

Accepter un paiement

1. Installer le serveur

- Installer la bibliothèque PHP de Stripe

Installer la bibliothèque avec composer et initialisez là avec votre clé API privée. Sinon, si vous utilisez scratch et avez besoin d'un fichier composer.json, téléchargez les documents avec le lien de téléchargement dans l'éditeur de code.

- Créer une session de paiement

Ajoutez une route sur votre serveur qui crée une session de paiement. Une session de paiement contrôle ce que votre client voit sur la page de paiement hébergée par Stripe, tel qu'une ligne de produits, la quantité et la devise monétaire de la commande, et une méthode de paiement acceptable. Retournez l'ID de la session de paiement dans la réponse pour référencer la session du client.

- Définir les méthodes de paiement

Le paiement avec Stripe supporte plusieurs méthodes de paiement en plus de celle par carte bancaire. Si de multiples méthodes de paiement peu utilisées, le paiement avec Stripe les réorganise dynamiquement afin de prioriser la méthode de paiement la plus pertinente basée sur l'emplacement du client ou d'autres caractéristiques. Si vous acceptez les cartes bancaires comme une méthode de paiement, Apple Pay et Google Pay sont affichés dans la vérification de Stripe lorsque cela est possible.

- Définir la ligne de produits

Conserver toujours les informations sensibles sur vos produits, comme le prix et la disponibilité, sur votre serveur afin d'éviter que le client les manipule. Définissez les informations du produit quand vous créez la session de paiement avec price_data ou alors utilisez les prix prédéfinis et passez leurs ID.

- Choisir le mode

Il existe trois modes de paiement : le paiement, l'abonnement ou l'installation. Utilisez le mode paiement pour un achat unique. Apprenez en plus sur les modes abonnement et installation dans les documents.

- Fournir les URL de redirection

Fournir les URL pour les pages de confirmation et d'annulation –assurez-vous qu'elles sont publiques pour que Stripe puisse rediriger le client vers-elles. Vous pouvez également manipuler les deux états (confirmation et annulation) avec le même URL.

2. Construit ton paiement

- Ajouter une page de confirmation

Créer une page de confirmation pour l'URL que vous avez fourni en tant que session de paiement success_url pour afficher le message de confirmation de commande ou les détails de la commande à votre client. Stripe redirige vers cette page une fois que le client a terminé le paiement avec succès.

- Ajouter une page d'annulation

Commenté [LF1]: C'est-à-dire la mise en place d'un système qui sauvegarde les informations de la carte bancaire et qui permet de les réutiliser par la suite

Ajoutez une autre page pour `cancel_url`. Stripe redirige sur cette page lorsque l'utilisateur clique sur le bouton retour sur la page de paiement.

- Ajouter une page aperçue de commande
Enfin, ajoutez une page pour afficher l'aperçu de la commande du client. Cela permet au client de revoir ou modifier sa commande—une fois que le client est envoyé sur la page de paiement, la commande est définitive et ne peut être modifiée sans créer une nouvelle session de paiement.
 - Charger `stripe.js`
Le paiement avec Stripe repose sur `stripe.js`, la bibliothèque JavaScript de du fondement de Stripe pour collecter les informations sensibles de paiement et détecter les fraudes avancées. Toujours charger `stripe.js` depuis `js.stripe.com` pour rester conforme. Ne pas inclure le script dans un paquet ou l'héberger vous-même.
 - Ajouter un bouton de paiement
Ajoutez un bouton à votre page aperçu de commande. Lorsque votre client cliquera sur ce bouton, il sera redirigé vers la page de paiement hébergée par Stripe.
 - Initialiser `Stripe.js`
Initialiser `Stripe.js` avec votre clé API public.
 - Chercher une session de paiement
Faites une requête vers la route de votre serveur pour créer une nouvelle session de paiement lorsque le client clique sur le bouton paiement.
 - Rediriger vers le paiement
Appelez `stripe.redirectToCheckout` avec l'ID de la session de paiement pour rediriger le client vers la page de paiement de Stripe.
3. Testez votre page
- Lancer L'application
Démarrez votre serveur et allez sur `http://localhost:4242/checkout.html`
 - Essayez-le
Cliquez sur le bouton de paiement pour être redirigé vers la page de paiement de Stripe. Utilisez n'importe laquelle de ces cartes bancaires test pour simuler un paiement.
 - Paiement réussi – 4242 4242 4242 4242
 - Paiement nécessite une authentification – 4000 0025 0000 3155
 - Paiement refusé – 4000 0000 0000 9995

