

EventHub

Plateforme de Gestion d'Événements

Rapport d'Utilisation

Guide d'Installation

Documentation API

Manuel Utilisateur

Module : Sécurité Web

Filière : DATA

Année : 2025-2026

Institut National des Postes et Télécommunications
Rabat, Maroc

Janvier 2026

Table des matières

1 Introduction

EventHub est une application web complète de gestion d'événements permettant aux utilisateurs de créer, gérer et participer à des événements. L'application est construite avec une architecture moderne MERN stack.

1.1 Fonctionnalités Principales

- **Authentification sécurisée** : Inscription, connexion avec JWT
- **Gestion des événements** : CRUD complet avec filtres avancés
- **Système d'inscription** : Inscription/désinscription aux événements
- **Upload de fichiers** : Images de couverture et avatars
- **Tableau de bord** : Statistiques et graphiques
- **Rôles utilisateurs** : User et Admin avec permissions différenciées

2 Installation et Configuration

2.1 Prérequis

| Logiciel | Version minimale | Téléchargement |
|----------|------------------|---|
| Node.js | 18.0.0 | https://nodejs.org |
| MongoDB | 6.0 | https://www.mongodb.com |
| npm | 9.0.0 | Inclus avec Node.js |
| Git | 2.40 | https://git-scm.com |

TABLE 1 – Prérequis système

2.2 Installation du Backend

1. Naviguer vers le dossier backend

```
cd eventhub/backend
```

2. Installer les dépendances

```
npm install
```

3. Créer le fichier .env

```
# Copier le fichier exemple
cp .env.example .env

# Editer avec vos valeurs
nano .env
```

4. Contenu du fichier .env

```
PORT=5000
NODE_ENV=development
MONGODB_URI=mongodb://localhost:27017/eventhub
JWT_SECRET=votre-cle-secrete-jwt-256-bits
JWT_EXPIRES_IN=15m
JWT_REFRESH_SECRET=votre-cle-refresh-secrete
JWT_REFRESH_EXPIRES_IN=7d
FRONTEND_URL=http://localhost:5173
UPLOAD_PATH=uploads
MAX_FILE_SIZE=5242880
```

5. Lancer le serveur de développement

```
npm run dev
```

Note

Le serveur backend sera accessible sur <http://localhost:5000>

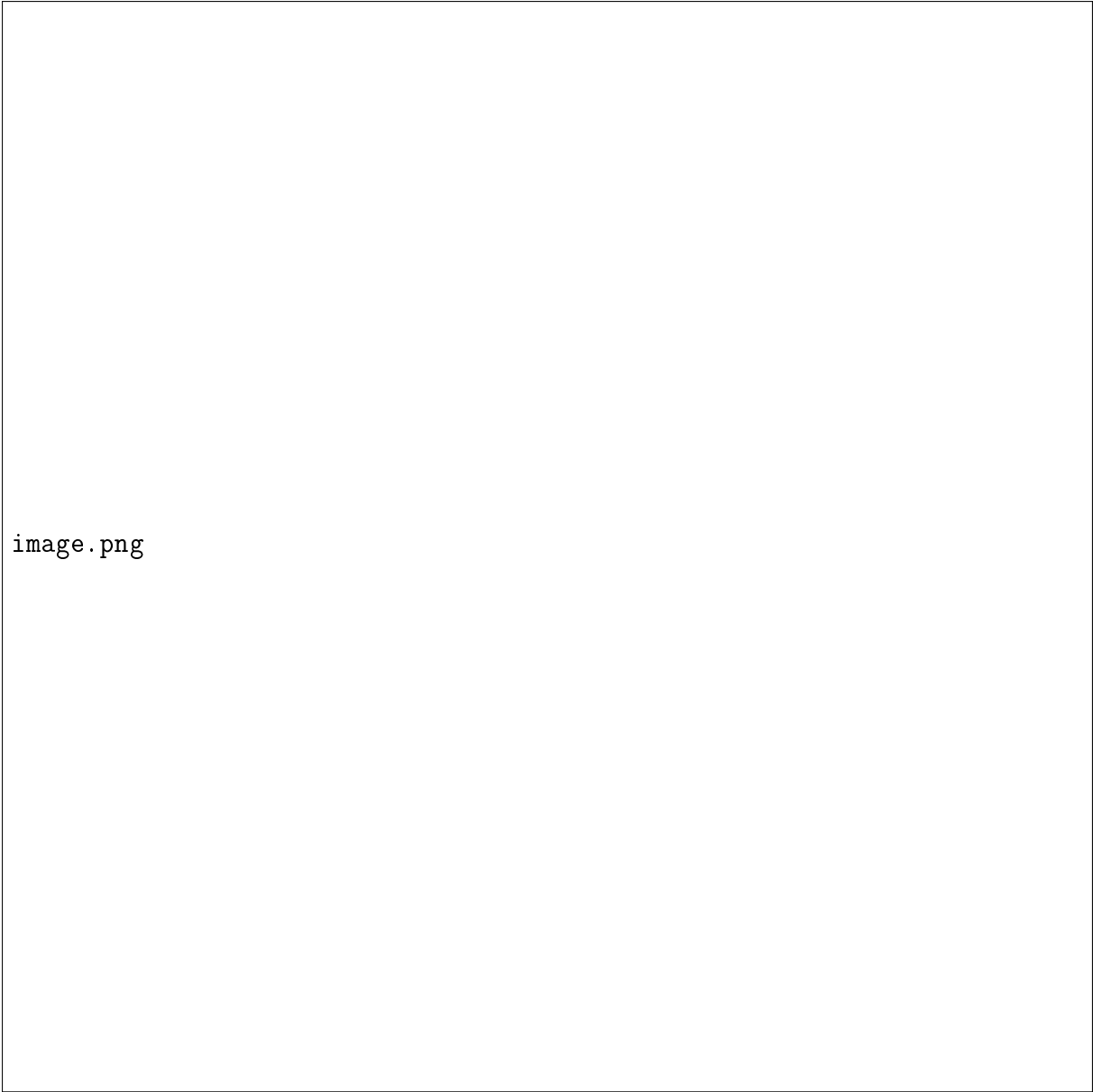


image.png

2.3 Installation du Frontend

1. Naviguer vers le dossier frontend

```
cd eventhub/frontend
```

2. Installer les dépendances

```
npm install
```

3. Lancer le serveur de développement

```
npm run dev
```

Note

L'application frontend sera accessible sur <http://localhost:5173>



1.png

2.4 Initialisation de la Base de Données

Pour peupler la base de données avec des données de test :

```
cd backend  
npm run seed
```

