

Receive an Inbound Transfer

Initiate a currency transfer into a user-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 user-controlled wallet. You'll learn to use Circle's sample application and how 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 user-controlled wallet, see [this guide](#)
- .

1. Acquire a Session Token

Make a request to [POST /users/token](#) using a previously created [userId](#). The `userToken` is a 60-minute session token to initiate requests requiring a user challenge (PIN code entry). After 60 minutes, the session expires, and a new `userToken` must be generated via the same endpoint.

```
Node.js cURL // Import and configure the user-controlled wallet SDK
const { initiateUserControlledWalletsClient } = require('@circle-fin/user-controlled-wallets');
const circleUserSdk = initiateUserControlledWalletsClient({ apiKey: "" });
```

```
const response = await circleUserSdk.createUserToken({
  userId: '2f1dcb5e-312a-4b15-8240-abefc0e3463'
});
curl --request POST \
--url 'https://api.circle.com/v1/w3s/users/token' \
--header 'accept: application/json' \
--header 'content-type: application/json' \
--header 'authorization: Bearer ' \
--data '{ "userId": "2f1dcb5e-312a-4b15-8240-abefc0e3463" }'
Response Body { "data": { "userToken": "eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVC9.eyJkZXZlbnR5c29ubWVudCI6IiRFRU1QilCjJlbnRpdHJlZC6lRmRldhOGM5LTl0OTAiNDVINC1hNjc0LWQyMGFkNjg4MWI3YlsmV4cCjUR8i4zMmfdURw3FFcQldSbm-BUg6M7FP_ip-cs9xBbNmRZA31gMd1aKdcajJ9SvIvUowYfGXM3VcNF8rTFfW-gk1-KzU4u10U35XXbbMcW1moxE0Rqx_fKotDgk2VdITuuds5d5TIQzAXECqeCOctNoDKktMkglltbnLxOaRI2ReZjGt-ctD2V0DbYNO4T_ndPSUDl6qD7dXQRRed5uDcezJYoha3Qj3tFGBglEnox2Y6DWTbllqjwmfTGrU8Pr0yz4jQz7suGwmiCzHPxcpYxMzYQ", "encryptionKey": "Tlcyxz7T5ztRLQq5+pic0MIETblYimOo2d7idV/UFM=" } }
```

2. Acquire the Wallet ID

Make a request to [GET /wallets](#) using the `userToken` returned in Step 1 to retrieve the wallet information for a given user.

```
Node.js cURL
const response = await circleUserSdk.listWallets({ userToken: "" });
curl --request POST \
--url 'https://api.circle.com/v1/w3s/wallets' \
--header 'accept: application/json' \
--header 'content-type: application/json' \
--header 'authorization: Bearer ' \
--header 'X-User-Token: '
Response Body { "data": { "wallets": [ { "id": "01899cf2-d415-7052-a207-f9862157e546", "state": "LIVE", "walletSetId": "01899cf2-d407-7f89-b4d9-84d63573f138", "custodyType": "ENDUSER", "userId": "2f1dcb5e-312a-4b15-8240-abefc0e3463", "address": "0x075e62c80e55d024cfd8fd4e3d1184834461db57", "addressIndex": 0, "blockchain": "MATIC-MUMBAI", "accountType": "SCA", "updateDate": "2023-07-28T14:41:47Z", "createDate": "2023-07-28T14:41:47Z" } ] }
```

3. Transfer Testnet Currency

Transfer 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 user wallet.

Here is a list of reputable faucets for each blockchain:

- Polygon:
 - [USDC on MATIC Faucet](#)
 - [Mumbai MATIC Faucet](#)
- Ethereum:
 - [USDC on ETH Faucet](#)
 - [Sepolia ETH Faucet](#)
- Avalanche:
 - [USDC on AVAX Faucet](#)
 - [Fuji AVAX Faucet](#)

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": "01899cf2-d415-7052-a207-f9862157e546", "tokenId": "38f2ad29-a77b-5a44-be05-8d03923878a2", "userId": "2f1dcb5e-312a-4b15-8240-abefc0e3463", "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 userId or userToken associated with your user.
```

```
Node.js cURL
const response = await circleUserSdk.listTransactions({ userToken: "" });
curl --request GET \
--url 'https://api.circle.com/v1/w3s/transactions' \
--header 'accept: application/json' \
--header 'content-type: application/json' \
--header 'authorization: Bearer ' \
--header 'X-User-Token: '
Response Body { "data": { "transactions": [ { "id": "97d22a88-6d25-5947-a7b6-61b3dc668057", "blockchain": "MATIC-MUMBAI", "tokenId": "38f2ad29-a77b-5a44-be05-8d03923878a2", "walletId": "01899cf2-d415-7052-a207-f9862157e546", "sourceAddress": "0x6e5eaf34c73d1cd0be4e24f923b97cf38e10d1f3", "destinationAddress": "0x075e62c80e55d024cfd8fd4e3d1184834461db57", "transactionType": "INBOUND", "custodyType": "ENDUSER", "state": "CONFIRMED", "amounts": [ "10" ], "nfts": null, "txHash": "0xdd2f81a78605dcd4759265c703fb2b4c507c5ea100319338422714bfcde77225", "blockHash": "0x4df6092fdb868331614771ff11944b43051cf6ed1067f8cfa55e9d40ef61426b", "blockHeight": 9423950, "networkFee": "", "firstConfirmDate": "2023-07-28T19:07:24Z", "operation": "TRANSFER", "userId": "2f1dcb5e-312a-4b15-8240-abefc0e3463", "abiParameters": null, "createDate": "2023-07-28T19:07:34Z", "updateDate": "2023-07-28T19:07:37Z" } ] } }
Updated 16 days ago
```

What's Next Congratulations! You've received your first transaction to your user-controlled wallet. To learn how to make a transfer out of your user-controlled wallet, check out the next guide [Send an Outbound Transfer](#) * [Table of Contents](#) * [1. Acquire a Session Token](#) * [2. Acquire the Wallet ID](#) * [3. Transfer Testnet Currency](#)