Partie en Anglais : <https://stripe.com/docs/checkout/integration-builder>

Accept a payment

Set up the server

Install the Stripe PHP library

Install the library with composer and initialize with your secret API key. Alternatively, if you are starting from scratch and need a composer.json file, download the files using the Download link in the code editor.

[Composer](#)

[GitHub](#)

Install the library:

```
composer require stripe/stripe-php
```

Server

Create a Checkout Session

Add an endpoint on your server that creates a [Checkout Session](#). A Checkout Session controls what your customer sees in the Stripe-hosted payment page such as line items, the order amount and currency, and acceptable payment methods. Return the Checkout Session's ID in the response to reference the Session on the client.

Server

Specify payment methods

Checkout supports [several payment methods](#) beyond cards. If multiple payment methods are passed, Checkout dynamically reorders them to prioritize the most relevant payment methods based on the customer's location and other characteristics. If you accept cards as a payment method, Apple Pay and Google Pay are displayed in Stripe Checkout [when applicable](#).

Server

Define the line items

Always keep sensitive information about your product inventory, like price and availability, on your server to prevent customer manipulation from the client. Define product information when you create the Checkout Session with `price_data` or alternatively use [pre-defined prices](#) and pass their IDs.

Server

Choose the mode

Checkout has three modes: `payment`, `subscription`, or `setup`. Use `payment` mode for one time purchases. Learn more about [subscription](#) and [setup](#) modes in the docs.

Server

Supply the redirect URLs

Specify URLs for success and cancel pages—make sure they are publicly accessible so Stripe can redirect customers to them. You can also handle both the success and canceled states with the same URL.

Server

Build your checkout

Add a success page

Create a success page for the URL you provided as the Checkout Session `success_url` to display order confirmation messaging or order details to your

customer. Stripe redirects to this page after the customer successfully completes the checkout.

Client

Add a canceled page

Add another page for `cancel_url`. Stripe redirects to this page when the customer clicks the back button in Checkout.

Client

Add an order preview page

Finally, add a page to show a preview of the customer's order. Allow the customer to review or modify their order—once a customer is sent to the Checkout page, the order is final and cannot be modified without creating a new Checkout Session.

Client

Load Stripe.js

Stripe Checkout relies on Stripe.js, Stripe's foundational JavaScript library for collecting sensitive payment information and advanced fraud detection. Always load Stripe.js from `js.stripe.com` to remain compliant. Do not include the script in a bundle or host it yourself.

Client

Add a checkout button

Add a button to your order preview page. When your customer clicks this button, they're redirected to the Stripe-hosted payment page.

Client

Initialize Stripe.js

Initialize Stripe.js with your publishable API key.

Client

Fetch a Checkout Session

Make a request to the endpoint on your server to create a new Checkout Session when your customer clicks on the checkout button.

Client

Redirect to Checkout

Call `stripe.redirectToCheckout` with the ID of the Checkout Session to redirect the customer to the Stripe Checkout page.

Client

Test your page

Run the application

Start your server and navigate to <http://localhost:4242/checkout.html>

Client

Try it out

Click the checkout button to be redirected to the Stripe Checkout page. Use any of these test cards to simulate a payment.

Payment succeeds

4242 4242 4242 4242

Payment requires authentication

4000 0025 0000 9155

Payment is declined

4000 0000 0000 9995