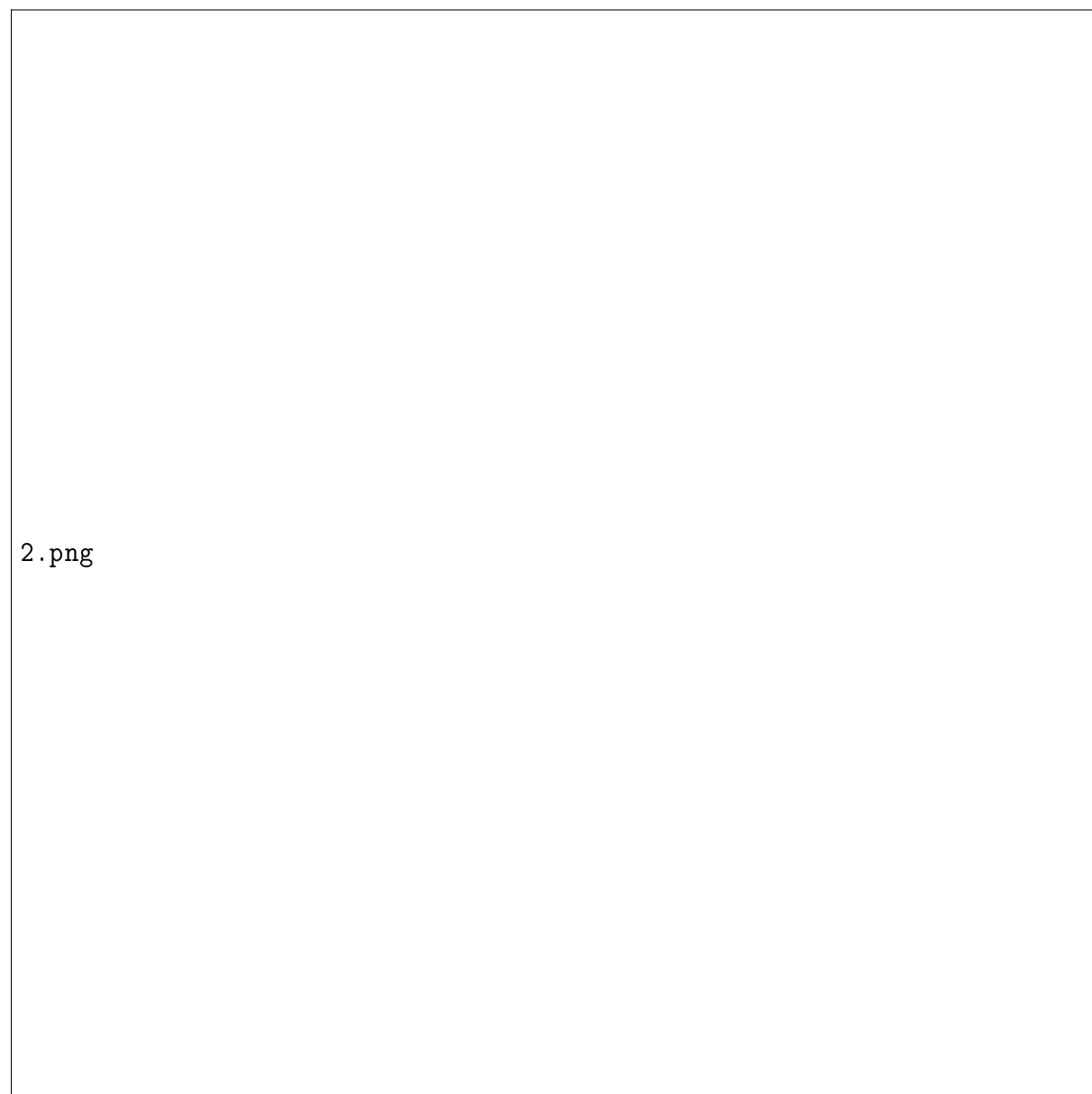
Comptes de test créés

Admin : admin@eventhub.ma / Admin123 !
User 1 : user@eventhub.ma / User123 !
User 2 : fatima@eventhub.ma / Fatima123 !
User 3 : omar@eventhub.ma / Omar123 !

3 Guide Utilisateur

3.1 Inscription et Connexion

3.1.1 Page d'Inscription



2.png

FIGURE 1 – Formulaire d'inscription

Champs requis :

- Prénom (2-50 caractères)
- Nom (2-50 caractères)
- Email (format valide)
- Mot de passe (min 8 caractères, 1 majuscule, 1 minuscule, 1 chiffre)
- Confirmation du mot de passe

