EventHub

Plateforme de Gestion d'Événements

Documentation Technique

Schéma de Base de Données
Diagramme de Classes
Diagrammes de Séquence

Module : Sécurité Web
Filière : DATA
Année : 2025-2026

Institut National des Postes et Télécommunications
Rabat, Maroc

Janvier 2026

Table des matières

1 Introduction

Ce document présente la documentation technique de l'application **EventHub**, une plateforme complète de gestion d'événements développée avec une architecture MERN (MongoDB, Express.js, React, Node.js).

1.1 Architecture Générale

L'application suit une architecture trois tiers :
— **Frontend** : Application React avec Vite, TailwindCSS
— **Backend** : API REST Node.js avec Express.js
— **Base de données** : MongoDB avec Mongoose ODM

1.2 Technologies Utilisées

Couche	Technologie	Version
Frontend	React	18.2.0
Frontend	Vite	5.0.10
Frontend	TailwindCSS	3.4.0
Frontend	React Router	6.21.1
Backend	Node.js	18+
Backend	Express.js	4.18.2
Backend	Mongoose	8.0.3
Base de données	MongoDB	6.0+
Authentification	JWT	jsonwebtoken 9.0.2

TABLE 1 – Stack technologique

2 Schéma de Base de Données

2.1 Modèle Entité-Relation

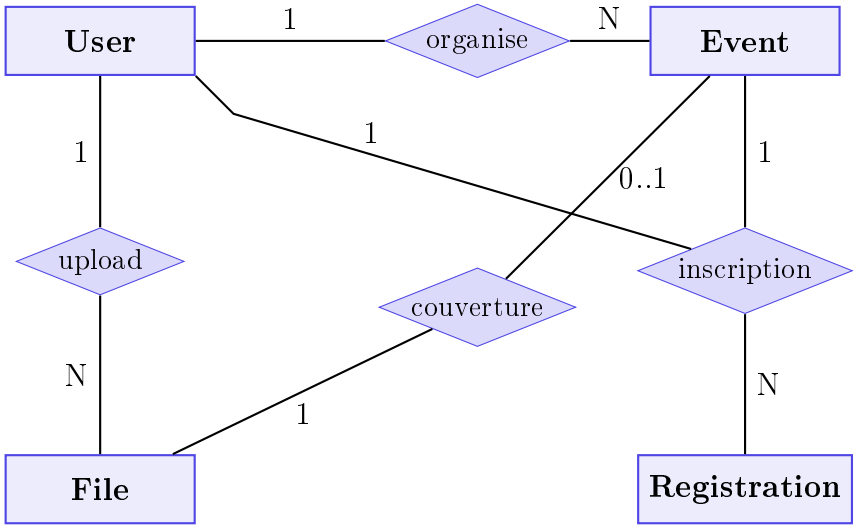

FIGURE 1 – Diagramme Entité-Relation simplifié

2.2 Collections MongoDB

2.2.1 Collection : Users

Champ	Type	Contraintes	Description
_id	ObjectId	PK, Auto	Identifiant unique
email	String	Unique, Required	Adresse email
password	String	Required, Min 8	Mot de passe hashé (bcrypt)
firstName	String	Required, Max 50	Prénom
lastName	String	Required, Max 50	Nom de famille
role	String	Enum : user, admin	Rôle utilisateur
avatar	ObjectId	Ref : File	Photo de profil
refreshToken	String	Select : false	Token de rafraîchissement
isActive	Boolean	Default : true	Statut du compte
createdAt	Date	Auto	Date de création
updatedAt	Date	Auto	Date de modification

TABLE 2 – Structure de la collection Users

2.2.2 Collection : Events

Champ	Type	Contraintes	Description
_id	ObjectId	PK, Auto	Identifiant unique
title	String	Required, Max 200	Titre de l'événement
description	String	Required, Max 5000	Description détaillée
category	String	Enum (6 valeurs)	Catégorie
status	String	Enum (4 valeurs)	État de publication
startDate	Date	Required	Date de début
endDate	Date	Required, > startDate	Date de fin
location.address	String	Required, Max 300	Adresse
location.city	String	Required, Max 100	Ville
location.postalCode	String	Max 20	Code postal
location.country	String	Default : France	Pays
capacity	Number	Required, 1-100000	Capacité maximale
price	Number	Min : 0, Default : 0	Prix en MAD
coverImage	ObjectId	Ref : File	Image de couverture
organizer	ObjectId	Ref : User, Required	Organisateur
registrationCount	Number	Default : 0	Nombre d'inscrits
tags	[String]	Max 10 items	Tags/mots-clés

