Headless Ecommerce:

Headless Ecommerce is basically use of Backend api of some provider like shopify and frontend is made by developers themselves. JAMSTACK is best for Ecommerce now a days as it is fast and SEO friendly

Project-01 (Gatsby-stripe-checkout)

This is the most easiest method for connecting payment gateways. Stripe provide everything like payment gateway, checkout page etc. No custom api creation required or no server is involved. Client side is directly connecting to stripe. Means checkout page will be of stripe

Creating Project:

gatsby new gatsby-ecommerce-1-stripe-checkout <https://github.com/gatsbyjs/gatsby-starter-minimal>

Starting project:

netlify dev

or,

gatsby develop

Clean index.js file like below:

import \* as React from "react"

// markup

const IndexPage = () => {

  return (

    <div>

      Hello world

    </div>

  )

}

export default IndexPage

*Dependencies*

Npm I @stripe/stripe-js

Npm I dotenv

*Get your Api or published key:*

Developers -> api (link: [API keys – personal – Stripe [Test]](https://dashboard.stripe.com/test/apikeys) )

pk\_test\_51JhERSGkYc9g1BvdAdto0rO94IZQ6BAqLaShXV1pie6mHqnY0dTObrEPGgBsZRffjvKBvB2SBviBv1FIPQ1Mlpit00Yw4DxR4C

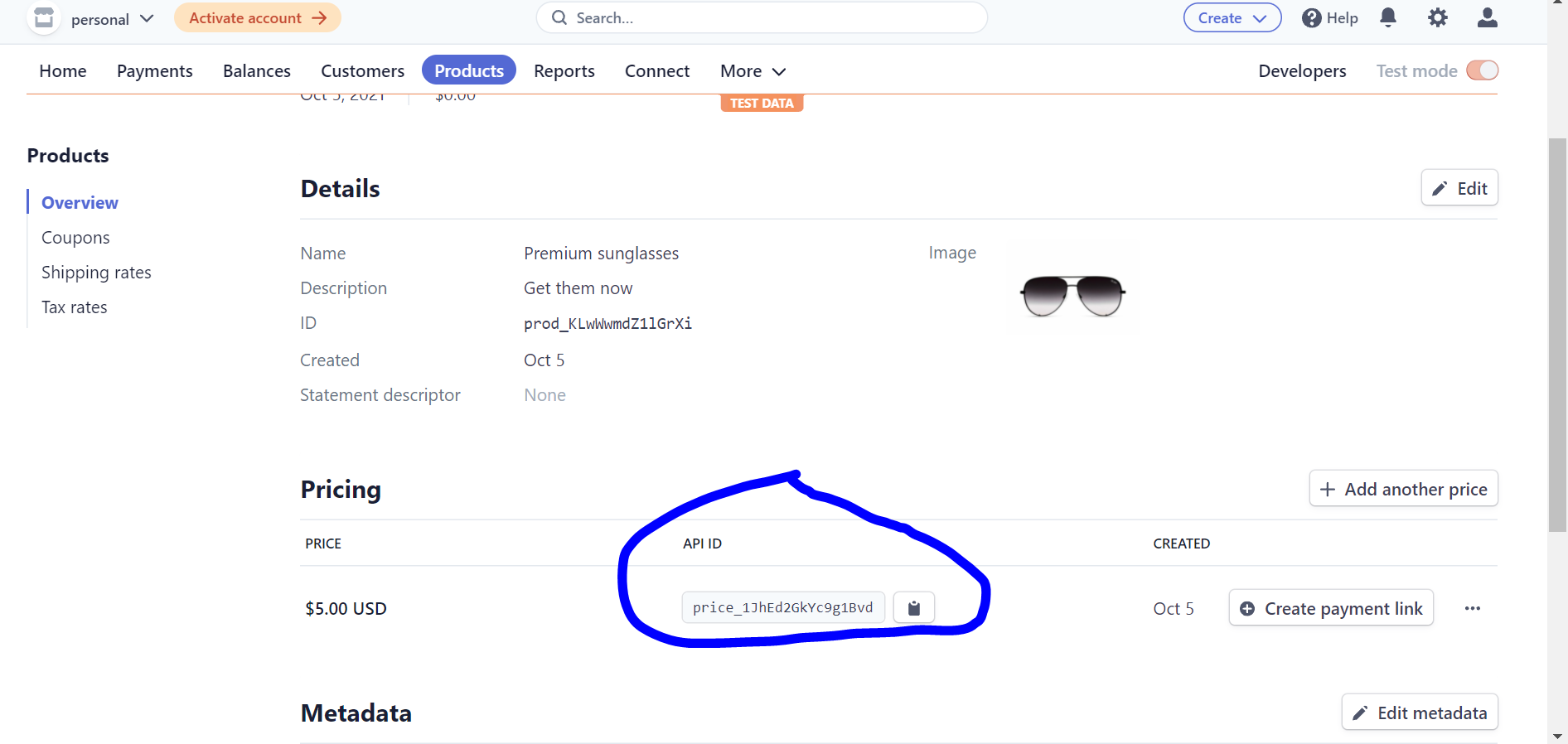
*Important thing:*

1. Create account on stripe
2. Go to below link and as we are integrating client side to stripe, so enable or click on client integration. After that first need to create a product

[Checkout and Payment Links – New Business – Stripe [Test]](https://dashboard.stripe.com/settings/checkout)

1. Go to this link to create your first product. Create 2 3 test products

[Checkout and Payment Links – New Business – Stripe [Test]](https://dashboard.stripe.com/settings/checkout)



After creating product, the highlighted api id is very much important

To get stripe checkout page or basic payment gateway code:

import \* as React from "react"

import { loadStripe } from "@stripe/stripe-js"

const stripePromise = loadStripe("pk\_test\_51JhERSGkYc9g1BvdAdto0rO94IZQ6BAqLaShXV1pie6mHqnY0dTObrEPGgBsZRffjvKBvB2SBviBv1FIPQ1Mlpit00Yw4DxR4C")

// markup

const IndexPage = () => {

  const redirectToCheckout = async () => {

    const stripe = await stripePromise

    const result = await stripe.redirectToCheckout({

      mode: "payment",

      lineItems: [

        { price: "price\_1JhEhQGkYc9g1Bvdhk8kZJ8z", quantity: 3 },

        { price: "price\_1JhEd2GkYc9g1Bvd737V5xW2", quantity: 6 }

    ], //array of objects

      successUrl: `http://localhost:8888/payment-success/`,

      cancelUrl: `http://localhost:8888/payment-error`,

    })

  }

  return (

    <div>

      <div>

      Hello world

      </div>

      <button onClick={redirectToCheckout}>Checkout</button>

    </div>

  )

}

export default IndexPage

Also create payment-succeful.js and payment-error.js so to redirect if payment succeful or failed

Under lineitems, priceid is basically api id from product page as discussed earlier and passed public key to loadstripe