Transfer Tokens from Wallet to Wallet

Transfer tokens across two developer-controlled wallets you've already created.Suggest Edits

This guide outlines transferring tokens across two developer-controlled wallets. If you do not already have two wallets, go to thereate your first wallets tutorial, where we guide you through it step by step. Also, if you do not have any tokens in your wallet, go to theinbound transfer guide.

Throughout this guide, you will utilize API requests. These can be made using cURL requests inline or Circle's API references. For instructions on navigating the API references, please consult thisuide .

Note: The following tutorial uses the Polygon MumbaiTestnet for illustration purposes. In production, you can create and use programmable wallets that support crypto tokens with the following blockchains and standards:

- Ethereum (ETH), Polygon (MATIC), and Avalanche (AVAX), both Testnet and Mainnet
- ERC-20 tokens
- ERC-721 and ERC-1155 NFTs (non-fungible tokens)

1. Check the Wallet's Balance

Before sending funds across wallets, you will need to do two more things:

1. Ensure the wallet has received the USDC or MATIC tokens.
2. Obtain the token ID for the received USDC or MATIC. The token ID is an assigned identifier that will be used for the token transfer in the upcoming step.

This can be done by making a request toGET /wallets/{id}/balances providing the ID of Wallet One from the previous step.

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.getWalletTokenBalance({ id: 'ce714f5b-0d8e-4062-9454-61aa1154869b' }); curl --request GET \ --url 'https://api.circle.com/v1/w3s/wallets/{id}/balances' \ --header 'accept: application/json' \ --header 'authorization: Bearer ' Response Body { "data": { "tokenBalances": [{ "token": { "id": "e4f549f9-a910-59b1-b5cd-8f972871f5db", // HERE "blockchain": "MATIC-MUMBAI", "name": "Polygon-Mumbai", "symbol": "MATIC-MUMBAI", "decimals": 18, "isNative": true, "updateDate": "2023-06-29T02:37:14Z", "createDate": "2023-06-29T02:37:14Z" }, "amount": "0", "updateDate": "2023-11-17T16:54:55Z" }, { "token": { "id": "7adb2b7d-c9cd-5164-b2d4-b73b088274dc", // HERE "blockchain": "MATIC-MUMBAI", "tokenAddress": "0x9999f7fea5938fd3b1e26a12c3f2fb024e194f97", "standard": "ERC20", "name": "USD Coin", "symbol": "USDC", "decimals": 6, "isNative": false, "updateDate": "2023-10-18T14:29:44Z", "createDate": "2023-10-18T14:29:44Z" }, "amount": "10", "updateDate": "2023-11-17T16:54:55Z" }] } }

2. Gas Fees (EOA Wallets Only)

In Web3,gas fees refers to transaction fees that are always paid in the native currency of the blockchain network. For example, if you send an ERC-20 token on the Ethereum network, you must pay the transaction fee in Ether (ETH). Similarly, on the Polygon network, the fee would be in MATIC. To ensure smooth transactions, it is important to have a sufficient amount of the native token in your wallet.

To gather tokens for gas and send MATIC from wallet to wallet, we recommend using thePolygon Faucet . When you visit the faucet's website, you will be prompted to provide your wallet's address. To acquire it, make a request toGET /wallets as shown below noting down Wallet One's address wallets[0].address .

Node.js cURL const response = await circleDeveloperSdk.listWallets({}); curl --request GET \ --url 'https://api.circle.com/v1/w3s/wallets' \ --header 'accept: application/json' \ --header 'authorization: Bearer ' Response Body { "data": { "wallets": [{ "id": "17c411ba-7aef-4c5d-a4f8-951a60ce7fb1", "state": "LIVE", "walletSetId": "018b42d3-8934-7ad5-8ec7-a34179a7e6e5", "custodyType": "DEVELOPER", "address": "0x09e768def76316d8f127efb73047a046598fef7b", "blockchain": "MATIC-MUMBAI", "accountType": "EOA", "updateDate": "2023-10-18T12:48:45Z", "createDate": "2023-10-18T12:48:45Z" }, { "id": "8172d817-32b5-4f52-b071-4a8d789e08fc", "state": "LIVE", "walletSetId": "018b42d3-8934-7ad5-8ec7-a34179a7e6e5", "custodyType": "DEVELOPER", "address": "0x48f2d63f13f62ff1e3ab9dbda56f3e5c531557b2", "blockchain": "MATIC-MUMBAI", "accountType": "EOA", "updateDate": "2023-10-18T12:48:45Z", "createDate": "2023-10-18T12:48:45Z" }] } }

3. Transfer Tokens

Now, you can make on-chain transfers from wallet to wallet. This is done by making a request toPOST /developer/transactions/transfer with the following request body parameters.

