In this lesson we will set up a stripe account and start to integrate it into a backend server.

We'll also set up a server endpoint that handles data that is POST to.

---

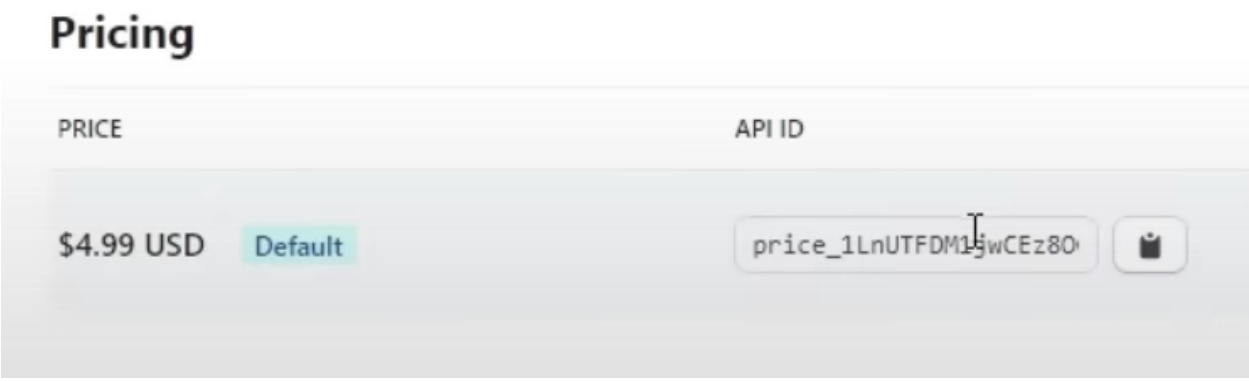
Create an account at Stripe, https://stripe.com/

Once created, copy your secret key, save it somewhere locally so you can access it later.


Next create a product by first clicking the "Products" tab in the navigation menu.

Once in the products section click that "Add a product" button.

Then, upon the product being saved, there will be an API ID that you want to copy. Paste it locally somewhere where you can access it later.




Create two more products for a total of three products.

Next go to your project directory, it should look something like this

```
project/
├── Frontend/
└── Backend/
```

In `cd` into your Backend folder and create a file by the name of Server.js,

Exit out of your Backend folder and go into your project folder (example above) then,

Run the following command `npm init -- yes`

Then run the following npm packages

```
npm install express cors stripe
```

Express, cors, and stripe are all dependencies for allowing a user to complete a purchase on the application.

I recommend you take some time to overview which dependency does seperately.

Now access your Server.js file, *the next code you'll write is JavaScript that is made to communicate with a server.*

*This is what's known as Node.JS.*

The first challenge is to require all the packages you'll need to make your Server.js file work.

The following packages are Required,
`express cors stripe`

Require is a method that is used instead of the Import syntax from React. You can see how it works here

The Require for the stripe package is a bit special so it's included below,

```
const stripe = require('stripe')('SECRET_KEY_HERE')
```

Add this middleware as well below your require statements,

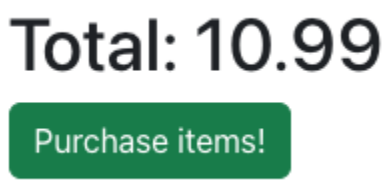
```
app.use(cors());
app.use(express.static('public'));
app.use(express.json());
```

Now what we want to do next is set up an express server endpoint that handles a simple *post*

request from the frontend.

This post request should have included within the `body` of the request a stringified version of your shopping cart.

The post request needs to execute when you click “Purchase Items”




In your Server.js file handle the request and send a HTTP response to the frontend as to whether the request has been a success or not.

To set up the POST route on the server, take a look at the documentation below,

https://expressjs.com/en/guide/routing.html

To set up a function that will send requests to the POST route checkout the url below,

https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API/Using_Fetch

Note the `body` property in the second argument of the fetch method,

```
body: JSON.stringify(data)
```