stripe

Card Payments Quickstart

Securely collect card information from your customers and create a card payment.

Accepting a card payment using Stripe is a two-step process, with a client-side and a server-side action:

1. From your website running in the customer's browser, Stripe securely collects your customer's payment information and returns a representative *token*. This, along with any other form data, is then submitted by the browser to your server.
2. Using the token, your server-side code makes an API request to create a charge and complete the payment.

> **Supported cards**
>
> Users in the United States can accept Visa, Mastercard, American Express, Discover, JCB, and Diners Club credit and debit cards.
>
> Stripe also supports a range of additional payment methods, depending on the country of your Stripe account.

Tokenization ensures that no sensitive card data ever needs to touch your server so your integration can operate in a PCI compliant way.

Step 1: Securely collecting payment information

The simplest way for you to securely collect and tokenize card information is with Checkout. It combines HTML, JavaScript, and CSS to create an embedded payment form. When your customer enters their payment information, the card details are validated and tokenized for your server-side code to use.

> **Checkout reference**
>
> Complete information about available options and parameters is provided in the Checkout reference.

To see Checkout in action, click the button below, filling in the resulting form with:

- Any random, syntactically valid email address (the more random, the better)
- One of Stripe's test card numbers, such as **4242 4242 4242 4242**
- Any three-digit CVC code
- Any expiration date in the future
- Any billing ZIP code, such as **12345**

Pay with Card

To get started, add the following code to your payment page, making sure that the form submits to your own server-side code:

```html
<form action="your-server-side-code" method="POST">
  <script
    src="https://checkout.stripe.com/checkout.js" class="stripe-button"
    data-key="pk_test_6pRNASCoBOKtIshFeQd4XMUh"
    data-amount="999"
    data-name="Stripe.com"
    data-description="Example charge"
    data-image="https://stripe.com/img/documentation/checkout/marketplace.png"
    data-locale="auto"
    data-zip-code="true">
  </script>
</form>
```

We've pre-filled the `data-key` attribute with a random test publishable API key. Replace it with your own to test this code through your Stripe account. When you're ready to go live with your payment form, you must replace the test key with your live key. Learn more about how the keys play into test and live modes.

Checkout also accepts the user's ZIP code, if enabled, and passes this to Stripe. Although optional, using address and ZIP code verifications is highly recommended as they'll help reduce fraud. You can also set Checkout to collect the user's full billing and shipping addresses by adding the required attributes to the form.

The amount provided in the Checkout form code is only shown to the user. It does not set the amount that the customer will be charged—you must also specify an amount when making a charge request. As you build your integration, make sure that your payment form and server-side code use the same amount to avoid confusion.

An alternative to the blue button demonstrated above is to implement a custom Checkout integration. The custom approach allows you to use any HTML element or JavaScript event to open Checkout, as well as be able to specify dynamic arguments, such as custom amounts.

Stripe.js and Elements

If you'd prefer to have complete control over the look and feel of your payment form, you can make use of Stripe.js and Elements, our pre-built UI components. Refer to our Elements quickstart to learn more.

Mobile SDKs

Using our native mobile libraries for iOS and Android, Stripe can collect your customer's payment information from within your mobile app and create a token for your server-side code to use.

Step 2: Creating a charge to complete the payment

Once a token is created, your server-side code makes an API request to create a one-time charge. This request contains the token, currency, amount to charge, and any additional information you may want to pass (e.g., metadata).

curl Ruby Python PHP Java Node Go .NET

```php
// Set your secret key: remember to change this to your live secret key in production
// See your keys here: https://dashboard.stripe.com/account/apikeys
\Stripe\Stripe::setApiKey("sk_test_BQokikJOvBiI2HlWgH4olfQ2");

// Token is created using Checkout or Elements!
// Get the payment token ID submitted by the form:
$token = $_POST['stripeToken'];

$charge = \Stripe\Charge::create([
    'amount' => 999,
    'currency' => 'usd',
    'description' => 'Example charge',
    'source' => $token,
]);
```

! These requests expect the ID of the Token API (e.g., **tok_KPte7942xySKBKyrBu11yEpf**) to be provided as the value of the `source` parameter.

Tokens can only be used once, and within a few minutes of creation. Using this approach, your customers need to re-enter their payment details each time they make a purchase. You can also save card details with Stripe for later use. Using this method, returning customers can quickly make a payment without providing their card details again.

Next steps

Congrats! You can now accept card payments with Stripe using Checkout. You may now want to check out these resources:

- Creating charges
- Getting paid
- Managing your Stripe account
- Supported payment methods
- Saving cards

Questions?

We're always happy to help with code or other questions you might have! Search our documentation, contact support, or connect with our sales team. You can also chat live with other developers in #stripe on freenode.