

Reset Account Pin Code

Walk a user through the steps in resetting a known account PIN code [Suggest Edits](#)

This guide outlines how to enable a user to reset their PIN code. Note users must know their original PIN code to reset it. Otherwise, they must follow the Account Recovery flow outlined in [this side](#).

Caution : If a user loses both their PIN code and the answers to their Security Questions, they will be permanently locked out of their account, losing access to all of their wallets and assets.

1. Run the Sample App

After you set up one of the web, iOS, or Android [sample applications](#), you can:

1. Run the sample app and simulator.
2. Obtain your App ID. To get the App ID, do one of the following:
 1. Access the developer console and navigate to the [configurator](#)
3.
 1. within user-controlled wallets. From there, copy the App ID.
4.
 1. Make an API request to [GET /config/entity](#)
5.
 1. and copy the App ID from the response body.
6. Add the App ID to the sample app.

2. Acquire a Session Token

Next, you need to acquire a session token. To do this, you make a request to the [POST /users/token](#) using the previously created userId in Step 1. The userToken is a 60-minute session token used to initiate requests requiring a user challenge (PIN code entry). After 60 minutes, the session expires, and a you must generate a new userToken using the same endpoint.

From this response, you can acquire the encryptionKey and userToken which you provide in the respective sample app fields. You also use the userToken in the next step.

```
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-abeffc0e3463' }); 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-abeffc0e3463" }' Response body { "data": { "userToken": "eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCQ9.eyJkZXZlbG9wZXJFbnRpdHIFbnZpcm9ubWVudCI6IiRlRlU1Q1ILCjI6IjRIMDdhOGM5LTlxOTAiNDVINC1hNjc0LWQyMGFkNjg4MWI3YyIsImV4cCjUR8i4zMmfdURw3FFcQldSbm-BUg6M7FP_fp-cs9xBbNmRZa31gMd1aKdcajJ9SvVrUowYfGXM3VcNF8rtTFtW-gk1-KzU4u10U35XXbbMcW1moxE0Rqx_fKotDgk2VdITuuds5d5TiQzAXECqeCOctNoDKktMkgllbnLxOaRI2ReZjGt-ctD2V0DbYNO4T_ndPSUDI6qD7dXQRed5uDcezJYoha3Qj3iFGBglEnox2Y6DWTBllqjwmfTGrU8Pr0yz4jQz7suGwmiCzHPxcpYxMzYQ", "encryptionKey": "Tlcyxz7Ts9ztRLQq5+pic0MIETbYimOo2d7idV/UFM=" } }
```

3. Initialize PIN reset and Acquire Challenge ID

Make a request to [PUT /user/pin](#) using the userToken returned from Step 1. This call returns a challengeId, which is used with the Circle Programmable Wallet SDK to have the user reset their PIN code.

```
Node.js cURL const response = await circleUserSdk.updateUserPin({ userToken: " " }); curl --request PUT \ --url 'https://api.circle.com/v1/w3s/user/pin' \ --header 'accept: application/json' \ --header 'content-type: application/json' \ --header 'authorization: Bearer ' \ --header 'X-User-Token: ' Response Body { "data": { "challenge": { "id": "c4d1da72-111e-4d52-bdbf-2e74a2d803d5", "correlationIds": [ "54399e5a-1bf6-4921-9559-10c1115678cd" ], "status": "COMPLETED", "type": "CHANGE_PIN" } } } Updated 16 days ago
```

4. Reset the PIN in the Sample App

Using the sample application, enter the userToken and secretKey returned from Step 1. Enter the challengeId returned from Step 2.

You should be ready to execute the reset PIN code workflow via the Circle Programmable Wallet SDK. Once you've entered the required fields indicated in Step 3, clickExecute to continue.

The sample application takes you through the authentication and PIN code reset process, which includes the user entering their PIN code to authorize the reset and entering a new PIN code.

5. Check the Challenge Status

Make a request [GET /user/challenges/{id}](#) using the challengeId received from Step 2 to retrieve the status of the challenge. Additionally, Circle sends a notification to [subscribed endpoint](#) once the PIN code reset is complete. For a full list of possible statuses, see the [Asynchronous States and Statuses guide](#).

```
Node.js cURL const response = await circleUserSdk.getUserChallenge({ userToken: " " }); curl --request GET \ --url 'https://api.circle.com/v1/w3s/user/challenges/{id}' \ --header 'accept: application/json' \ --header 'content-type: application/json' \ --header 'authorization: Bearer ' \ --header 'X-User-Token: ' Response Body { "data": { "challenge": { "id": "c4d1da72-111e-4d52-bdbf-2e74a2d803d5", "correlationIds": [ "54399e5a-1bf6-4921-9559-10c1115678cd" ], "status": "COMPLETED", "type": "CHANGE_PIN" } } } Updated 16 days ago
```

What's Next The user has successfully reset their PIN code! Move on to recover an account guide to complete: [Recover an Account](#) * [Table of Contents](#) * * [1. Run the Sample App](#) * * [2. Acquire a Session Token](#) * * [3. Initialize PIN reset and Acquire Challenge ID](#) * * [4. Reset the PIN in the Sample App](#) * * [5. Check the Challenge Status](#)