

The background features a glowing, 3D-style 'API' text in the center, surrounded by a complex, circuit-like pattern of lines and dots in shades of blue and purple. The overall aesthetic is high-tech and digital.

# CAHIER DES CHARGES TECHNIQUE - API

# 1. Description du projet

backend développée en Node.js/Express qui gère un système d'événements avec authentification utilisateur et système de favoris.

- Fournir une API RESTful pour la gestion d'événements
- Gérer l'authentification et l'autorisation des utilisateurs
- Permettre la création, lecture, modification et suppression d'événements
- Implémenter un système de favoris pour les utilisateurs
- Intégrer des événements externes depuis l'API officielle "Que Faire à Paris"

## 2. ARCHITECTURE TECHNIQUE



### Node.js

Environnement d'exécution JavaScript  
côté serveur



### Framework

Express.



### Base de données

SQLite



### Authentication

JWT



### Tests

Jest

# 3. SPÉCIFICATIONS FONCTIONNELLES

## Gestion des utilisateurs

- Modèle utilisateur
- Endpoints d'authentification

## Gestion des événements

- Modèle événement
- Fonctionnalités événement

## Fonctionnalités avancées

- Système de favoris
- Récupération avec filtres (catégorie, utilisateur, recherche textuelle)
- Intégration d'événements externes depuis l'API Paris Open Data
- Géolocalisation (latitude/longitude)

# 4. SPÉCIFICATIONS TECHNIQUES

## Sécurité



### Authentication JWT

- Token expirant
- Middleware de protection des routes
- Hachage des mots de passe avec bcrypt



### Validation des données

- Validation email regex : `/^[^\\s@\\]+@[\\^\\s@\\]+\\.\\.[^\\s@\\]+$/`
- Mot de passe minimum 6 caractères
- Nom minimum 2 caractères
- Sanitisation des entrées

# Intégration API externe

## API "Que Faire à Paris" - Open Data Paris

- **URL de base** : <https://opendata.paris.fr/explore/dataset/que-faire-a-paris-/api/>
- **Description** : Agenda participatif officiel de la Ville de Paris
- **Données** : Événements et activités de Paris et sa région
- **Format** : JSON via API REST
- Filtres disponibles

# Base de données et Gestion des erreurs

## SQLite

- Base locale pour développement
- Chemin configurable via DB\_PATH
- Initialisation automatique des tables
- Contraintes d'intégrité référentielle

## Middleware global

### Validation des contraintes

- Email unique
- Relations cohérentes
- Gestion des doublons dans favoris

# 5. CONFIGURATION ET DÉPLOIEMENT

## Installation

Configuration des dépendances et de l'environnement

## Configuration

Paramétrage des variables d'environnement

## Déploiement

Mise en production de l'API



# Endpoint de santé

GET /health

Vérification du statut du serveur

## 6. TESTS (avec jest)

- Tests unitaires sur modèles et contrôleurs
- Tests d'intégration sur les endpoints