:** Individuals or Institutions utilizing the Paxos APIs to act on their own behalf and perform actions in their own Paxos account.
  - b. **Third-Party Integration:** Institutions who are utilizing the Paxos APIs to offer Paxos functionality to the end-customers of their platform. End-customers must first authenticate by linking their Paxos account and granting consent to the Institution (Platform) to act on their behalf and perform actions in that account.
3. Create the corresponding **Paxos API Credentials** via the Partner Onboarding forms:
  - a. First-Party Integration: <https://account.paxos.com/first-party-onboard>

- b. Third-Party Integration: <https://account.paxos.com/third-party-onboard>
    - i. Only Institutional Paxos accounts may implement the Third-Party Integration. All Institutions must also be approved by a Paxos Admin for this type of integration. Please email [API-Support@paxos.com](mailto:API-Support@paxos.com) in order to request access for your platform.
4. Proceed to the Authentication section below and begin implementation.

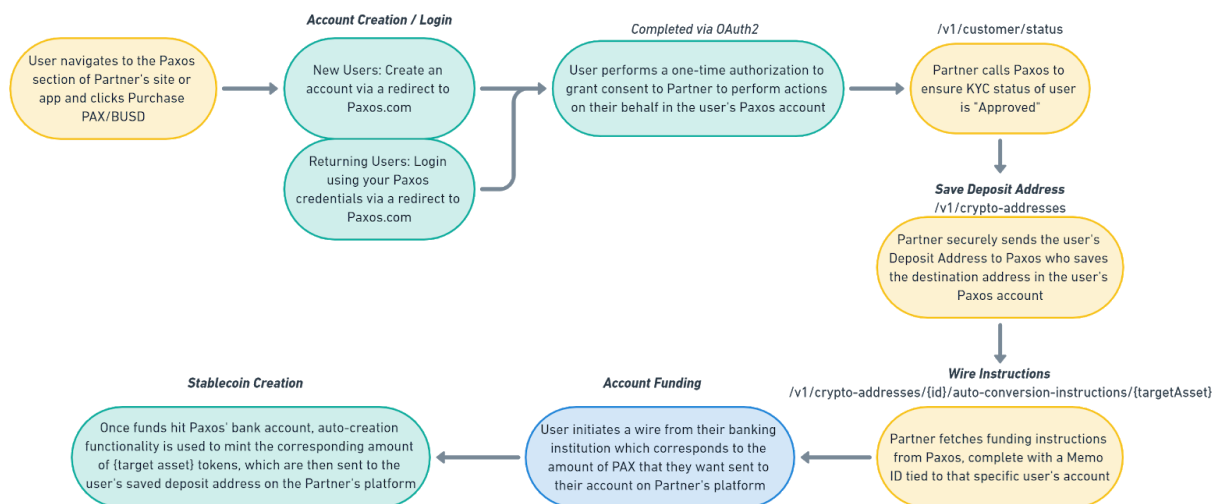
## Authentication

Please reference the [Authentication Guide](#) for technical specifications of the OAuth2 flow.

## User Flows

Please see below for flow diagrams outlining the customer journey for both Deposits (Stablecoin Creations) and Withdrawals (Stablecoin Redemptions).

## Fiat-Deposit Flow



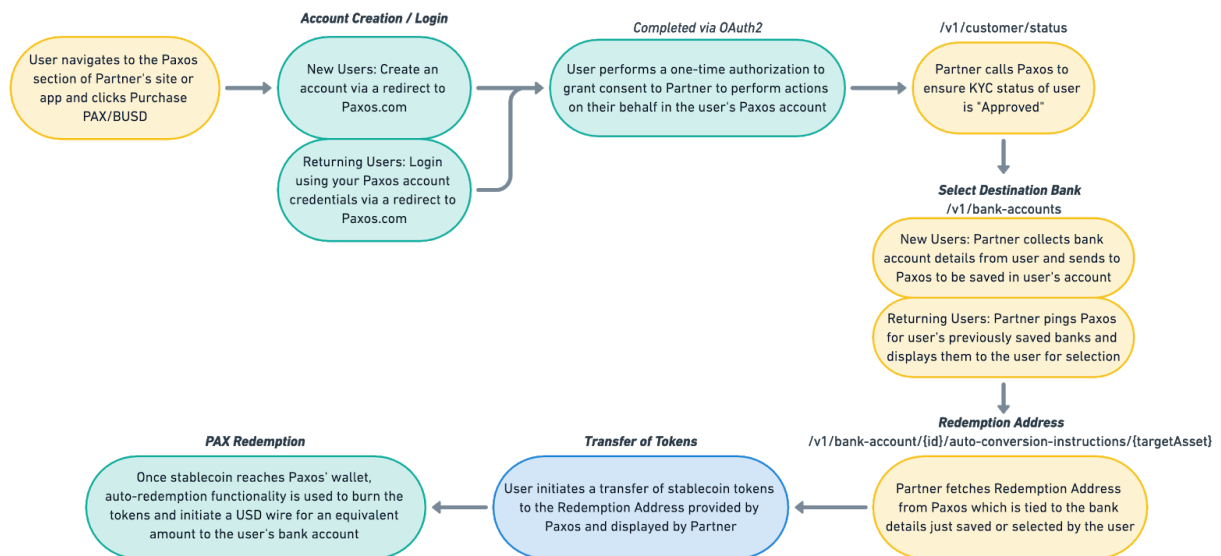
The diagram above outlines the steps taken by the partner, and the end-user, in order for a customer to fund their account on the partner's platform using fiat-currency.