TABLE 3 – Structure de la collection Events

Valeurs Enum pour category : conference, workshop, concert, sport, networking, other

Valeurs Enum pour status : draft, published, cancelled, completed

2.2.3 Collection : Registrations

Champ	Type	Contraintes	Description
_id	ObjectId	PK, Auto	Identifiant unique
user	ObjectId	Ref : User, Required	Utilisateur inscrit
event	ObjectId	Ref : Event, Required	Événement concerné
status	String	Enum (3 valeurs)	État de l'inscription
registeredAt	Date	Default : now	Date d'inscription
createdAt	Date	Auto	Date de création
updatedAt	Date	Auto	Date de modification

TABLE 4 – Structure de la collection Registrations

Index unique : (user, event) - Empêche les inscriptions multiples

2.2.4 Collection : Files

Champ	Type	Contraintes	Description
_id	ObjectId	PK, Auto	Identifiant unique
originalName	String	Required	Nom original du fichier
filename	String	Unique, Required	Nom stocké (UUID)
mimeType	String	Required	Type MIME
size	Number	Required	Taille en octets
path	String	Required	Chemin de stockage
uploadedBy	ObjectId	Ref : User, Required	Utilisateur uploader
associatedWith.model	String	Enum : Event, User	Type d'entité liée
associatedWith.id	ObjectId	Required	ID de l'entité liée
isPublic	Boolean	Default : false	Visibilité publique

TABLE 5 – Structure de la collection Files

2.3 Index de Base de Données

Collection	Index	Type
Users	email	Unique
Users	role	Simple
Users	createdAt	Descendant
Events	title, description	Text
Events	category	Simple
Events	status	Simple
Events	startDate	Simple
Events	organizer	Simple
Events	location.city	Simple
Registrations	(user, event)	Unique Composé
Registrations	event	Simple
Registrations	user	Simple
Files	uploadedBy	Simple
Files	(associatedWith.model, associatedWith.id)	Composé