3.1.2 Page de Connexion

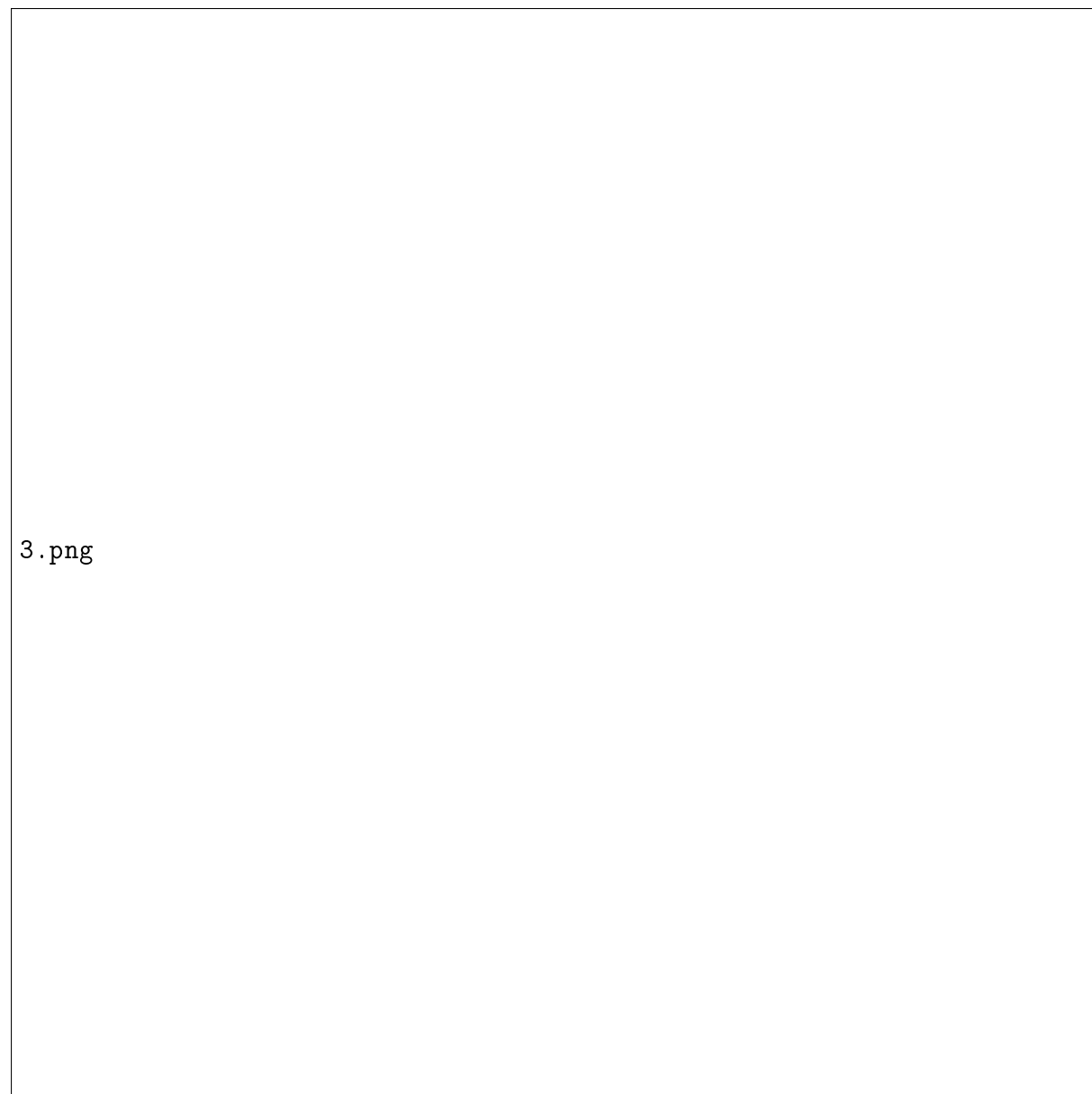


FIGURE 2 – Formulaire de connexion

3.2 Tableau de Bord

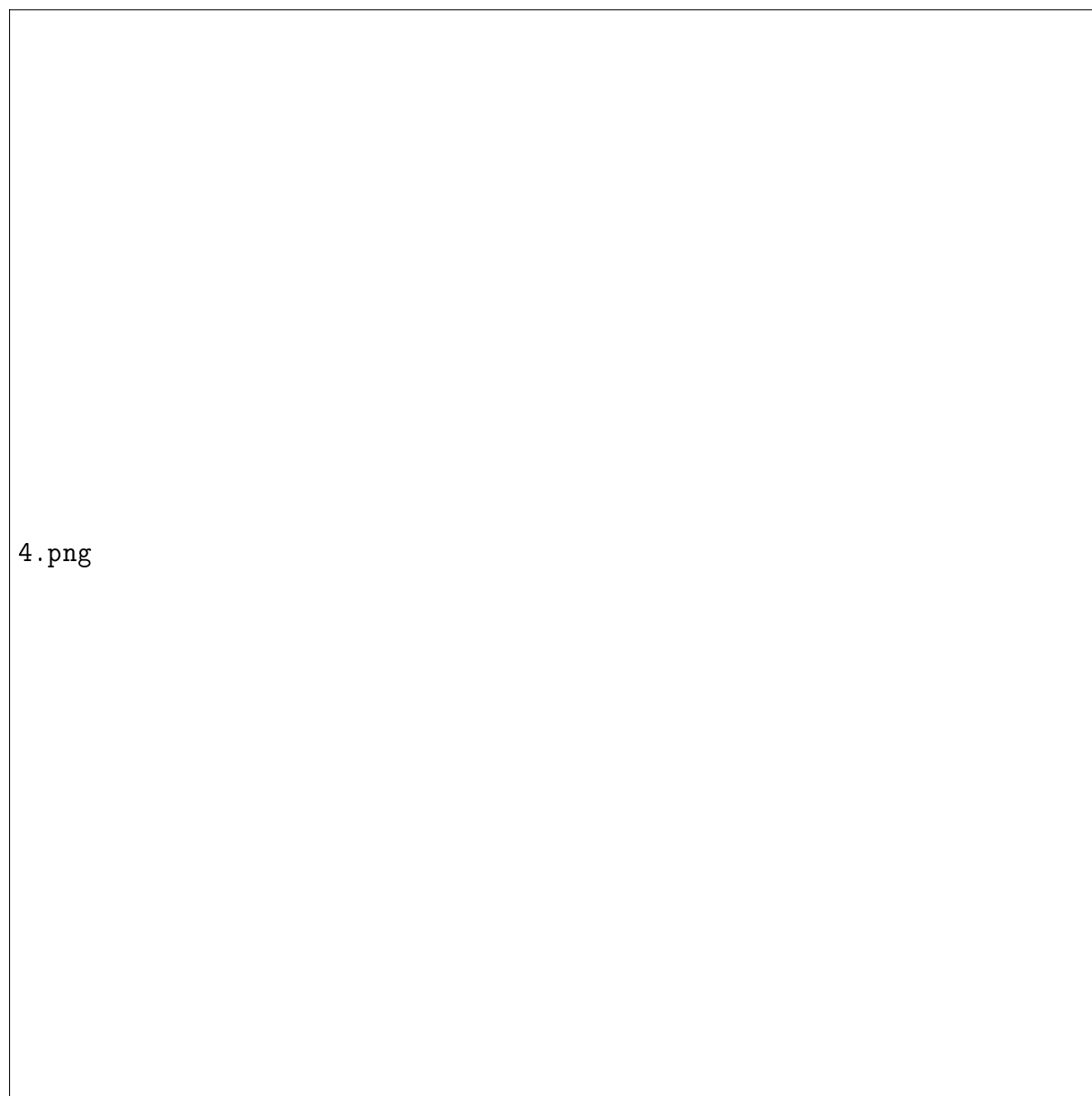


FIGURE 3 – Tableau de bord utilisateur

Le tableau de bord affiche :

- Statistiques générales (événements, inscriptions)
- Graphique de tendance des inscriptions
- Répartition par catégorie
- Événements à venir
- Top événements populaires

