

# Documentation de la gestion de Paiement

Touil Ilyes

March 6, 2025

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Objectif de l'Application . . . . .	2
1.2	Technologies Utilisées . . . . .	2
1.3	Structure du Projet . . . . .	2
<b>2</b>	<b>Architecture de l'Application</b>	<b>2</b>
<b>3</b>	<b>Fonctionnalités</b>	<b>2</b>
3.1	Paie ment en Ligne . . . . .	2
3.2	Paie ment sur Place . . . . .	2
3.3	Gestion des Paiements . . . . .	3
3.4	Envoi d'Emails de Confirmation . . . . .	3
3.5	Génération de PDF . . . . .	3
<b>4</b>	<b>API Documentation</b>	<b>3</b>
4.1	Endpoints . . . . .	3
4.2	Exemples de Requêtes et Réponses . . . . .	3
<b>5</b>	<b>Front-End</b>	<b>3</b>
5.1	Structure des Composants . . . . .	3
5.2	Services et Modèles . . . . .	4
<b>6</b>	<b>Back-End</b>	<b>4</b>
6.1	Services Spring Boot . . . . .	4
6.2	Modèles de Données . . . . .	4
<b>7</b>	<b>Conclusion</b>	<b>4</b>

# 1 Introduction

## 1.1 Objectif de l'Application

L'application de paiement permet aux utilisateurs de réaliser des paiements en ligne ou sur place, de gérer leurs transactions, et de recevoir des confirmations par email.

## 1.2 Technologies Utilisées

- Front-End : Angular, TypeScript, HTML, SCSS
- Back-End : Spring Boot, Java
- Base de Données : MySQL
- Autres : EmailJS pour l'envoi d'emails, jsPDF pour la génération de PDF

## 1.3 Structure du Projet

Le projet est divisé en deux parties principales :

- **Front-End** : Géré par Angular, avec des composants pour les formulaires de paiement et la gestion des transactions.
- **Back-End** : Géré par Spring Boot, avec des endpoints pour les opérations CRUD sur les paiements.

# 2 Architecture de l'Application

- **Front-End** : Composants Angular pour l'interface utilisateur.
- **Back-End** : Services Spring Boot pour la logique métier.
- **Base de Données** : Base H2 pour stocker les informations sur les paiements.

# 3 Fonctionnalités

## 3.1 Paiement en Ligne

Les utilisateurs peuvent effectuer des paiements en ligne en saisissant les informations de leur carte.

## 3.2 Paiement sur Place

Les utilisateurs peuvent prendre un rendez-vous pour effectuer un paiement dans une agence.

### 3.3 Gestion des Paiements

Les administrateurs peuvent consulter, mettre à jour ou supprimer des paiements.

### 3.4 Envoi d’Emails de Confirmation

Un email de confirmation est envoyé après chaque paiement réussi.

### 3.5 Génération de PDF

Un PDF de confirmation est généré et téléchargé après chaque paiement.

## 4 API Documentation

### 4.1 Endpoints

- **GET /paiements** : Récupère tous les paiements.
- **POST /paiements/enligne** : Ajoute un paiement en ligne.
- **POST /paiements/surplace** : Ajoute un paiement sur place.
- **DELETE /paiements/id** : Supprime un paiement.

### 4.2 Exemples de Requêtes et Réponses

Listing 1: Exemple de requête POST pour un paiement en ligne

```
{
  "montant": 100,
  "date_paiement": "2023-10-01",
  "numeroCarte": "4111111111111111",
  "expiration": "12/25",
  "cvv": "123"
}
```

## 5 Front-End

### 5.1 Structure des Composants

- **PaiementfrontComponent** : Gère les formulaires de paiement.
- **PaiementService** : Interagit avec l’API back-end.

## 5.2 Services et Modèles

- **PaielementService** : Contient les méthodes pour interagir avec l'API.
- **Paielement**, **PaielementEnLigne**, **PaielementSurPlace** : Modèles de données.

## 6 Back-End

### 6.1 Services Spring Boot

Les services Spring Boot gèrent la logique métier et interagissent avec la base de données.

### 6.2 Modèles de Données

- **Paielement** : Classe de base pour les paiements.
- **PaielementEnLigne** : Classe pour les paiements en ligne.
- **PaielementSurPlace** : Classe pour les paiements sur place.

## 7 Conclusion

L'application de paiement offre une solution complète pour gérer les transactions en ligne et sur place. Elle est facile à utiliser et extensible pour de futures améliorations dont l'intégration avec les autres gestions.