TABLE 6 – Index de la base de données

3 Diagramme de Classes

3.1 Architecture Backend

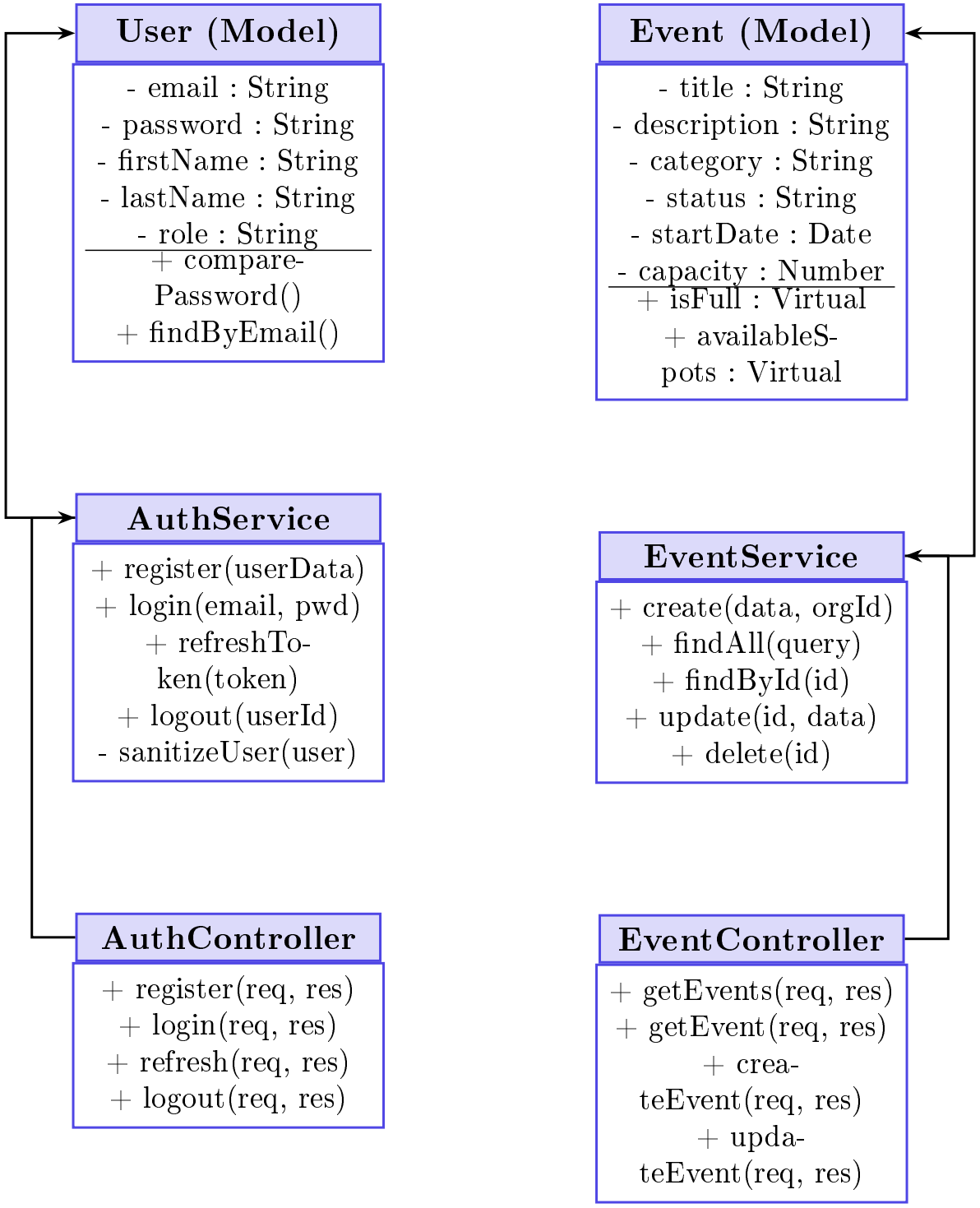

FIGURE 2 – Diagramme de classes simplifié - Architecture MVC

3.2 Middlewares

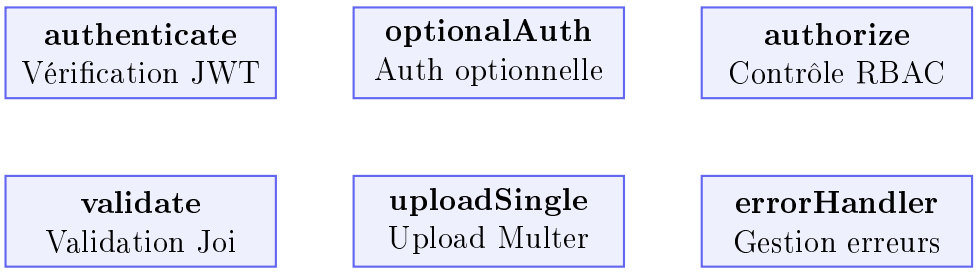

FIGURE 3 – Middlewares de l'application

4 Diagrammes de Séquence

4.1 Authentification - Inscription

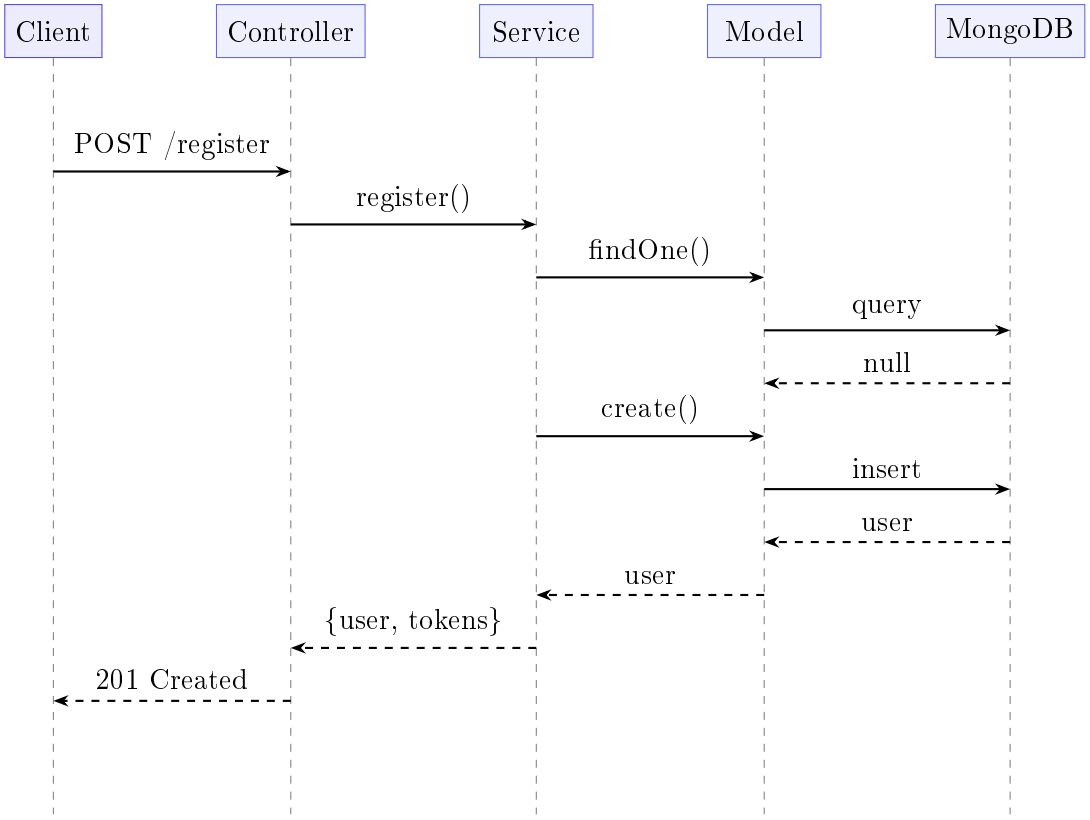

FIGURE 4 – Diagramme de séquence - Inscription utilisateur

4.2 Authentification - Connexion

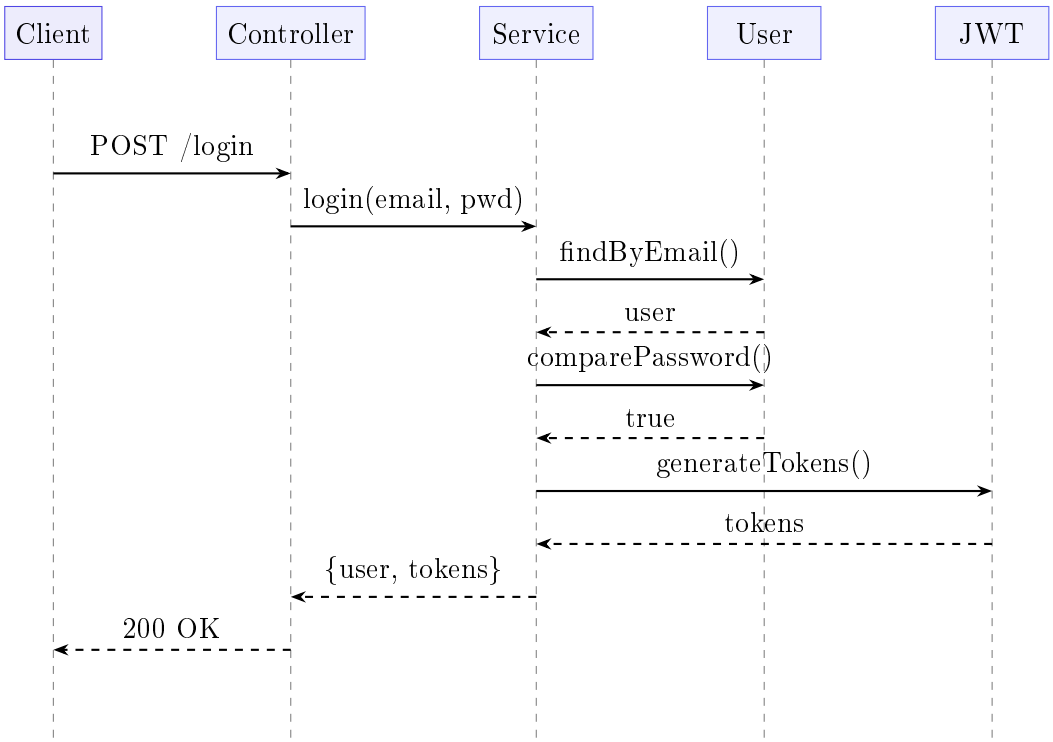

FIGURE 5 – Diagramme de séquence - Connexion utilisateur

4.3 Gestion des Événements - Création

