

Payement Stripe

Payement avec Stripe

Site officiel

<https://stripe.com/fr-us>

Après Inscription

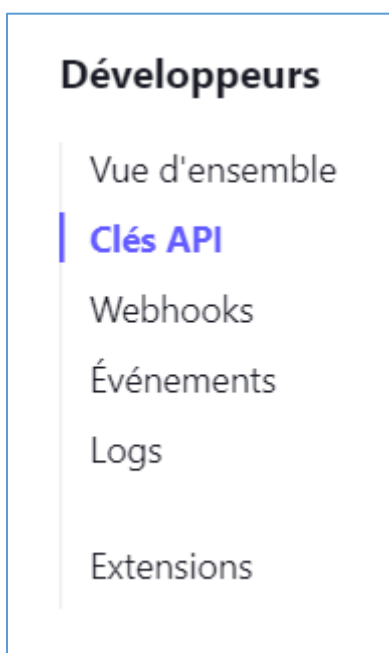
Compte Stripe créé :

hachicha.hs@gmail.com
yekjbx8eaa

Cliquer sur Développeurs (coin en haut à droite)



Puis cliquer sur Clés API (menu à gauche)



Puis copier clé publique :

Clés standard		
Ces clés vous permettront d'authentifier les requêtes API. En savoir plus		
NOM	TOKEN	DERNIÈRE UTILISATION
Clé publique	pk_test_51KtYRU03HS4vNAiatvnaqAEXLKKX11U0cpkHFLnr9UPI9kZ7AJC0eLkqik61wHFXLmRGHud4aBvp45v82DpskK1300bMfznuIE	28 avr. Cliquez pour copier
Clé secrète	<input type="text" value="Révéler la clé de test"/>	30 avr.

Clé publique

pk_test_51LPiuwJrAgLp47RR0P5cCV0FGec4QaTZMrsjcyECBJbuNdjZGhbhGgcUPu3u51KoXBTUFH2JrOeFqMtrx2bUkx0800QZNcawhc

Clé secrète

sk_test_51LPiuwJrAgLp47RRfBG034RmkceXlk1OVxBft9yEPuwawbG9rC1xQ0fFH49mT5JWRTcBKMEJfPcdAyYAmWdu74ud008LGYOVVH

Back end : nodejs

```
npm i stripe
```

Fichier `routes/payment.route.js`

```
const express=require('express');
const router = express.Router();
const stripe = require ('stripe');
const Stripe =
stripe('sk_test_51LPiuwJrAgLp47RRfBG034RmkceXlk1OVxBft9yEPuwawbG9rC1xQ0fFH49mT5JWRTcBKMEJfPcdAyYAmWdu74ud008LGYOVVH');

router.post('/', async (req, res) => {
  let status, error;
  const { token, amount } = req.body;
  try {
    await Stripe.charges.create({
      source: token.id,
      amount,
      currency: 'usd',
    });
  }
```

```
    status = 'success';  
  } catch (error) {  
    console.log(error);  
    status = 'Failure';  
  }  
  res.json({ error, status });  
});  
  
module.exports = router;
```

Dans **app.js** ajouter

```
const paymentRouter = require( "./routes/payment.route.js")  
  
app.use('/api/payment', paymentRouter);
```

RQ

pour modifier les changement dans vercel, il faut faire un commit dans github
update git :

git add .

git commit -a -m "Commit new version"

git push --set-upstream origin main

Front end : reactjs

```
npm i react-stripe-checkout
```

```
npm i sweetalert2
```

```
npm i sweetalert2-react-content
```

créer le composant StripePayment

Fichier **src/components/articlesRedux/StripePayment.js**

```

import '../App.css';
import StripeCheckout from 'react-stripe-checkout';
import React, { useState } from 'react';
import axios from 'axios';
import Swal from 'sweetalert2';
import withReactContent from 'sweetalert2-react-content';
import { useNavigate, useParams } from "react-router-dom";

const MySwal = withReactContent(Swal);

function StripePayment() {

  let navigate=useNavigate();

  const {total} = useParams();

  const publishableKey =
    'pk_test_51LPiuwJrAgLp47RR0P5cCV0FGec4QaTZMrsjcyECBJbuNdjZGhbhGgcUPu3u51KoXBTUFH2JrOeFqMtrx2bUkx0800QZNcawhc';
  const [product] = useState({
    name: 'Total',
    price: `${total}`,
  });
  const priceForStripe = product.price * 100;

  const handleSuccess = () => {
    MySwal.fire({
      icon: 'success',
      title: 'Payment was successful',
      time: 6000,
    });
    window.location = "/";
  };
  const handleFailure = () => {
    MySwal.fire({
      icon: 'error',
      title: 'Payment was not successful',
      time: 4000,
    });
    navigate('/')
  };
  const payNow = async (token) => { console.log(JSON.stringify(token))
    try {
      const response = await axios({
        //url: 'https://ecommercebackend2023j.vercel.app/api/payment',
        url: 'http://localhost:4000/api/payment',
        method: 'post',
        data: {
          amount: product.price * 100,