1. idempotencyKey
2. : Universally unique identifier (UUID v4) idempotency key. This key is utilized to ensure exactly-one execution of a mutating requests.
3. tokenId
4. : Token ID for the token that will be transferred. In this case, you will provide the MATIC token ID from the previous step.
5. walletId
6. : Wallet ID of the source wallet where the token amount will be transferred from. In this case, you will provide Wallet One's ID
7. wallets[0].id
8. from step one.
9. destinationAddress
10. : Wallet address that will receive the tokens. In this case, use Wallet Two's address
11. wallets[1].id
12. from step one.
13. amounts
14. : The amount of tokens to be sent. In this case, we will use a minimal amount.
15. feeLevel
16. : A dynamic blockchain fee level setting (LOW, MEDIUM, or HIGH) that will be used to pay gas for the transaction. Calculated based on network traffic, supply of validators, and demand for transaction verification. Estimates for each fee level can be obtained through the estimate fee API call.
17. entitySecretCiphertext
18. : The Entity Secret Ciphertext is an RSA encryption value generated from your Entity Secret and Circle's public key. This asymmetrically encrypted value is sent in API requests like wallet creation or transaction initiation to ensure secure critical actions. This process enables secure usage of the Entity Secret to ensure it cannot be easily accessed or misused.

Node.js cURL const response = await circleDeveloperSdk.createTransaction({ walletId: 'ce714f5b-0d8e-4062-9454-61aa1154869b', tokenId: 'e4f549f9-a910-59b1-b5cd-8f972871f5db', destinationAddress: '0xc90e058234d4b2db799d787a855ec68d801a53a3', amounts: ['.01'], fee: { type: 'level', config: { feeLevel: 'MEDIUM' } } }); curl --request POST \ --url 'https://api.circle.com/v1/w3s/developer/transactions/transfer' \ --header 'accept: application/json' \ --header 'content-type: application/json' \ --header 'authorization: Bearer ' \ --data ' { "idempotencyKey": "a0eebc99-9c0b-4ef8-bb6d-6bb9bd380a11", "walletId": "ce714f5b-0d8e-4062-9454-61aa1154869b", "tokenId": "e4f549f9-a910-59b1-b5cd-8f972871f5db", "destinationAddress": "0xc90e058234d4b2db799d787a855ec68d801a53a3", "amounts": [".01"], "feeLevel": "MEDIUM", "entitySecretCiphertext": "XVXDshXQAjJSkDX1+9veL+pRYaPhYlab4os7SoLPwgXfDym6bF5fW7l07WAB4jSzkQombZ3bfDKEcLhg2C296WCuD7kJM8e6UNrzTeC3d6Pkv+BlOf45N8j/6njAGaBFIu9TVbmSSvK/Nxj0IwTOrEAum3 } ' Response Body { "data": { "id": "1af639ce-c8b2-54a6-af49-7aebc95aaac1", "state": "INITIATED" } }

4. Check the Transfer State

Transfers are a type of blockchain and, therefore, are asynchronous. To ensure the transfer has successfully reached its destination, make a request toGET /transactions/{id} using the ID from the transfer in the previous step.

Node.js cURL const response = await circleDeveloperSdk.getTransaction({ id: '1af639ce-c8b2-54a6-af49-7aebc95aaac1' }); curl --request GET \ --url 'https://api.circle.com/v1/w3s/transactions/{id}' \ --header 'accept: application/json' \ --header 'authorization: Bearer ' Response Body { "data": { "transaction": { "id": "1af639ce-c8b2-54a6-af49-7aebc95aaac1", "blockchain": "MATIC-MUMBAI", "tokenId": "e4f549f9-a910-59b1-b5cd-8f972871f5db", "walletId": "ce714f5b-0d8e-4062-9454-61aa1154869b", "sourceAddress": "0xf5c83e5fede8456929d0f90e8c541dcac3d63835", "destinationAddress": "0xc90e058234d4b2db799d787a855ec68d801a53a3", "transactionType": "OUTBOUND", "custodyType": "DEVELOPER", "state": "COMPLETE", "amounts": ["0.01"], "nfts": null, "txHash": "0xc0337cad6164f22eeb37af3e21488f77e7bc5ba4a92a2293e0a8bb8beaa61c88", "blockHash": "0xbcaf5d0dc5d50250a137808ca81eb4abebf13a639398c44660b35e387e5ad2d3", "blockHeight": 38738377, "networkFee": "0.000035910000189", "firstConfirmDate": "2023-08-07T14:41:02Z", "operation": "TRANSFER", "feeLevel": "MEDIUM", "estimatedFee": { "gasLimit": "21000", "baseFee": "0.000000016", "priorityFee": "1.709999993", "maxFee": "1.710000025" }, "refId": "", "abiParameters": null, "createDate": "2023-08-07T14:40:54Z", "updateDate": "2023-08-07T14:42:04Z" } } } Once the transaction.state is COMPLETE you can rest assured that the token has moved from wallet one to wallet two.

You can also check the transaction using the txHash orPolygon Mumbai explorer and inspect the balance of each wallet usingGET /wallets/{id}/balances . Updated2 months ago *Table of Contents * *1. Check the Wallet's Balance* *2. Gas Fees (EOA Wallets Only)* *3. Transfer Tokens* *4. Check the Transfer State