Receive an Inbound Transfer

Initiate a currency transfer into a developer-controlled wallet you've already created.Suggest Edits

Circle Programmable Wallets provide a comprehensive developer solution to storing, sending, and spending Web3 digital currencies and NFTs. You or your users can manage asset infrastructure. Circle provides a one-stop-shop experience with all the tools and services to handle the complex parts, including security, transaction monitoring, account recovery flows, and more.

This guide outlines initiating a currency transfer into a previously created developer-controlled wallet. You'll learn to make API requests via Circle's API references or cURL requests. In this guide, you'll find cURL requests presented inline, while API references are linked from the API endpoint text. You can find instructions on using it in the testing via the reference pages guide.

- As with most of our quickstarts, all API calls and transactions in this guide occur within the Testnet environment; no real-world funds will be transferred.
- If you have not yet created a developer-controlled wallet, see this guide
- .

1. Acquire the Wallet Address

To receive funds on-chain, you will need the wallet's address. The address is the on-chain identifier provided to the sending wallet or faucet in the following step. To retrieve the address, make a request to GET /wallets as shown below. You should also note down the wallets.id this will be used in step 3.

Node.js cURL // Import and configure the developer-controlled wallet SDK const { initiateDeveloperControlledWalletsClient } = require('@circle-fin/developer-controlled-wallets'); const circleDeveloperSdk = initiateDeveloperControlledWalletsClient({ apiKey: '', entitySecret: '' });

const response = await circleDeveloperSdk.getWallet({}); curl --request GET \ --url 'https://api.circle.com/v1/w3s/wallets' \ --header 'accept: application/json' \ --header 'content-type: application/json' \ --header 'authorization: Bearer ' Response Body { "data": { "wallets": [{ "id": "ce714f5b-0d8e-4062-9454-61aa1154869b", "state": "LIVE", "walletSetId": "0189bc61-7fe4-70f3-8a1b-0d14426397cb", "custodyType": "DEVELOPER", "address": "0xf5c83e5fede8456929d0f90e8c541dcac3d63835", "blockchain": "MATIC-MUMBAI", "accountType": "SCA", "updateDate": "2023-08-03T19:33:14Z", "createDate": "2023-08-03T19:33:14Z" }] } }

2. Transfer Testnet Currency

Send Testnet currency from an external wallet outside the Programmable Wallet infrastructure into your applicable wallet address . The best way to achieve this is through the use of a faucet . In our case, we will use the USDC Faucet to transfer USDC on Polygon Mumbai to the wallet.

Here is a list of reputable faucets for each blockchain:

1. Polygon:
2. USDC on MATIC Faucet
3. |Mumbai MATIC Faucet
4. Ethereum:
5. USDC on ETH Faucet
6. |Sepolia ETH Faucet
7. Avalanche:
8. USDC on AVAX Faucet
9. |Fuji AVAX Faucet

3. Check the Transfer State

Once an inbound transfer is made to the address and completed, Circle sends a notification to a subscribed endpoint . The Webhook notification will be similar to the one below.

Webhook Request Body { "subscriptionId": "d4c07d5f-f05f-4fe4-853d-4dd434806dfb", "notificationId": "05b3f4e5-ec27-44b8-aa40-3698577f6d92", "notificationType": "transactions.inbound", "notification": { "id": "2f4b6bcd-a752-5d8b-996b-92e3e04bd33b", "blockchain": "MATIC-MUMBAI", "walletId": "ce714f5b-0d8e-4062-9454-61aa1154869b", "tokenId": "38f2ad29-a77b-5a44-be05-8d03923878a2", "destinationAddress": "0x075e62c80e55d024cfd8fd4e3d1184834461db57", "amounts": ["10"], "nftTokenIds": [], "state": "COMPLETED", "transactionType": "INBOUND", "createDate": "2023-07-28T16:03:08Z", "updateDate": "2023-07-28T16:06:40Z" }, "timestamp": "2023-07-28T16:06:40.907831464Z", "version": 2 } Alternatively, you can poll GET /transactions using the walletId as query parameter.

Node.js cURL const response = await circleDeveloperSdk.listTransactions({ walletIds: ['ce714f5b-0d8e-4062-9454-

61aa1154869b'] }); curl --request GET \ --url 'https://api.circle.com/v1/w3s/transactions?walletIds=ce714f5b-0d8e-4062-9454-61aa1154869b' \ --header 'accept: application/json' \ --header 'content-type: application/json' \ --header 'authorization: Bearer ' Response Body { "data": { "transactions": [{ "id": "97d22a88-6d25-5947-a7b6-61b3dc668057", "blockchain": "MATIC-MUMBAI", "tokenId": "38f2ad29-a77b-5a44-be05-8d03923878a2", "walletId": "ce714f5b-0d8e-4062-9454-61aa1154869b", "sourceAddress": "0x6e5eaf34c73d1cd0be4e24f923b97cf38e10d1f3", "destinationAddress": "0xf5c83e5fede8456929d0f90e8c541dcac3d63835", "transactionType": "INBOUND", "custodyType": "DEVELOPER", "state": "CONFIRMED", "amounts": ["10"], "nfts": null, "txHash": "0xdd2f81a78605dcad759265c703fb2b4c507c5ea100319338422714bfcde77225", "blockHash": "0x4df6092fdb868331614771ff11944b43051cf6ed1067f8cfa55e9d40ef61426b", "blockHeight": 9423950, "networkFee": "", "firstConfirmDate": "2023-07-28T19:07:24Z", "operation": "TRANSFER", "abiParameters": null, "createDate": "2023-07-28T19:07:34Z", "updateDate": "2023-07-28T19:07:37Z" }] } } You have successfully transferred 10 USDC into your developer-controlled wallet! Now you can move on to transferring tokens from wallet to wallet . Updated2 months ago *Table of Contents * *1. Acquire the Wallet Address * *2. Transfer Testnet Currency * *3. Check the Transfer State