Build better payment forms using new “embedded” Stripe Checkout

6 MIN · YOUTUBE · HTTPS://WWW.YOUTUBE.COM/WATCH?V=7WFXL4-ACXS
https://www.youtube.com/watch?v=7WFXl4-aCxs

SUMMARY

The video discusses the evolution of payment processing for software businesses, focusing on the integration of Stripe payments. It highlights the importance of a seamless checkout experience and introduces a new method for embedding Stripe's checkout directly into websites, enhancing user experience and brand consistency.

- *Many side hustles fail, but validating ideas through payment forms can reveal product viability.*
- *Poor checkout experiences can deter potential customers, regardless of product quality.*
- *Stripe has optimized payment processing, now handling $1 trillion annually.*
- *A new course teaches how to implement Stripe payments in software as a service (SaaS) products.*
- *Stripe checkout can now be embedded directly into websites, eliminating the need for redirects.*
- *The implementation involves creating a checkout session on the server and using Stripe's SDK for frontend integration.*
- *Developers can customize the embedded checkout form to align with their brand's aesthetics.*
- *The video concludes with a discount code for viewers interested in the full course on Stripe integration.*

over the years I've tried to make money
with a lot of different side hustles
most of them have failed miserably but
some have gone on to make me literally
dozens of dollars if you've ever built a
piece of software as a business or side
hustle one of the best ways to validate
your idea is to slap a payment form on
your website and see if anybody pays you
for
it when a project makes 0 it usually
means one of two things a your product
sucks or B your product is amazing but
the marketing sucks however there is a
third possibility the checkout form
experience is so bad that people just
give up trying to pay you when I first
learned a code implementing payments was
a nightmare you had to deal with this

thing called authorized.net which had a horrible API it was expensive there were no sdks and a million different ways to shoot yourself in the foot security wise it strip payments saw an opportunity to optimize this experience for developers and that's why today the company is now processing $1 trillion in payments every year I just released a new course for fireship pro members about implementing strip payments in a software as a service product and one of the features in that course is stripe checkout which will redirect a user to a web page hosted by stripe when the user wants to buy something on your website one thing that kind of sucks about this approach though is that the user has to be redirected to a thirdparty website it just feels a little bit awkward and breaks the consistency of your brand well luckily a few months ago it became possible to embed stripe checkout directly into your own website which means that user never has to leave in today's video I'll show you exactly what that implementation looks like what you're looking at here is the demo for the full stripe course and when you click this by now button it redirects you to a website on checkout. stripe.com when the payment is complete it then redirects back to the main website that works but we can streamline this process by embedding the checkout form directly in our own website and that just feels totally frictionless like stripe checkout you don't have total control over what you can customize like you can't directly edit the CSS or anything like that but it is responsive out of the box and you can customize things

like colors and fonts to fit in with your brand in this demo I'm using Daisy UI to trigger aot window then render the checkout form inside of that if the modal is wide it goes to a horizontal layout but will shrink down to a vertical layout for mobile devices now let's take a look at the code to see how it's implemented I'm here in a nextjs project which has also been configured with the stripe SDK and the first step is to create a checkout session on the server a checkout session is a Json object created by stripe that contains things like the product details that you're selling it tells stripe what the user is paying for and how to render the checkout form in your UI we could do this with nextjs server actions but I'm going to use a more traditional API route with a route. TS file the route exports a post function which will handle a post request to this URL now inside the function body we'll use the stripe SDK to create a checkout session which includes the price IDs for the products that you're selling but there's a couple of unique details first of all the UI mode is embedded as opposed to hosted on the stripe website then second you'll notice down here we have a return URL as opposed to a success and cancel URL and and it also includes the session ID as a URL parameter and that's because after the payment goes through we'll want to immediately fetch the session ID to check it status so stripe will redirect back to a URL that looks like this then in our front end code we can make a get request to the back end so it grabs the session ID from the URL it then makes a call to stripe checkout

sessions retrieve that returns a big
Json object that will give us the status
of the payment then we can design our UI
around that accordingly that takes care
of our backend code now we need to
figure out how to render the checkout
form in the front end in react stripe.js
provides built-in components to handle
this in nextjs I've created this
embedded checkout button component which
is a client component inside of which it
Imports load stripe as well as the
pre-built react components currently
this component is using an HTML dialogue
to render a modal which is all styled
with tailwind and Daisy UI by the way
but from here we need to make a post
request to the backend API that we just
defined I'm using the browser fetch AP
to do that and also wrapping it in use
callback is so the function doesn't get
redefined every time this component
renders but most importantly the API
needs to return the client secret on the
checkout session now we're not going to
call this function directly but instead
put it as a property on an object called
options which will get passed to the
embedded checkout Provider from stripe
let's go down into the jsx and declare
that component here along with the
options as a prop when this component is
rendered stripe will automatically make
a call to our back back end to get the
client secret along with the details for
the checkout session like the products
the user wants to buy and by the way the
client secret is a value that should
never be stored in your database it's
just a temporary way for stripe to show
the payment form in the front end based
on the data you provided for the

checkout session and now at this point
we should be able to open up the Modo on
the UI and it will render the checkout
page there and you can of course
customize this in a variety of ways to
fit in with your own branding now once
the checkout session is complete it's
going to redirect to a return URL we can
Implement that with a page. TSX file
which is a react server component nextjs
before defining the component I'm first
going to Define our data fetching logic
which needs to get the session ID from
the URL that stripe redirected to now we
could create another API route and make
a fetch call to it but because we're in
a react server component here we can
actually just use the stripe SDK
directly and call the stripe checkout
sessions retrieve method with the
session ID as an argument and now in the
server component all we have to do is
get the search parameter session ID and
pass it as an argument to this function
after we fetch the session from stripe
it's going to have a bunch of data on it
like the session status which might be
open or complete if it's open we'll tell
the user their payment hasn't gone
through yet but if it's complete we'll
send them a thank you message
congratulations you now have a secure
payment form embedded directly on your
website but there's a lot more to
payments in that if you want to learn
everything you need to know check out
the full stripe course and because you
graciously watch this entire video
here's a discount code you can use on
the course itself or on a pro membership
to get get access to everything thank
you for supporting the channel and I

will see you in the next one