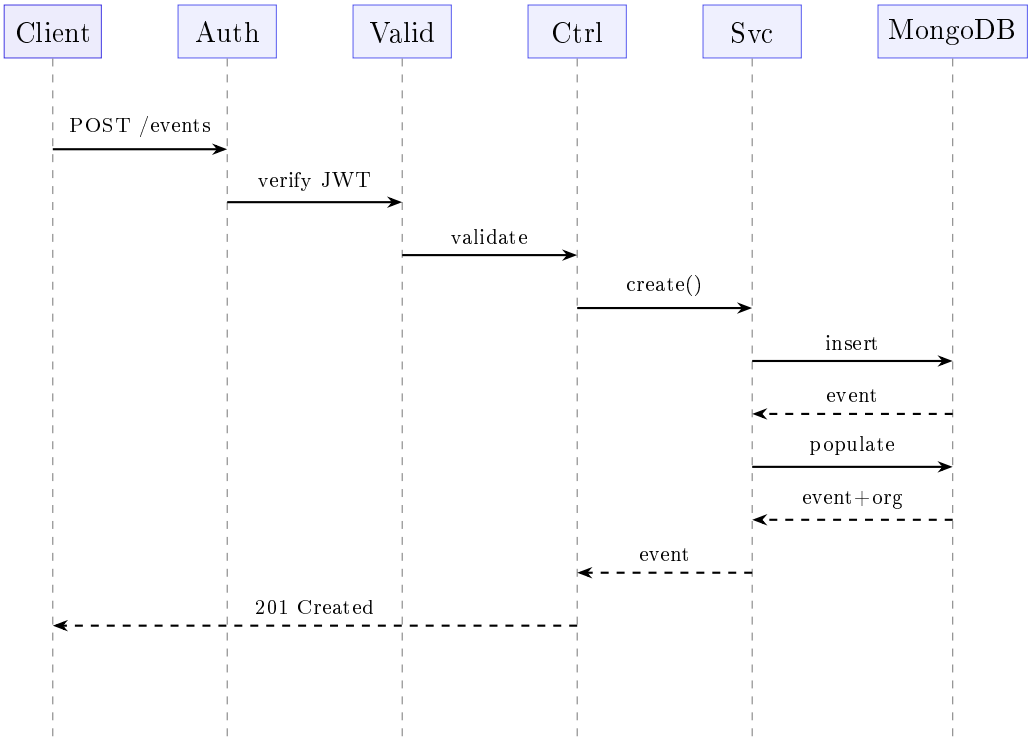

FIGURE 6 – Diagramme de séquence - Création d'événement

4.4 Inscription à un Événement

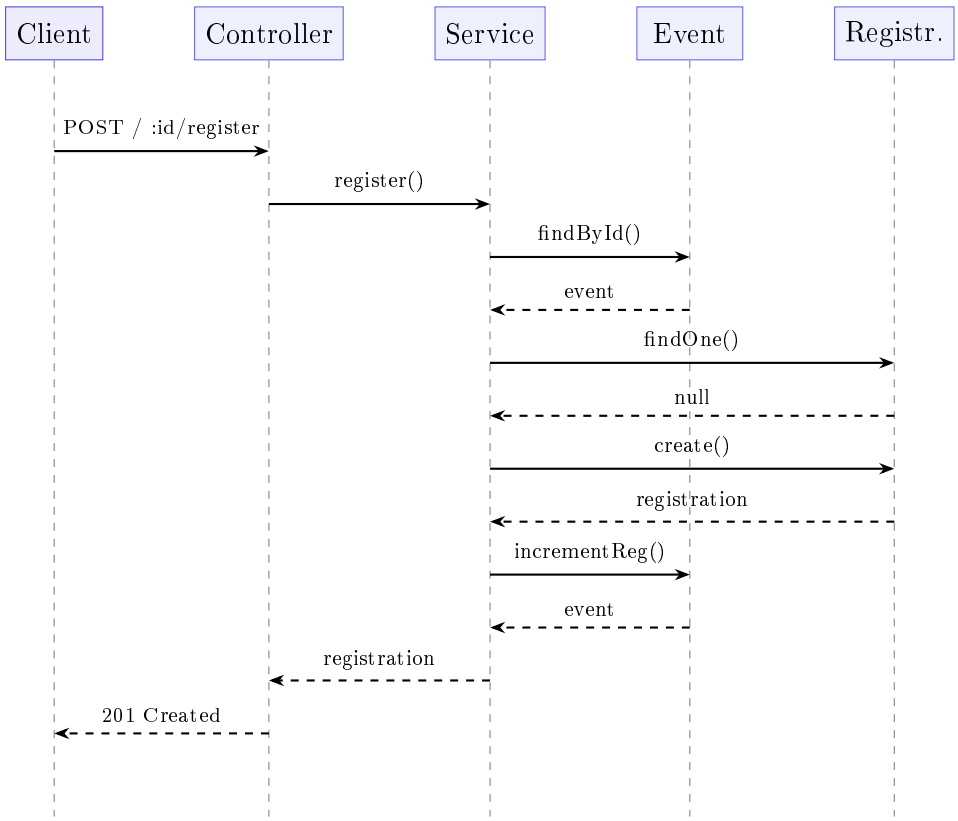

FIGURE 7 – Diagramme de séquence - Inscription à un événement

5 Sécurité

5.1 Mécanismes d'Authentification

Mécanisme	Implémentation
Hashage mot de passe	bcrypt avec salt (12 rounds)
Token d'accès	JWT signé (HS256), expire en 15min
Token de rafraîchissement	JWT signé, expire en 7 jours
Stockage refresh token	Base de données (champ user)

Table 7 – Mécanismes d'authentification

5.2 Contrôle d'Accès (RBAC)

Action	User	Admin