1. End-user navigate to the experience rendered by partner on their site or app to facilitate this functionality. User clicks something to the effect of:
  - a. "Purchase PAX/BUSD"
  - b. "Fund your account with USD"
  - c. "Deposit Fiat-Currency"
2. End-user creates a Paxos account or uses their existing Paxos credentials in order to link their Paxos account to the partner's platform.
3. Consent/Authorization
  - a. End-user performs a one-time authorization to grant consent to the Partner and enable them to perform actions on their behalf in the user's Paxos account.
4. KYC Status
  - a. Partner calls the endpoint [GetCustStatus](#) to ensure KYC status of user is Approved
    - i. If the KYC status of the user is not approved, Partner will not be able to call any other endpoints on their behalf.
5. Saving Deposit Address
  - a. Partner securely sends the user's PAX Deposit Address to Paxos who saves the address in the user's Paxos account. Refer to these endpoints:
    - i. [CreateCryptoAddress](#) - Saves the asset-specific deposit address associated with the customer on the partner's platform. This is where tokens will be sent after funding occurs. Depending on the use-case, this can also be used to save additional deposit addresses for that user.
      1. *POST /v1/crypto-addresses*
    - ii. [GetCryptoAddress](#) - Returns a saved destination address of the customer, specified by id.
      1. *GET /v1/crypto-addresses/{id}*
    - iii. [ListCryptoAddresses](#) - Returns a list of the customer's saved destination addresses, where tokens can be sent.
      1. *GET /v1/crypto-addresses*
6. Wire Instructions / Account Funding
  - a. Partner fetches fiat wire instructions from Paxos, complete with a Memo ID tied to that specific user's account.
    - i. [GetCryptoAddressAutoConversionInstructions](#) - Returns funding instruction options for the auto-creation of Paxos-issued USD-backed tokens (target asset = PAX or BUSD). For fiat-funding, a Memo ID will be provided. Funds sent to Paxos using these instructions will be automatically converted to the token requested by the customer and sent to the destination address which corresponds to the specific Memo-ID or ETH address.
      1. *GET /v1/crypto-addresses/{id}/auto-conversion-instructions/{targetAsset}*

## 7. PAX Creation

- a. Once funds hit Paxos' bank account, auto-creation functionality is used to mint the corresponding amount of PAX tokens, which are then sent to the user's specified deposit address on the partner's platform.
  - i. [GetActivity](#) - Return the transaction details for a specific customer activity (deposit/withdrawal)
    1. `GET /v1/activities/{id}`
  - ii. [GetActivities](#) - Read the transaction history associated with the end-customer's Paxos account. Returns information on deposits and withdrawals of crypto-assets and fiat currency. Apply filters as necessary.
    1. `GET /v1/activities`

## Fiat-Withdrawal Flow



1. End-user navigate to the experience rendered by partner on their site or app to facilitate this functionality. User clicks something to the effect of:
  - a. "Redeem PAX/BUSD"
  - b. "Withdraw Account Balance"
  - c. "Withdraw Fiat-Currency"
2. End-user creates a Paxos account or uses their existing Paxos credentials in order to link their Paxos account to the partner's platform.
3. Consent/Authorization

- a. End-user performs a one-time authorization to grant consent to the Partner and enable them to perform actions on their behalf in the user's Paxos account.
4. KYC Status
  - a. Partner calls the endpoint [GetCustStatus](#) to ensure KYC status of user is Approved
    - i. If the KYC status of the user is not approved, Partner will not be able to call any other endpoints on their behalf.
5. Select Destination Bank
  - a. New Users: Partner collects bank account details from the user and sends it to Paxos to be saved in the user's account.
    - i. [CreateBankAccount](#) - Creates a new saved bank account for the customer, where the proceeds of a token redemption can be sent.
      1. *POST /v1/bank-accounts*
  - b. Returning Users: Partner pings Paxos for the user's previously saved banks and displayed them to the user for selection.
    - i. [ListBankAccounts](#) - Returns a comprehensive list of bank accounts that have been saved by the customer. Used to enable the customer to designate which bank account they want their funds sent to after token redemption.
      1. *GET /v1/bank-accounts*
6. Redemption Address
  - a. Partner fetches Redemption Address from Paxos which is tied to the bank details just saved or selected by the user.
    - i. [BankAccountAutoConversionInstructions](#) - Returns a specific Paxos-issued deposit address. This address is associated with the bank account where the customer wants their funds sent. Tokens sent to this address will be automatically converted to USD and sent to the corresponding bank account of the customer. Each bank account saved by the user will be assigned its own unique redemption address.
      1. *GET /v1/bank-account/{id}/auto-conversion-instructions/{targetAsset}*
7. Transfer of Tokens
  - a. The user initiates a transfer of PAX or BUSD tokens to the Redemptions Address provided by Paxos and displayed by Partner
8. Token Redemption
  - a. Once PAX or BUSD tokens arrive at Paxos, auto-redemption functionality is used to burn the tokens and initiate an outgoing USD wire for an equivalent amount to the user's bank account.