# Introduction to the Stripe API and the stripe-node npm package

Hello, everyone, and welcome back.

In this lesson, we are going to continue our implementation of our checkout payment flow.

So the user at this point has clicked here on the buy course button and an Http request has been sent to a server under our control where this route handler createCheckoutSession has been triggered.

We know which course the user wants to purchase now, another thing that we will need to know is who is the user performing the purchase?

We are going to learn how to implement that later on in this course as we secure this endpoint.

Right now, let's get to the most important functionality of this endpoint, which is to perform a Stripe API call in order to initialize a stripe checkout process.

So we need to tell to the Stripe servers that our website wants a customer to purchase a given product with a given description by a given price.

So for that we need to make a signed API call to the Stripe servers.

Now, instead of using an Http client directly, we are going to be using the Stripe node package.

You can find this package here on GitHub under the account stripe/stripe-node.

So this is a nodejs library for the Stripe API.

You can also make all your API calls directly using any Http client, but it's much simpler to simply use the Stripe NodeJS library.

This library gives us a much friendlier user interface.

As you can see, if we want to create a Stripe customer, we simply have to call Stripecustomers.create passing a configuration object.

This package has a promise based API so we can await for this promise to get resolved and we get back here a Stripe customer object.

If you are looking for full documentation on all the API calls that are available, you can find it here under the Stripe API reference under stripe.com/docs/api.

For example, the customer object that we got from that example call, we can find its documentation here under customers.

So we have here the multiple rest calls that we can make to the Stripe endpoint in order to create a customer, delete etcetera.

And this is the documentation of the object that we get after creating a Stripe customer.

As you can see, we have here an ID, we have a creation date and we have here a lot more information about the customer.

If you want to know what is the API for retrieving a given customer, we just have to select it here.

And as we can see, we get here examples on multiple languages.

And let's have a look here at the documentation in Node.

As we can see, we simply have to do a StripeCustomers.retrieve call using the stripe node NPM package.

We need to pass in here the customer ID and we can either provide here a function callback or alternatively, we can also await for the call to retrieve to resolve its promise.

The Stripe node package in general supports both a callback based API and a promise based API.

We recommend always using the promise based API together with the async await TypeScript functionality.

In this API documentation reference, we will be able to find the API call that we need to do to initialize a checkout session.

We can find it here under the checkout section, under the object sessions.

So as you can see, a checkout session represents a customer session as they pay for one time purchases or subscriptions through the checkout stripe payment solution.

So this is the API that we need to use.

We need to create a new Stripe checkout session and here is the documentation of how to do it.

So this is the API that we need to call from our rest endpoint.

We need to call stripe checkout dot sessions, dot create and we need to pass in a configuration object with a series of parameters.

We need to pass in a success, which is the URL to which stripe is going to redirect the customer after a successful payment.

We need to define a cancel URL, which is the URL that stripe will redirect the customer.

If the payment fails, we need to specify a payment method, in this case credit card and we need to provide here some extra configuration telling for example, the description of the product, the amount, the currency, the quantity and so forth.

For activating subscriptions, we need a slightly different configuration.

This API call is doing a one time purchase of a given product, so it's exactly what we need for our buy course button implementation.

Notice that there is a difference here.

We are providing a quantity of two, while in our case we are only buying one course in order to be able to perform this.

We are going to need to initialize the Stripe package.

As you can see, we need to require the Stripe NPM module and we need to pass it here a parameter.

So this parameter that we see here is the private secret key that the Stripe NPM package is going to use to sign the Http request that the NPM package will be creating and sending to the Stripe servers.

So it's this signature using the secret parameter that is going to tell Stripe which server on the internet is initializing this stripe checkout session.

So Stripe needs to know this information in order to know where to transfer the money being withdrawn from the customer credit card.

So before being able to do our Stripe call, we need to create a Stripe account and get the stripe secret key needed to initialize the Stripe NPM package.

Let's learn how to do that in the following lessons.