```

```

        token,
      },
    ));
    if (response.status === 200) {
      handleSuccess();
    }
  } catch (error) {
    handleFailure();
    console.log(error);
  }
};

return (
  <div className="home-container">
    <div className="products">
      <div className="product">
        <h2>Complete payment </h2>
        <p>
          {product.name}
        </p>
        <p>
          {product.price} TND
        </p>
        <StripeCheckout
          stripeKey={publishableKey}
          label="Pay Now"
          name="Pay With Credit Card"
          billingAddress
          shippingAddress
          amount={priceForStripe}
          description={`Your total is ${product.price} TND`}
          token={payNow}
        />
      </div>
    </div>
  </div>
);
}

export default StripePayment;

```

Fichier `src/App.js`

```
import StripePayment from './components/articlesRedux/StripePayment';
```

.....

```
<Route path='/pay/:total' element={<StripePayment/>}/>
```

- Ajouter le lien au fichier cart.js

```
import React,{useEffect,useCallback} from 'react'
import { useDispatch, useSelector } from "react-redux";
import { Link } from 'react-router-dom';
import {
  addToCart,
  clearCart,
  decreaseCart,
  getTotals,
  removeFromCart,
} from "../../features/cartSlice";

const Cart = () => {
  const cart = useSelector((state) => state.storecart);
  const dispatch = useDispatch();

  const computeTotal = useCallback(() => {
    dispatch(getTotals());
  }, [cart, dispatch])

  useEffect(() => {
    computeTotal()
  }, [computeTotal])

  const handleAddToCart = useCallback((product) => {
    dispatch(addToCart(product));
  }, [dispatch])

  const handleDecreaseCart = useCallback((product) => {
    dispatch(decreaseCart(product));
  }, [dispatch])

  const handleRemoveFromCart = useCallback((product) => {
    dispatch(removeFromCart(product));
  }, [dispatch])

  const handleClearCart = useCallback(() => {
    dispatch(clearCart());
  }, [dispatch])

  return (
    <div className="cart-container">
```

```
<h2>Shopping Cart</h2>
{cart.cartItems.length === 0 ? (
  <div className="cart-empty">
    <p>Panier Vide</p>
    <div className="start-shopping">
      <Link to="/articlescard">

        <span>Start Shopping</span>
      </Link>
    </div>
  </div>
) : (
  <div>
    <div className="titles">
      <h3 className="product-title">Product</h3>
      <h3 className="price">Price</h3>
      <h3 className="quantity">Quantity</h3>
      <h3 className="total">Total</h3>
    </div>
    <div className="cart-items">
      {cart.cartItems &&
        cart.cartItems.map((cartItem) => (
          <div className="cart-item" key={cartItem._id}>
            <div className="cart-product">
              <img src={` ${cartItem.imageart}`}
alt={cartItem.designation}/>
              <div>
                <h3>{cartItem.designation}</h3>
                <p>{cartItem.reference}</p>
                <button onClick={() => handleRemoveFromCart(cartItem)}>
                  Remove
                </button>
              </div>
            </div>
            <div className="cart-product-price"> {cartItem.prix}
TND</div>
            <div className="cart-product-quantity">
              <button onClick={() => handleDecreaseCart(cartItem)}>
                -
              </button>
              <div className="count">{cartItem.cartQuantity}</div>
              <button onClick={() =>
handleAddToCart(cartItem)}>+</button>
            </div>
            <div className="cart-product-total-price">
              {(cartItem.prix * cartItem.cartQuantity).toFixed(3)} TND
            </div>
          </div>
        ))}
    </div>
  </div>
)
```

```

    </div>
    <div className="cart-summary">
      <button className="clear-btn" onClick={() => handleClearCart()}>
        Clear Cart
      </button>
      <div className="cart-checkout">
        <div className="subtotal">
          <span>Subtotal</span>
          <span className="amount">{cart.cartTotalAmount.toFixed(3)}
TND</span>
        </div>
        <p>Taxes and shipping calculated at checkout</p>

        <button onClick={()=>{dispatch(clearCart())}}>
          { <Link to={"/pay/" + cart.cartTotalAmount}
style={{"color":"yellow"}}>
            Validate and Pay
          </Link>
        }
        </button>

        <div className="continue-shopping">
          <Link to="/articlescard">
            <span>Continue Shopping</span>
          </Link>
        </div>
      </div>
    </div>
  </div>
)}
</div>

)
}

export default Cart

