D4D4D4;--ch-t-background: #1E1E1E;--ch-t-lighter-inlineBackground: #1e1e1ee6;--ch-t-editor-background: #1E1E1E;--ch-t-editor-foreground: #D4D4D4;--ch-t-editor-rangeHighlightBackground: #ffffff0b;--ch-t-editor-infoForeground: #3794FF;--ch-t-editor-selectionBackground: #264F78;--ch-t-focusBorder: #007FD4;--ch-t-tab-activeBackground: #1E1E1E;--ch-t-tab-activeForeground: #ffffff;--ch-t-tab-inactiveBackground: #2D2D2D;--ch-t-tab-inactiveForeground: #ffffff80;--ch-t-tab-border: #252526;--ch-t-tab-activeBorder: #1E1E1E;--ch-t-editorGroup-border: #444444;--ch-t-editorGroupHeader-tabsBackground: #252526;--ch-t-editorLineNumber-foreground: #858585;--ch-t-input-background: #3C3C3C;--ch-t-input-foreground: #D4D4D4;--ch-t-icon-foreground: #C5C5C5;--ch-t-sideBar-background: #252526;--ch-t-sideBar-foreground: #D4D4D4;--ch-t-sideBar-border: #252526;--ch-t-list-activeSelectionBackground: #094771;--ch-t-list-activeSelectionForeground: #fffffe;--ch-t-list-hoverBackground: #2A2D2E; }

Onramp with Stripe

TheStripe fiat-to-crypto onramp service(opens in a new tab) allows you to integrate a secure widget into your application that enables users to purchase cryptocurrencies using their credit card or bank account.

Prerequisites

1. Node.js and npm(opens in a new tab)
2. Stripe account(opens in a new tab)
3. A web application using your favorite CLI and language. For exampleReact with NextJS(opens in a new tab)
4. ,Vue with Nuxt(opens in a new tab)
5. orSvelte with SvelteKit(opens in a new tab)
6. .
7. Adeployed Safe
8. for your users.

Integrate the Stripe fiat-to-crypto onramp widget

Obtain your public and private keys

To use the Stripe fiat-to-crypto onramp service, you need to obtain yourpublic and private keys(opens in a new tab) . You have to apply for the crypto onramp service and add at least your business address and information to your Stripe account. When your application is approved, you will find your public and private keys in yourStripe Developer Dashboard(opens in a new tab) .

Install dependencies

First, install Stripe's client library.

npm yarn pnpm _10 npm install --save @stripe/stripe-js @stripe/crypto

Generate a newclient_secret

To authenticate your users, you need to generate aclient_secret to initialize the Stripe widget. For this, you must make an API request to theStripe API(opens in a new tab) using your Stripe private key. It will return a uniqueclient_secret that you can use to initialize the Stripe widget for your users.

To ensure you don't leak your private key, you should make the request from your backend. The backend can then send theclient_secret to your front end. You can use theStripe server example(opens in a new tab) as a starting point for your backend.

Here is how you generate a crypto onramp session using your private key:

TypeScript curl _16 const stripeSessionResponse = await fetch(_16 'https://api.stripe.com/v1/crypto/onramp_sessions', _16 { _16 method: 'POST', _16 headers: { _16 'Content-Type': 'application/x-www-form-urlencoded', _16 Authorization: _16 'Bearer sk_test_51...Eg7o' // your private key for Stripe _16 }, _16 // optional parameters, for example the users' Safe address _16 body: 'wallet_addresses[ethereum]=0x3A16E3090e32DDeD2250E862B9d5610BEF13e93d' _16 } _16) _16 _16 const decodedResponse = await stripeSessionResponse.json() _16 const clientSecret = decodedResponse['client_secret']

Initialize the Stripe widget

The Stripe widget is a secure iframe that allows users to purchase cryptocurrencies.

You can initialize the Stripe widget using theclient_secret you obtained from the previous step:

TypeScript HTML _10 import { loadStripeOnramp } from '@stripe/crypto' _10 _10 // StripeOnramp is imported with the scripts from step one _10 const stripeOnramp = await loadStripeOnramp(_10 'pk_test_51...GgYH' _10) _10 _10 // Use the client secret from the previous step _10 const onrampSession = stripeOnramp.createSession({ clientSecret }) _10 onrampSession.mount('#onramp-element')

Listen to Stripe events

You can listen to thefrontend events(opens in a new tab) from the Stripe widget to update your UI or handle errors.

TypeScript _17 // Listen to events using the onrampSession object from one of the previous step _17 onrampSession.addEventListener('onramp_ui_loaded', event => { _17 console.log('Onramp UI loaded:', event) _17 }) _17 _17 onrampSession.addEventListener('onramp_session_updated', event => { _17 console.log('Onramp session updated:', event) _17 }) _17 _17 // For modal overlay render mode only _17 onrampSession.addEventListener('onramp_ui_modal_opened', event => { _17 console.log('Onramp UI modal opened:', event) _17 }) _17 _17 onrampSession.addEventListener('onramp_ui_modal_closed', event => { _17 console.log('Onramp UI modal closed:', event) _17 }) Now, Stripe will render the widget in theonramp-element div, allowing users to purchase cryptocurrencies securely.

Test the Stripe widget

Test customer data

In production, each customer should pass an individual KYC process, but you should probably test your application before that. You can use the following test data to bypass the KYC process while intest mode(opens in a new tab) . Make sure to select USD as the currency to buy cryptocurrency with.

Field Value Description Email your-email@example.com Use any test or fake emails Phone Number +18004444444 Use +18004444444 for phone number OTP Verification Code 000000 Use 000000 for the OTP verification code First Name John Use any first name Last Name Verified Use Verified for the last name Birthday January 1, 1901 Use January 1, 1901 for successful identity verification Identification Type Social Security Number Select Social Security Number for identification type Identification Number 000000000 Enter 000000000 to fill the identification number field Country United States Select United States for country Address Line 1 address_full_match Use address_full_match for successful identity verification City Seattle Use Seattle for city State Washington Select Washington for state Zip Code 12345 Use 12345 for zip code Test Credit Card Number 4242424242424242 Use test credit card 4242424242424242 Expiration Date 12/24 Use future expiration date 12/24 CVC 123 Use any CVC 123 Billing Zip Code 12345 Use any zip code 12345 for billing

Example images for KYC and payment method

In the following images, you'll find examples of how to complete the KYC process and setup the payment method for a successful test purchase.

Personal Info

Address

Payment Method

These data will allow you to test the Stripe widget without passing the KYC process.

Conclusion

Well done, you have successfully integrated the Stripe fiat-to-crypto onramp service into your application. Your users can now purchase cryptocurrencies securely within your app.

If you have any questions or encounter any issues, contact theStripe support(opens in a new tab) team.

Monerium Was this page helpful?

Report issue