

Guide de Paiement avec Stripe

Table des matières

1. [Comment fonctionne Stripe](#)
 2. [Architecture du paiement dans l'API](#)
 3. [Passer une commande avec Postman](#)
 4. [Tester le paiement](#)
 5. [Troubleshooting](#)
-

Comment fonctionne Stripe

Concept simple

Stripe est comme un **intermédiaire de confiance** entre ton site et les banques :

```
Client (toi) → API Porsche → Stripe → Banque → Confirmation
```

Les 3 étapes clés

1. Création d'une session de paiement

- Tu demandes à Stripe : "Je veux vendre cet accessoire pour 500€"
- Stripe te donne un **lien de paiement sécurisé**

2. Le client paie

- Le client clique sur le lien
- Il entre ses infos bancaires **directement chez Stripe** (pas sur ton serveur = sécurité ✅)
- Stripe traite le paiement

3. Confirmation (webhook)

- Stripe envoie une notification à ton API : "Paiement réussi !"
 - Ton API met à jour la commande : `statut_paiement: "payé"`
-

Architecture du paiement dans l'API

Fichiers importants

```
controllers/  
├─ payment.controller.js    ← Création session Stripe  
├─ Commande.controller.js   ← Gestion des commandes  
routes/
```

└─ payment.route.js	← Route POST /create-checkout-session
└─ Commande.route.js	← Routes commandes
models/ └─ Commande.model.js	← Schéma MongoDB (statut_paiement, total, etc.)
server.js	← Webhook Stripe POST /webhook

Schéma d'une Commande

```
{
  _id: "64abc...",
  user: "64xyz...",           // ID utilisateur
  lignes_commande: ["64def...", ...], // IDs des lignes de commande
  statut_commande: "en_attente", // ou "confirmée", "expédiée",
  "annulée"
  statut_paiement: "en_attente", // ou "payé", "échoué",
  "remboursé"
  montant_total: 1250.00,      // en euros
  stripe_session_id: "cs_test...", // ID session Stripe
  createdAt: "2025-11-13T10:30:00Z"
}
```

Passer une commande avec Postman

Étape 1 : Se connecter (obtenir le token)

Requête :

```
POST http://localhost:3000/user/login
Content-Type: application/json

Body:
{
  "email": "client@example.com",
  "password": "motdepasse123"
}
```

Réponse :

```
{
  "message": "client@example.com est connecté",
  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
  "user": { "id": "64abc...", "email": "client@example.com" }
}
```

✅ **Copie le token** → Tu en as besoin pour les étapes suivantes

Étape 2 : Créer une commande avec accessoires

Requête :

```
POST http://localhost:3000/commande/new
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
Content-Type: application/json

Body:
{
  "lignes_commande": [
    {
      "accessoire": "64def123...", // ID de l'accessoire (jantes,
spoiler, etc.)
      "quantite": 2,
      "prix_unitaire": 250.00
    },
    {
      "accessoire": "64ghi456...",
      "quantite": 1,
      "prix_unitaire": 750.00
    }
  ],
  "adresse_livraison": {
    "rue": "123 Avenue des Champs",
    "ville": "Paris",
    "code_postal": "75008",
    "pays": "France"
  }
}
```

Réponse :

```
{
  "_id": "6554abc...",
  "user": "64abc...",
  "lignes_commande": ["6554def...", "6554ghi..."],
  "montant_total": 1250.0,
  "statut_commande": "en_attente",
  "statut_paiement": "en_attente",
  "createdAt": "2025-11-13T10:35:00Z"
}
```

✅ **Copie l'ID de la commande** (_id) → Tu en as besoin pour le paiement

Étape 3 : Créer une session de paiement Stripe

Requête :

```
POST http://localhost:3000/api/payment/create-checkout-session
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
Content-Type: application/json

Body:
{
  "commandeId": "6554abc..." // ID de la commande créée à l'étape 2
}
```

Réponse :

```
{
  "url": "https://checkout.stripe.com/c/pay/cs_test_a1b2c3d4e5f6..."
}
```

✅ **Copie l'URL** → C'est le lien de paiement Stripe

Étape 4 : Payer (simulation en mode test)

Avec un navigateur :

1. Colle l'URL Stripe dans ton navigateur
2. Entre ces infos de **carte de test** :
 - Numéro : 4242 4242 4242 4242
 - Date : n'importe quelle date future (ex: 12/26)
 - CVC : n'importe quel 3 chiffres (ex: 123)
 - Nom : n'importe quel nom
3. Clique sur "Payer"

Stripe redirige vers : <http://localhost:3000/success> (ou [/cancel](http://localhost:3000/cancel) si annulé)

Étape 5 : Vérifier que le paiement est validé

Requête :

```
GET http://localhost:3000/commande/6554abc...
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
```

Réponse (après webhook Stripe) :

```
{
  "_id": "6554abc...",
  "statut_commande": "confirmée",      ← Changé !
  "statut_paiement": "payé",          ← Changé !
  "stripe_session_id": "cs_test_...",
  "montant_total": 1250.00
}
```

