

Plateforme de paiement en ligne

Cahier des charges du projet

CONTACT

kmarques@vetixy.com

1 - Pré-requis

1 - Consignes

Nombre de personnes par groupe : **5**

Règles de constitution des groupes : **Libre**

2 - Objectifs

Sujet du projet : **Mise en place d'une plateforme de paiement en ligne**

Type de rendu : **Code + Soutenance**

Date de rendu de projet :

- **Soutenance fin de semestre**
- **Code à rendre 1 semaine avant la soutenance**

Ce qu'il va falloir développer :

- **Api de la plateforme (NodeJS)**
- **Génération du script (page) de paiement par la plateforme (NodeJS)**
- **BackOffice de la plateforme (ReactJS)**
- **Site marchand de tests (Full NodeJS ou NodeJS+React)**
 - **Générer une transaction**
 - **Recevoir la redirection de la page de paiement (page de confirmation de paiement)**
 - **Recevoir la notification de paiement par l'API**
- **Scrapping des taux de change des devises au jour le jour (NodeJS)**
- **Emulateur PSP (NodeJS)**

2 - Cahier des charges

1 - Description du projet

Le but du projet est de développer une plateforme de paiement en ligne.

Le scope du projet s'arrête à la communication avec le PSP (plateforme processant le paiement) et l'ISSUER (banque recevant le paiement). En résumé, les interconnexions à créer sont celles entre l'acheteur / le marchand / la plateforme / un émulateur de PSP.

2 - Processus d'inscription à la plateforme

Un marchand s'enregistre sur la plateforme avec au minimum :

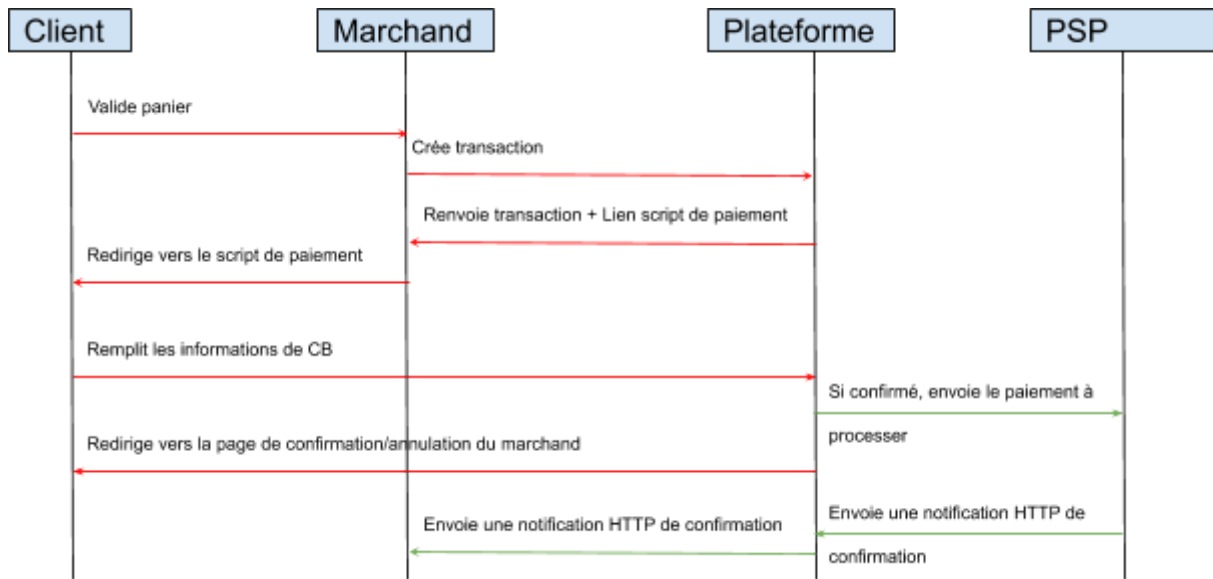
- Un nom de société
- Un KBIS
- Des informations de contact
- Une URL de redirection pour confirmation
- Une URL de redirection pour annulation
- Une devise de reversement

Une fois le marchand inscrit, un email de confirmation lui est envoyé lui spécifiant que son compte est en attente de validation par un Admin.

Une fois le compte validé, un jeu de credentials (client_token & client_secret) est généré par la plateforme et disponible sur l'espace utilisateur du marchand.

Le marchand pourra donc utiliser ses credentials afin d'accéder à l'API.