3.3 Gestion des Événements

3.3.1 Liste des Événements

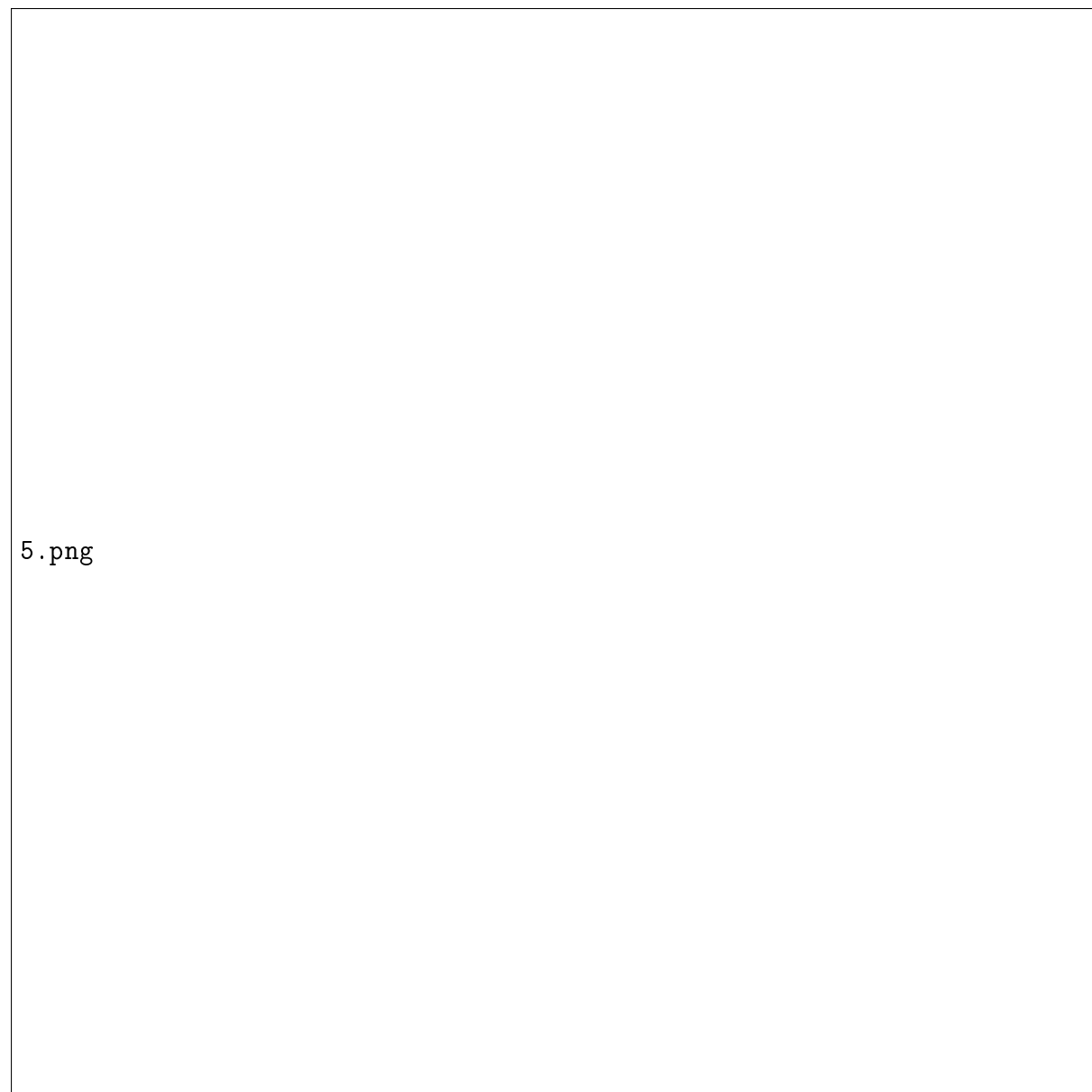


FIGURE 4 – Liste des événements avec filtres

Filtres disponibles :

- Recherche textuelle (titre, description, ville)
- Catégorie (Conférence, Atelier, Concert, Sport, Networking, Autre)
- Ville
- Plage de dates
- Plage de prix