```

En cliquant sur

Validate and Pay

Complete payment

Total

350 TND

Pay Now

Pay With Credit Card

Your total is 350 TND

 Adresse e-mail

☒ Même adresse de livraison

 Nom

 Adresse

CP

Ville

Tunisia

Païement

Pay With Credit Card

Your total is 350 TND

✉

hsan.hachicha@gmail.com

☒

Même adresse de livraison

Hassan Hachicha

Av. Kateb Yessine

3003

Sfax

Tunisia

Paiement

Pay With Credit Card

Your total is 350 TND

4242 4242 4242 4242

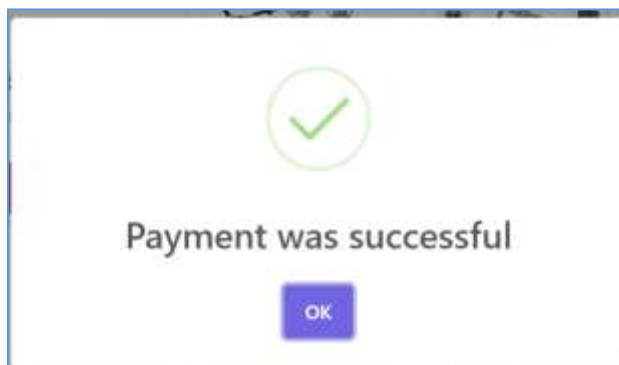
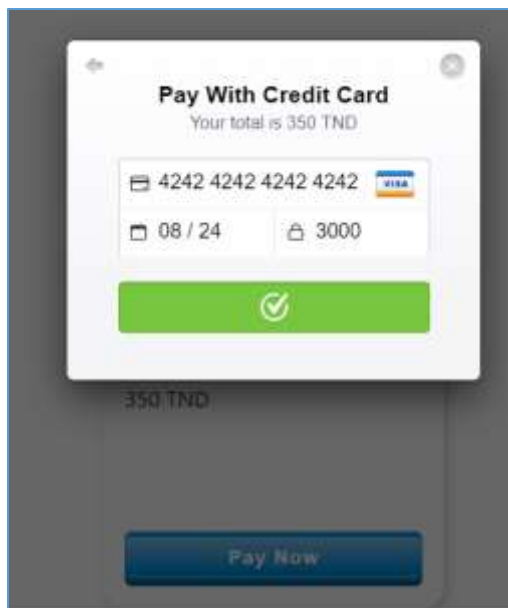
08 / 24

3000

Payer 350,00 \$

350 TND

Pay Now



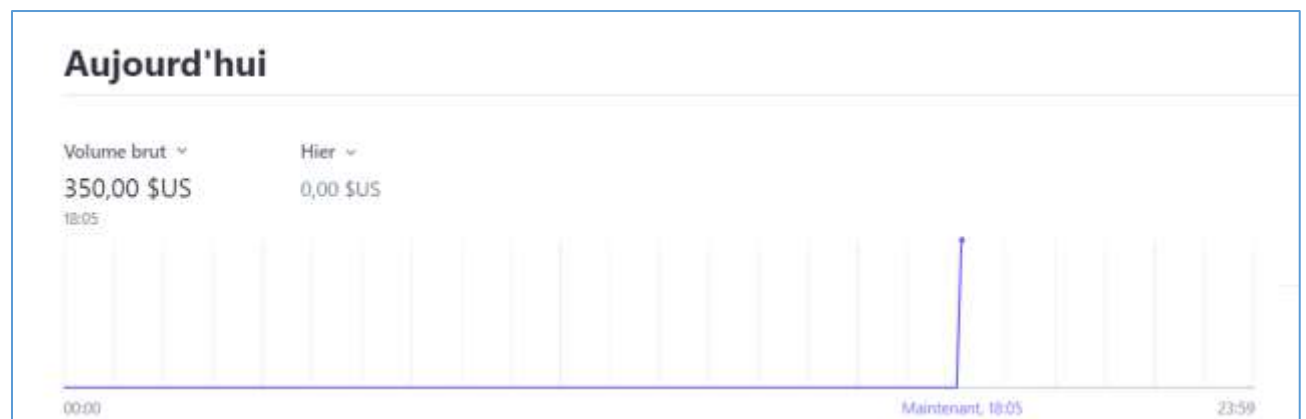
Dans la console du serveur (NodeJS)

```
card: {
  id: 'card_1NUumzD3HS4vNAwaoAYNdFZg',
  object: 'card',
  address_city: 'Sfax',
  address_country: 'Tunisia',
  address_line1: 'Av. Kateb Yessine',
  address_line1_check: 'unchecked',
  address_line2: null,
  address_state: null,
  address_zip: '3003',
  address_zip_check: 'unchecked',
  brand: 'Visa',
  country: 'US',
  cvc_check: 'unchecked',
  dynamic_last4: null,
  exp_month: 8,
  exp_year: 2024,
  funding: 'credit',
  last4: '4242',
  name: 'Hassan Hachicha',
  tokenization_method: null,
  wallet: null
},
client_ip: '41.228.240.103',
created: 1689613249,
email: 'hsan.hachicha@gmail.com',
livemode: false,
type: 'card',
used: false
}
```

Dans le site stripe.com

Aller à l'adresse suivante pour voir le paiement

<https://dashboard.stripe.com/test/dashboard>



<https://dashboard.stripe.com/test/payments>

PAIEMENTS 100%

Exporter

Crier un paiement

Paielements

Tous les paiements Litiges Toutes les transactions

Tous

Réussis

Remboursés

Non capturés

En échec

Date

Montant

État

Moyen de paiement

	MONTANT	DESCRIPTION	CLIENT	DATE	
<input type="checkbox"/>	350,00 \$US	Réussi ✓ c9_382a0c103f54a0f8a61PDS7T3A	Hassan Hachicha	17 juil. à 18:00	...