

# Overview

Learn how to use the Programmable Wallets User-Controlled Web, iOS, and Android client-side SDKs [Suggest Edits](#)

Circle designed the Programmable Wallets SDK to guarantee that end-users have complete control over the private keys of their [user-controlled wallets](#). Users create their own PIN code and security questions upon account creation to ensure their access, control, and recovery processes are in place to use their wallets.

## Installation and SDK References

Go to one of the following guides to set up your local environment and review the SDK API references:

- [Web SDK guide](#)
- [Android SDK guide](#)
- [iOS SDK guide](#)

You can use Circle's [sample applications](#) to test the functions of the Programmable Wallets SDK.

## SDK Github Repositories

- [Web SDK Repository](#)
- [iOS SDK Repository](#)
- [Android SDK Repository](#)

## What is a PIN?

The user's PIN is a 6-digit code chosen by the user during the user initialization process. It is required to authorize interactions with their wallet. The PIN is encrypted during input on the user's personal device, ensuring developers never have access to user PINs.

Users provide their PIN to access and authorize the movement of assets. It is vital that a user remembers their PIN code and never shares it with another party.

If the user inputs the incorrect PIN more than three times, the PIN locks, and they will need to wait 30 minutes to unlock.

## What are Security Questions?

The user's security questions are set during the initialization process, allowing the user to input answers to their chosen questions, which can be used in the wallet recovery process. If users lose their PIN code, they can unfreeze their wallets by going through the recovery process, which requires them to answer the previously chosen security questions correctly.

If the user inputs incorrect security answers more than three times, the recovery locks, and they must wait 30 minutes to unlock.

## SDK Architecture

The SDK infrastructure uses the following design flow:

1. A user performs a request within their user-controlled wallet on the developer's platform. For example, a user initiates a USDC transfer to an external wallet.
2. The developer requests a user's transaction through Circle's Programmable Wallets RESTful APIs.
3. Circle's RESTful APIs return a response containing a challenge ID. The challenge must be completed to execute the requested transaction.
4. The developer passes the challenge ID to the Programmable Wallets SDK, prompting users to input their PIN code to complete the challenge.
5. The user completes the challenge by entering their PIN code, and the encrypted input is delivered to Circle's SDK APIs.
6. Circle delivers a callback notification to the developer (or the developer queries the corresponding GET endpoint for the given transaction) to receive status updates on the requested transaction.
7. Upon completion of the request, the developer notifies the end-user within the developer's application. Updated 16 days ago

What's Next

- [iOS](#)
- [Android](#)

- [Sample Applications](#)
- [Table of Contents](#)
- - [Installation and SDK References](#)
- - [SDK Github Repositories](#)
- - [What is a PIN?](#)
- - [What are Security Questions?](#)
- - [SDK Architecture](#)