3 - Processus de paiement



Les **flèches rouges** représentent le workflow principal d'un paiement (transaction), à savoir :

- L'acheteur valide son panier sur le site du marchand
- Le marchand va créer une transaction sur la plateforme de paiement avec les informations de la commande: information client / facturation / livraison / panier utilisateur / montant de la commande avec devise
- La plateforme crée alors la transaction et génère une URL vers un script de paiement
- Le marchand redirige donc l'acheteur vers le script de paiement
- L'acheteur remplit ses informations bancaires sur le script de paiement et confirme ou annule le paiement
- Si l'acheteur confirme, la plateforme lance le **workflow asynchrone** et redirige alors l'acheteur vers la page de confirmation de paiement
- Si l'acheteur annule, la plateforme redirige l'acheteur vers la page d'annulation du site marchand
- Le marchand passe sa transaction respectivement en attente ou en annulation

Les **flèches vertes** représentent le workflow asynchrone d'un paiement (opération), à savoir :

- Si l'acheteur remplit ses informations de paiement, la plateforme envoie la demande de paiement au PSP (infos CB + prix + devise)
- Le PSP envoie une notification HTTP à la plateforme pour confirmer le paiement
- La plateforme va donc notifier en HTTP le marchand que la transaction a été confirmée

Je vous laisse deviner comment gérer les passages d'un intermédiaire à l'autre de la manière la plus sécurisée possible, sachant qu'il faut pouvoir tracer tout le processus à partir de la commande.

Pour simplifier le projet, il existe deux types d'opérations :

- Capture : correspond à l'opération asynchrone déclenchée par le workflow principal (le "paiement")
- Refund : correspond au remboursement déclenché par le BackOffice ou par le site marchand, celui-ci peut être partiel ou total

En cas de remboursement partiel, il peut donc y en avoir plusieurs.

4 - BackOffice de la plateforme

Fonctionnalités Admin :

- Affichage d'un dashboard global de la plateforme (je laisse votre inventivité travailler sur les KPIs / Charts)
- Listing des comptes marchands
 - Validation manuelle du compte
- Listing de toutes les transactions avec une système de recherche de transactions
- Possibilité d'avoir la vue d'un marchand (filtre automatique de l'ensemble du BackOffice par rapport à un marchand sélectionné)

Fonctionnalités Marchand :

- Affichage des transactions du marchand avec une système de recherche de transactions
- Affichage d'un dashboard (je laisse votre inventivité travailler sur les KPIs / Charts)
- Affichage de l'espace marchand
 - informations
 - credentials (possibilité de les supprimer et d'en générer des nouveaux)

Fonctionnalités pour les deux :

- Lors de l'affichage d'une transaction
 - Proposer certaines KPIs par rapport à la transaction
ex : panier moyen de l'acheteur, nombre de transactions remboursées ; utiles pour de la gestion de fraude
 - Pouvoir effectuer un remboursement partiel ou total d'une transaction (création d'une nouvelle opération - workflow vert)
 - Afficher l'historique des différents statuts de la transaction
 - Afficher les opérations d'une transaction ainsi que leur historique, il y a donc plusieurs statuts par opération

5 - API de la plateforme

L'API de la plateforme doit pouvoir être accessible pour le BackOffice ainsi que pour le marchand.

Le marchand doit pouvoir :

- Créer une transaction
- Lister ses transactions
- Afficher une transaction
- Lister les opérations d'une transaction
- Afficher une opération
- Créer des opérations de remboursement (partiel ou intégral)

L'API doit avoir un endpoint permettant de recevoir les notifications de validation d'opérations par le PSP.

6 - Site marchand

Fonctionnalités :

En tant que marchand :

- Page technique permettant de renseigner les credentials générés par le BackOffice
- Page permettant d'effectuer un remboursement d'une commande
- Page permettant de lister toutes les commandes

En tant qu'acheteur :

- Page permettant de renseigner un panier pour soumission
- Page d'annulation de la commande
- Page de confirmation de la commande

Le site doit pouvoir recevoir une notification HTTP de la part de la plateforme quand celle-ci a reçu une validation d'opération par le PSP.

7 - Emulateur PSP

Le PSP reçoit une demande d'opération par la plateforme, attend 10 sec, puis renvoie une confirmation que l'opération s'est bien déroulée à la plateforme.

3 - Spécifications techniques

Plateforme de paiement :

- Le backend doit être codé en NodeJS avec le framework Express comme vu en cours
- Le frontend doit être codé en ReactJS
- La BDD doit être postgresql avec l'ORM Sequelize pour tout ce qui touche à l'aspect relationnel
- La BDD de recherche doit être MongoDB

Site Marchand :

- Le site peut être codé entièrement en NodeJS ou NodeJS+ReactJS

Scrapping + Emulateur PSP :

- Les deux solutions doivent être codées en NodeJS

2 points bonus seront attribués sur des fonctionnalités technologiques/fonctionnelles supplémentaires, par exemple :

- Mise à jour en temps réel du dashboard marchand/listing des transactions sur le BackOffice avec HTTP2 ou HTTP3 (pas de pulling ni websocket)
- Mise en place d'un moteur de fraude sur certains paramètres de la transaction, ex :
 - Récence de paiement par un nom d'acheteur, n° de CB, etc...
 - Pays d'origine de la transaction
 - etc...
- etc...