

# SP-Clients

Application Mobile E-commerce – React Native / API Node.js

---

Séraphin PARYS – Documentation Technique  
Version 1.0 – 2025

## I. Informations générales

---

Nom du projet : SP-Clients

Description : Application mobile cross-platform (Android & iOS) permettant aux clients de la société Séraphin PARYS de consulter le catalogue produits, passer des commandes et effectuer un paiement en ligne ou en boutique.

Objectif principal : Ouvrir un nouveau canal de vente e-commerce et fidéliser la clientèle en dehors des points de vente physiques.

Contexte / Historique : Développement initié en 2025 dans le cadre de la phase 1 du projet numérique de SP.

Version	Date	Rédacteur	Description
V1.0	2025	armand	Version initiale

## II. Architecture du système

---

### Technologies utilisées

- Mobile : React Native (JavaScript, cross-platform Android & iOS)
- Back-end / API : Node.js (API REST)
- Base de données : partagée avec SP-Vendeurs (MySQL 8.0)
- Infrastructure : VPS Debian 11.11 hébergé chez Magic Online

## III. Installation et déploiement

---

### Prérequis

- Node.js 20.x LTS
- npm 10.x ou yarn
- React Native CLI ou Expo
- Android Studio (émulateur Android) / Xcode (iOS)

### Installation locale – API Node.js

```
git clone https://github.com/<votre-repo>/sp-clients-api
cd sp-clients-api
npm install
cp .env.example .env # configurer les variables d'environnement
node index.js # ou : npx nodemon index.js
```

### Installation locale – Application React Native

```
git clone https://github.com/<votre-repo>/sp-clients-app
cd sp-clients-app
npm install
```

```
npx react-native start
eas build --platform android --profile production
```

## IV. Configuration

---

- Fichier de configuration API : .env (ne pas versionner)

### Variables d'environnement (API)

```
DB_HOST=localhost
DB_USER=<utilisateur>
DB_PASSWORD=<mot_de_passe>
DB_NAME=sp_parys
JWT_SECRET=<clé_secrète>
PORT=3000
```

## V. Base de données

---

### Informations générales

- Type et version : MySQL 8.0 (partagée avec SP-Vendeurs)
- L'API Node.js accède en lecture au catalogue et en écriture aux commandes et comptes clients

### Données sensibles

- Mots de passe clients : hachage bcrypt
- Tokens JWT : durée de validité à définir
- Conformité RGPD : données personnelles clients protégées et anonymisables

## VI. Code et structure du projet

---

### API Node.js

```
|--- config/           # Configuration Swagger, BDD
|--- routes/          # Définition des routes Express
|--- controllers/     # Logique métier
|--- models/          # Accès BDD
|--- middlewares/     # Auth JWT, validation
|--- index.js         # Point d'entrée
|--- swagger.js       # Configuration Swagger UI
|--- .env              # Variables d'environnement (non versionné)
```

### Application React Native

### Documentation API – Swagger

- Swagger UI accessible en local : <http://localhost:3000/docs>
- Spécification JSON : <http://localhost:3000/api-docs.json>

## VII. Sécurité et authentification

---

- Authentification client : JWT (JSON Web Token)
- HTTPS obligatoire en production
- Activation / désactivation des comptes clients depuis l'Intranet SP-Vendeurs

### Rôles et permissions

Fonctionnalité	Client authentifié	Non authentifié
Consulter le catalogue	✓	✓
Passer une commande	✓	X
Accéder à l'historique	✓	X
Gérer son profil	✓	X

## VIII. Tests et qualité

---

- Tests API : Swagger UI (Try it out) ou Postman

## IX. Intégrations externes

---

- Paiement en ligne : à définir (ex. Stripe)
- Catalogue produits : synchronisation avec la BDD partagée SP-Vendeurs / Application Bureau

## X. Supervision et maintenance

---

- Disponibilité requise : 24h/24, 7j/7 (exigence cahier des charges)

## XI. Historique et roadmap

---

- V1.0 (2025) : application mobile e-commerce SP-Clients
- Limitations connues : paiement en ligne à intégrer, pas de multi-langues

## XII. Annexes

---

- Cahier des charges Séraphin PARYS (D&H Parys, 10/2024)
- Guide de documentation technique (T. Vosgiens, 08/2025)
- Documentation Swagger : <http://localhost:3000/docs>