Voir événements publiés	✓	✓
Créer un événement	✓	✓
Modifier ses événements	✓	✓
Modifier tous les événements		✓
S'inscrire à un événement	✓	✓
Voir toutes les statistiques		✓
Accéder aux stats admin		✓

Table 8 – Matrice des permissions

5.3 Autres Mesures de Sécurité

— **Helmet** : Headers HTTP sécurisés
— **CORS** : Configuration restrictive avec origine autorisée
— **Rate Limiting** : 100 requêtes / 15 minutes par IP
— **Validation** : Joi pour toutes les entrées utilisateur
— **Upload** : Validation MIME type, limite de taille (5MB)

6 Annexes

6.1 Variables d'Environnement

```
# Server
PORT=5000
NODE_ENV=development

# Database
MONGODB_URI=mongodb://localhost:27017/eventhub

# JWT
JWT_SECRET=your-secret-key
JWT_EXPIRES_IN=15m
JWT_REFRESH_SECRET=your-refresh-secret
JWT_REFRESH_EXPIRES_IN=7d

# Frontend
FRONTEND_URL=http://localhost:5173

# Upload
UPLOAD_PATH=uploads
MAX_FILE_SIZE=5242880
```

6.2 Scripts NPM

```
# Backend
npm run dev           # Developpement avec nodemon
npm start             # Production
npm run seed          # Peupler la base de donnees

# Frontend
npm run dev           # Serveur de developpement Vite
npm run build         # Build de production
npm run preview       # Preview du build
```