

# Detailed Checkout Guide

[Checkout](#) is the best payment flow, on web and mobile. Checkout builds on top of [Stripe.js](#) to provide your users with a streamlined, mobile-ready payment experience that is constantly improving. If you need help after reading this, check out our answers to [common questions](#) or chat live with other developers in [#stripe](#) on freenode.

Try out Checkout right now by clicking the button below! Use our [test card number](#)—4242 4242 4242 4242, any future month and year for the expiration, any three-digit number for the CVC, and any random postal code.

Pay with Card

Checkout securely accepts your customer's payment details and directly passes them to Stripe's servers. Stripe returns a *token* representation of those payment details, which can then be submitted to your server for use. Checkout doesn't actually create charges—it only creates tokens. You can use those tokens to create the actual [charge on your server](#). Or you can [save the card](#) for charging later, or sign the user up for [recurring charges](#).

## Features

- Custom [logo](#) and [text](#)
- One-click payment through [Remember Me](#)
- Optional [billing](#) and [shipping](#) address
- Desktop, tablet, and mobile support

## Integration

You can integrate Checkout in as little as a single line of client-side code. As we release new Stripe features, we'll automatically roll them out to your existing Checkout integration, so that you will always be using our latest technology without needing to change a thing.

Checkout supports two different integrations:

**Simple:** The simple integration provides a blue **Pay with Card** button as shown above. Upon completion of the payment form and receipt of the [token](#), Checkout stores the token within a hidden input in your payment form and automatically submits the form for server-side use.

## Example integrations

We have Checkout integration guides for several platforms and languages to get you up and running as quickly as possible:

- [Sinatra](#)
- [Rails](#)
- [Flask](#)
- [PHP](#)

**Custom:** The custom integration lets you create a custom button and passes a Stripe [token](#) to a JavaScript callback. Your JavaScript callback will need to send the token to your server for use.

## Simple Custom

The simple integration uses a script tag inside your payment form to render the blue Checkout button. Upon completion of the Checkout process, Checkout submits your form to your server, passing along a `stripeToken` and any elements your form contains. When adding the following code to your page, make sure that the form submits to your own [server-side code](#) within the **action** attribute:

```
<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="2000"
    data-name="Stripe.com"
    data-description="2 widgets"
    data-image="https://stripe.com/img/documentation/checkout/marketplace.png"
    data-locale="auto"
    data-zip-code="true">
  </script>
</form>
```

We've placed a random API key in the code. Replace it with your [actual publishable API key](#) to test this code through your Stripe account.

The above configuration also accepts the user's postal code, when applicable, and passes this to Stripe. Although optional, using [address and postal code verifications](#) is highly recommended as they'll help reduce fraud.

**Note:** Checkout must be loaded directly from <https://checkout.stripe.com/checkout.js>. Using a local copy of Checkout is unsupported, and may result in user-visible errors.

## Received parameters

The following parameters are submitted to your form's action endpoint, along with any other elements in your form, once Checkout completes.

PARAMETER	DESCRIPTION
<code>stripeToken</code>	The ID of the token representing the payment details
<code>stripeEmail</code>	The email address the user entered during the Checkout process

PARAMETER	DESCRIPTION
<code>stripeBillingName</code>	Billing address details (if <a href="#">enabled</a> )
<code>stripeBillingAddressLine1</code>	
<code>stripeBillingAddressZip</code>	
<code>stripeBillingAddressState</code>	
<code>stripeBillingAddressCity</code>	
<code>stripeBillingAddressCountry</code>	
<code>stripeShippingName</code>	Shipping address details (if <a href="#">enabled</a> )
<code>stripeShippingAddressLine1</code>	
<code>stripeShippingAddressZip</code>	
<code>stripeShippingAddressState</code>	
<code>stripeShippingAddressCity</code>	
<code>stripeShippingAddressCountry</code>	

## Configuration options

Change how Checkout looks and behaves using the following configuration options.