3.3.2 Création d'un Événement

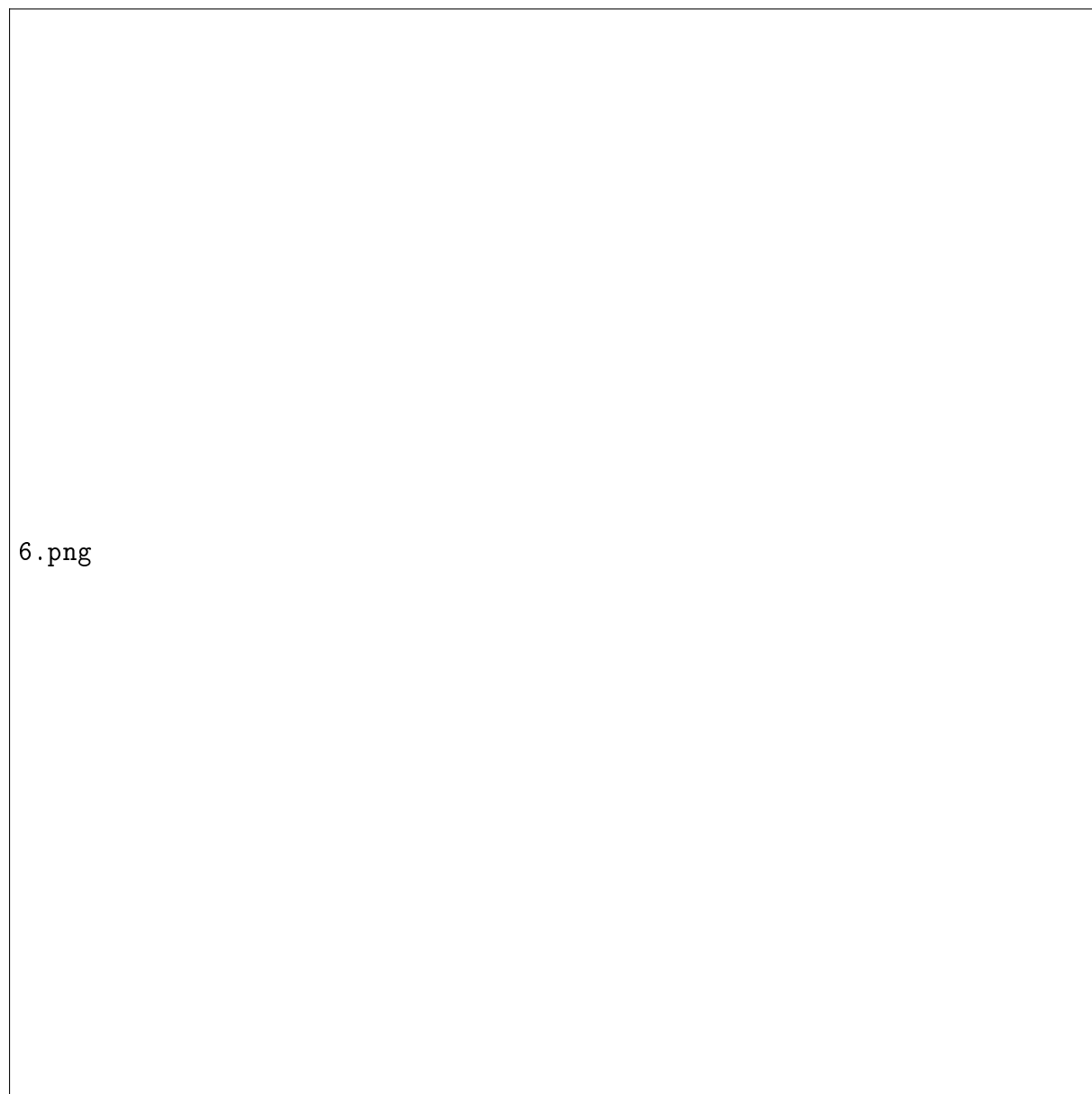


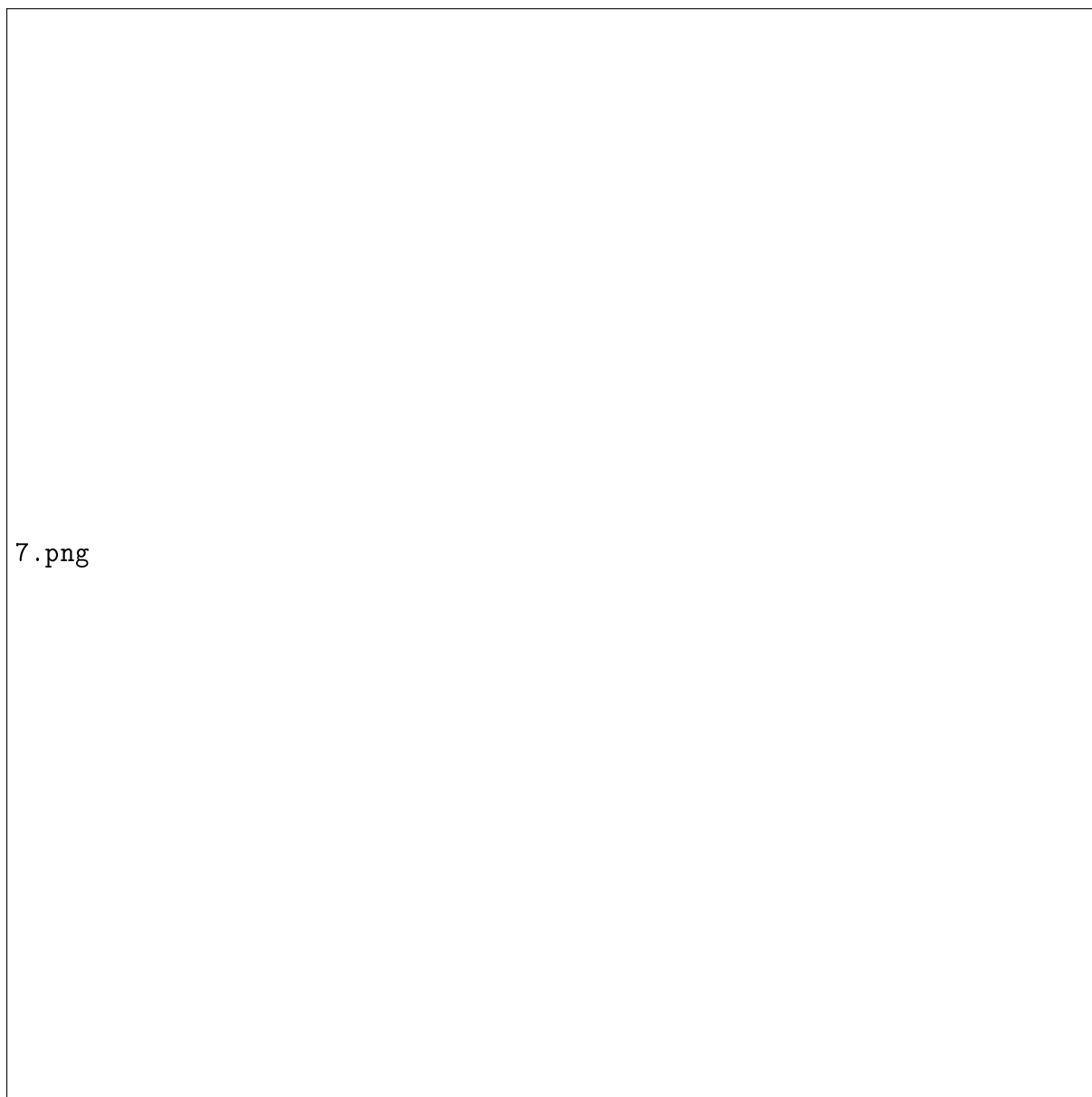
FIGURE 5 – Formulaire de création d'événement

Informations requises :

- Titre (5-200 caractères)
- Description (20-5000 caractères)
- Catégorie
- Date et heure de début/fin
- Localisation (adresse, ville, code postal, pays)
- Capacité (1-100 000)
- Prix (optionnel, 0 = gratuit)
- Tags (optionnel, max 10)

— Image de couverture (optionnel)