✅ Commande payée et confirmée 🎉

Tester le paiement

Cartes de test Stripe

Carte	Numéro	Résultat
Succès	4242 4242 4242 4242	✅ Paiement réussi
Refusé	4000 0000 0000 0002	❌ Carte refusée
Insuffisant	4000 0000 0000 9995	❌ Fonds insuffisants
3D Secure	4000 0025 0000 3155	🔒 Authentification requise

Toutes acceptent :

- Date d'expiration : toute date future
- CVC : n'importe quel 3 chiffres
- Code postal : n'importe lequel

Vérifier dans Stripe Dashboard

1. Va sur <https://dashboard.stripe.com/test/payments>
2. Connecte-toi avec ton compte Stripe
3. Tu verras tous les paiements de test

Troubleshooting

❌ Erreur : "No such checkout session"

Problème : Le webhook reçoit un ID session invalide

Solution :

```
# Vérifie que la clé secrète Stripe est configurée
echo $STRIPE_SECRET_KEY

# Si vide, ajoute-la dans .env
STRIPE_SECRET_KEY=sk_test_51...
```

✗ Erreur : "Webhook signature verification failed"

Problème : La signature du webhook est incorrecte

Solution :

```
# Installe Stripe CLI
brew install stripe/stripe-cli/stripe

# Connecte-toi
stripe login

# Redirige les webhooks locaux
stripe listen --forward-to localhost:3000/webhook

# Copie le signing secret affiché et ajoute-le dans .env
STRIPE_WEBHOOK_SECRET=whsec_...
```

✗ La commande reste "en_attente" après paiement

Problème : Le webhook n'a pas été reçu ou traité

Solutions :

1. Vérifie que le serveur tourne : <http://localhost:3000/>
2. Vérifie les logs du serveur pendant le paiement
3. Vérifie dans Stripe Dashboard → Webhooks → Events

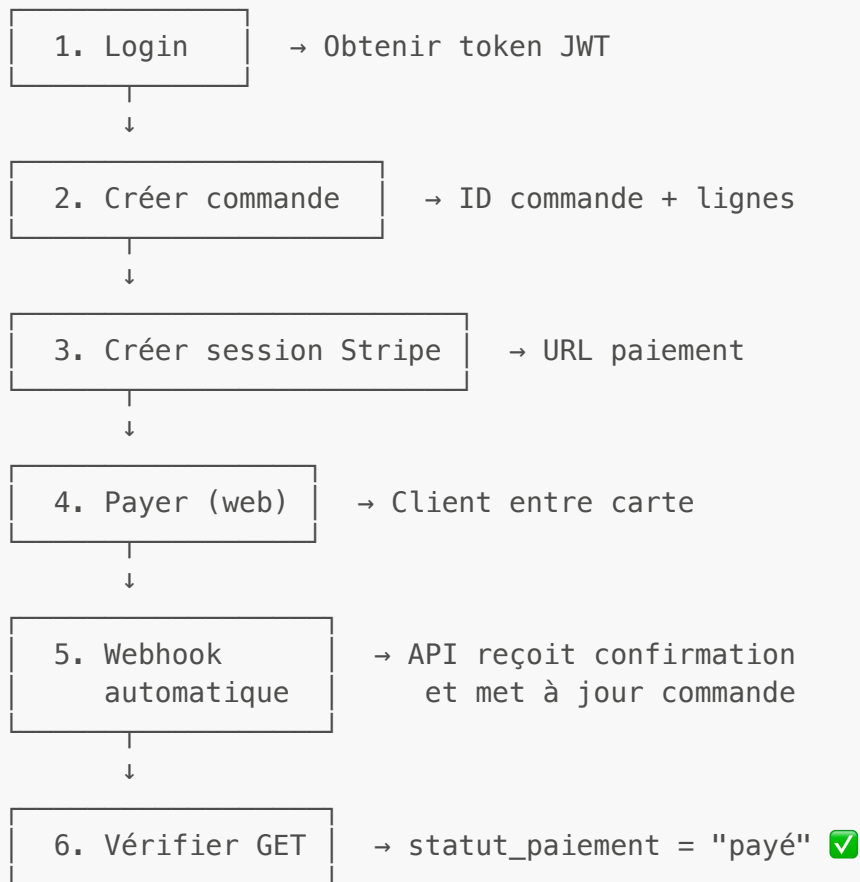
✗ Erreur 401 "Non autorisé"

Problème : Token manquant ou invalide

Solution :

```
Headers de Postman :
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
    ↑ N'oublie pas le mot "Bearer" avant le token !
```

Résumé du flux complet



Variables d'environnement requises

Vérifie ton fichier `.env` :




```
# Stripe (mode test)
STRIPE_SECRET_KEY=sk_test_51xxxxxxxxxxxxx
STRIPE_PUBLISHABLE_KEY=pk_test_51xxxxxxxxxxxxx
STRIPE_WEBHOOK_SECRET=whsec_XXXXXXXXXXXXX

# URLs de redirection
STRIPE_SUCCESS_URL=http://localhost:3000/success
STRIPE_CANCEL_URL=http://localhost:3000/cancel

# JWT
SECRET_KEY=ton_secret_jwt_super_secure

# MongoDB
MONGODB_URI=mongodb://localhost:27017/plateforme_porsche
```

Pour aller plus loin

-  [Documentation Stripe](#)
-  [Stripe Testing](#)
-  [Stripe Checkout Video](#)

Bon courage ! 