### Required

<code>data-key</code>	Your publishable key (test or live)
<code>token</code> (Only available with the custom integration)	The callback to invoke when the Checkout process is complete <code>function(token, args)</code>  <code>token</code> is the <a href="#">Token object</a> created  <code>token.id</code> can be used to create a charge or can be attached to a customer  <code>token.email</code> contains the email address entered by the user  <code>args</code> is an object containing the <a href="#">billing</a> and <a href="#">shipping</a> addresses, if enabled ( <a href="#">see example</a> )

### Highly recommended

<code>data-image</code>	A relative or absolute URL pointing to a square image of your brand or product. The recommended minimum size is 128x128px. The supported image types are: <code>.gif</code> , <code>.jpeg</code> , and <code>.png</code> .
-------------------------	--

<b>data-name</b>	The name of your company or website
<b>data-description</b>	A description of the product or service being purchased
<b>data-amount</b>	The amount (in cents) that's shown to the user. Note that you will still have to explicitly include the amount when you create a charge using the API.
<b>data-locale</b>	Specify <b>auto</b> to display Checkout in the <a href="#">user's preferred language</a> , if available. English will be used by default.
<b>data-zip-code</b>	Specify whether Checkout should validate the billing postal code (true or false). The default is false, but we highly recommend setting to true.
<b>data-billing-address</b>	Specify whether Checkout should collect the user's billing address (true or false). The default is false.

## Optional

<b>data-currency</b>	The currency of the <a href="#">amount</a> (3-letter <a href="#">ISO code</a> ). The default is USD.
<b>data-panel-label</b>	The label of the payment button in the Checkout form (e.g. <b>Subscribe</b> , <b>Pay {{amount}}</b> , etc.). If you include <b>{{amount}}</b> in the label value, it will be replaced by a localized version of <b>data-amount</b> . Otherwise, a localized <b>data-amount</b> will be appended to the end of your label. Checkout does not translate custom labels to the user's preferred language.
<b>data-shipping-address</b>	Specify whether Checkout should collect the user's shipping address (true or false). The default is false.
<b>data-email</b>	If you already know the email address of your user, you can provide it to Checkout to be prefilled.
<b>data-label</b>	The text to be shown on the blue button. Default is <b>Pay with Card</b> . Checkout does not currently translate this label.
<b>data-allow-remember-me</b>	Specify whether to include the option to "Remember Me" for future purchases (true or false). The default is true.
<b>data-bitcoin</b>	Specify whether to accept <a href="#">Bitcoin</a> (true or false). The default is false.
<b>data-alipay</b>	Specify whether to accept <a href="#">Alipay</a> ("auto", true, or false). The default is false.
<b>data-alipay-</b>	Specify if you need <a href="#">reusable access to the customer's Alipay account</a> (true or false). The default is

<b>reusable</b>	false.
<b>opened</b>	<b>function()</b> (Only available with the custom integration)
	The callback to invoke when Checkout is opened (not supported in IE6 and IE7)
<b>closed</b>	<b>function()</b> (Only available with the custom integration)
	The callback to invoke when Checkout is closed (not supported in IE6 and IE7). Called after the <a href="#">token</a> callback (for successful tokenizations).

## Supported languages

In addition to English, Checkout supports the following languages:

Simplified Chinese (zh)

Danish (da)

Dutch (nl)

English (en)

Finnish (fi)

French (fr)

German (de)

Italian (it)

Japanese (ja)

Norwegian (no)

Spanish (es)

Swedish (sv)

Custom strings passed to Checkout (e.g., `name` , `description` , `panelLabel` and `label` ) are not automatically translated.

Checkout also uses the locale to format numbers and currencies. For example, when loading Checkout with currency set to **EUR** and locale set to **auto**, a browser configured to use English (en) could see **€25.00** while

one configured for Hungarian (hu) would see **25,00 €**. If you provide a specific locale, Checkout will use that locale for number and currency formatting.

[Alipay](#) is only supported in English and Chinese. Users of other languages will see Alipay in Chinese but will still see the option to complete a card payment in their native language.

---

## FAQ

**Does Checkout require HTTPS?**

**Does Checkout verify that credit cards are valid?**

**What browsers does Checkout support?**

**How do I prevent the Checkout popup from being blocked?**

---

## Questions?

We're always happy to help with code or other questions you might have! [Search](#) our site for more information or [send us an email](#)!