3.3.3 Détail d'un Événement



7.png

FIGURE 6 – Page de détail d'un événement

3.4 Mes Événements et Inscriptions

3.4.1 Mes Événements Créés

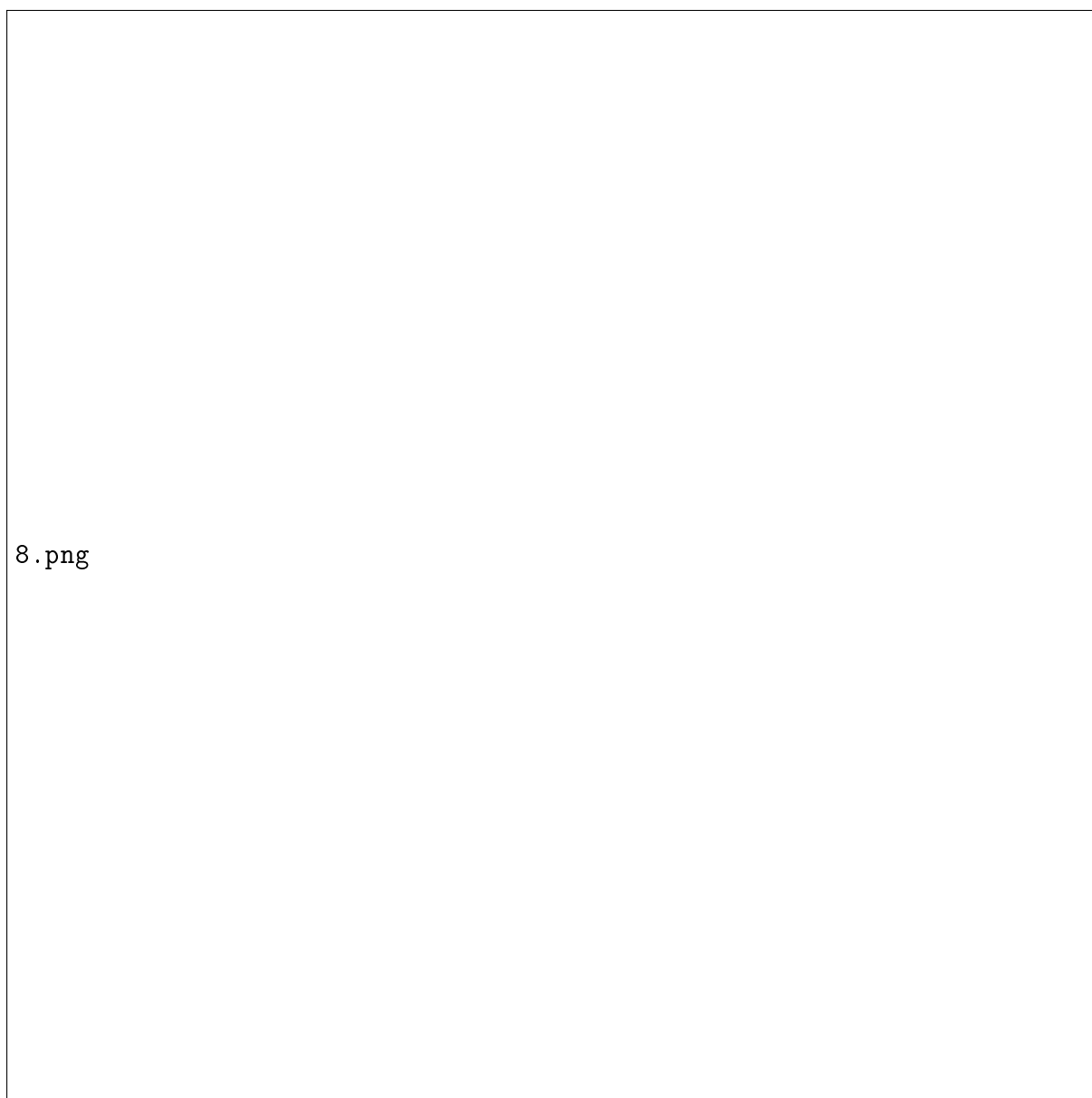


FIGURE 7 – Mes événements créés

3.4.2 Mes Inscriptions



FIGURE 8 – Mes inscriptions aux événements

4 Documentation API

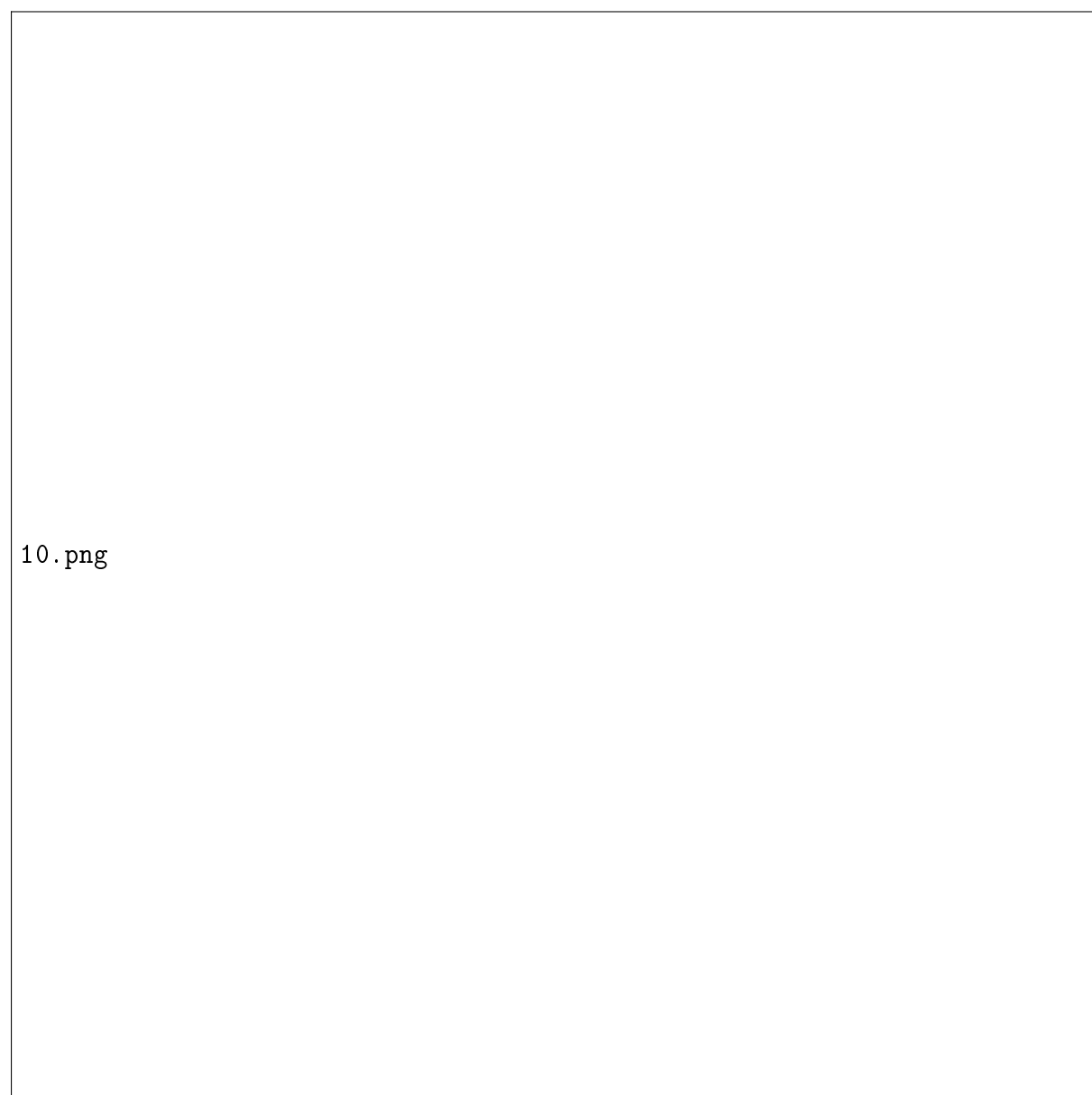
4.1 Informations Générales

| | |
|-----------------------|---------------------------|
| Base URL | http://localhost:5000/api |
| Format | JSON |
| Authentication | Bearer Token (JWT) |
| Rate Limiting | 100 requêtes / 15 minutes |

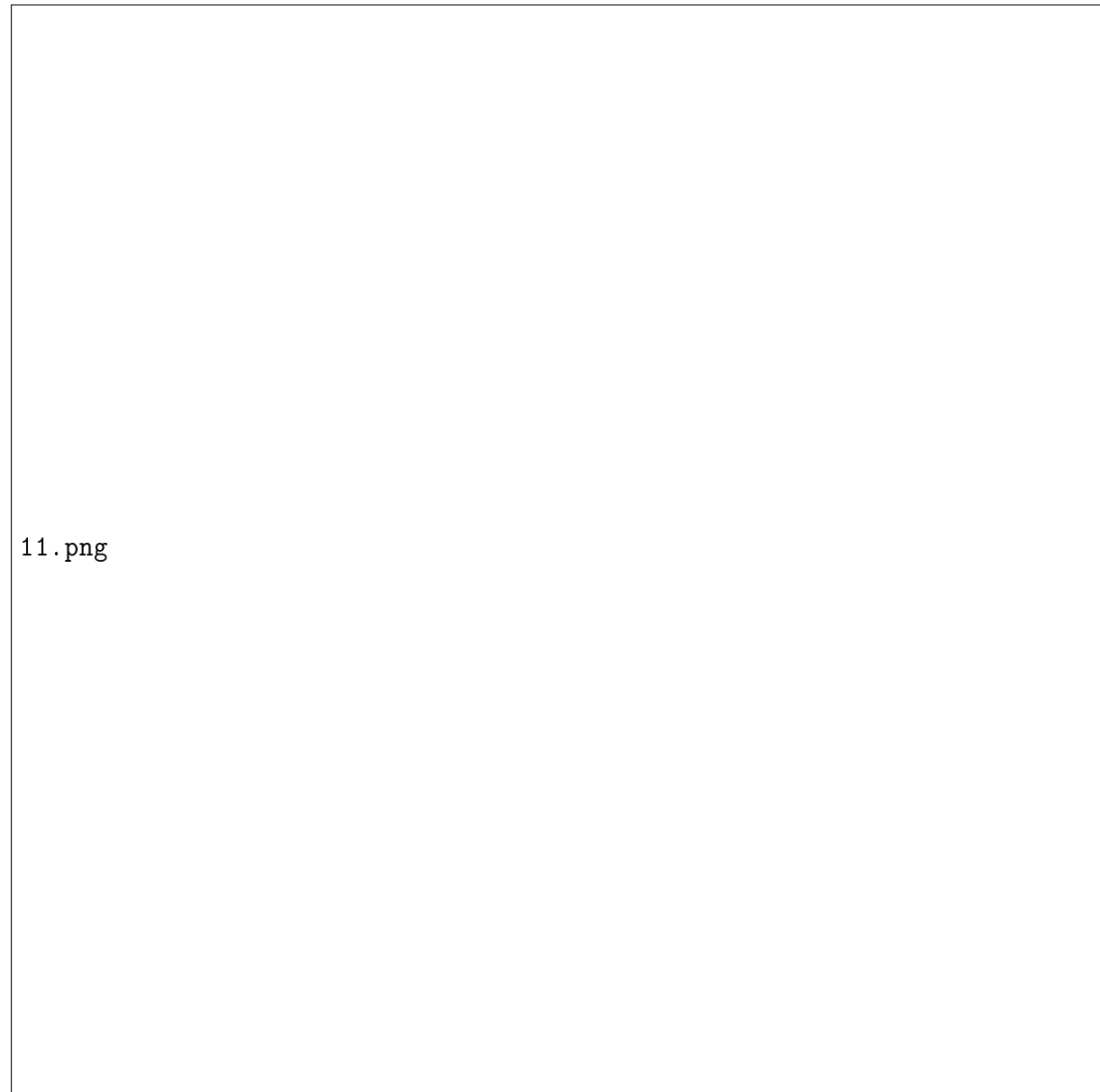
TABLE 2 – Configuration API

4.2 Format des Réponses

4.2.1 Réponse Succès



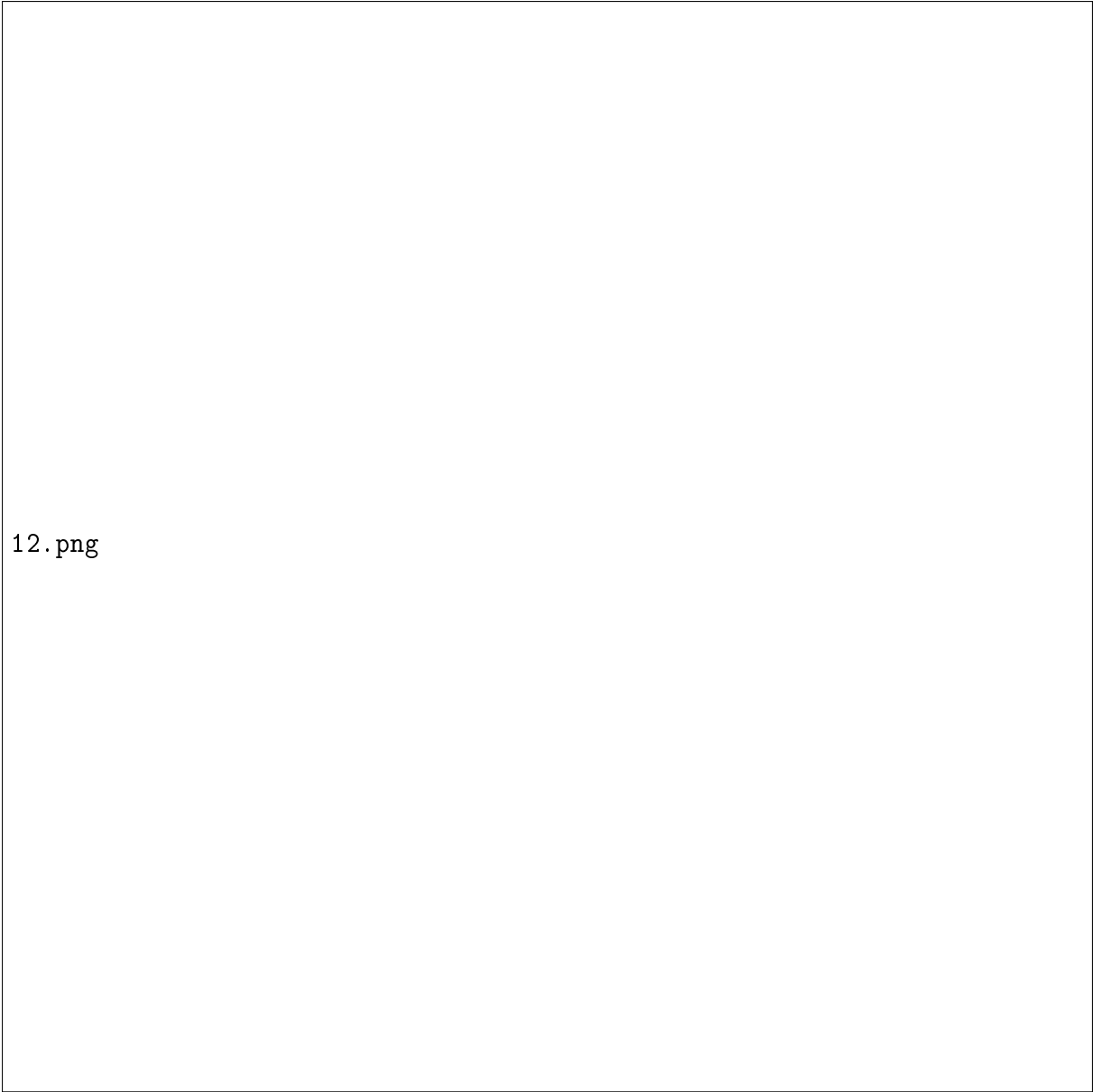
4.2.2 Réponse Erreur



4.3 Authentification

4.3.1 **POST** /api/auth/register

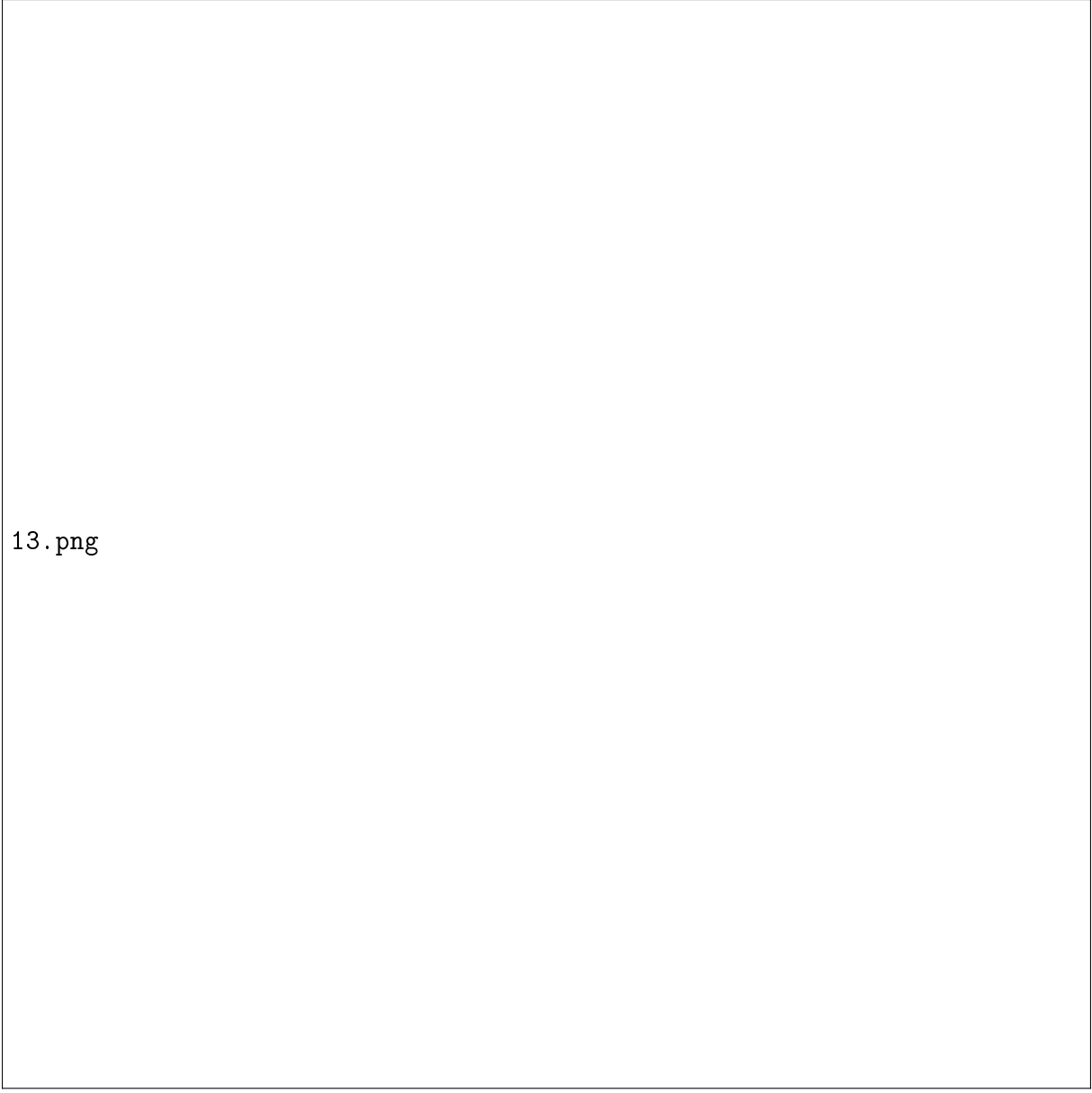
Inscription d'un nouvel utilisateur.



12.png

4.3.2 **POST** /api/auth/login

Connexion utilisateur.



13.png

4.3.3 **POST** /api/auth/logout

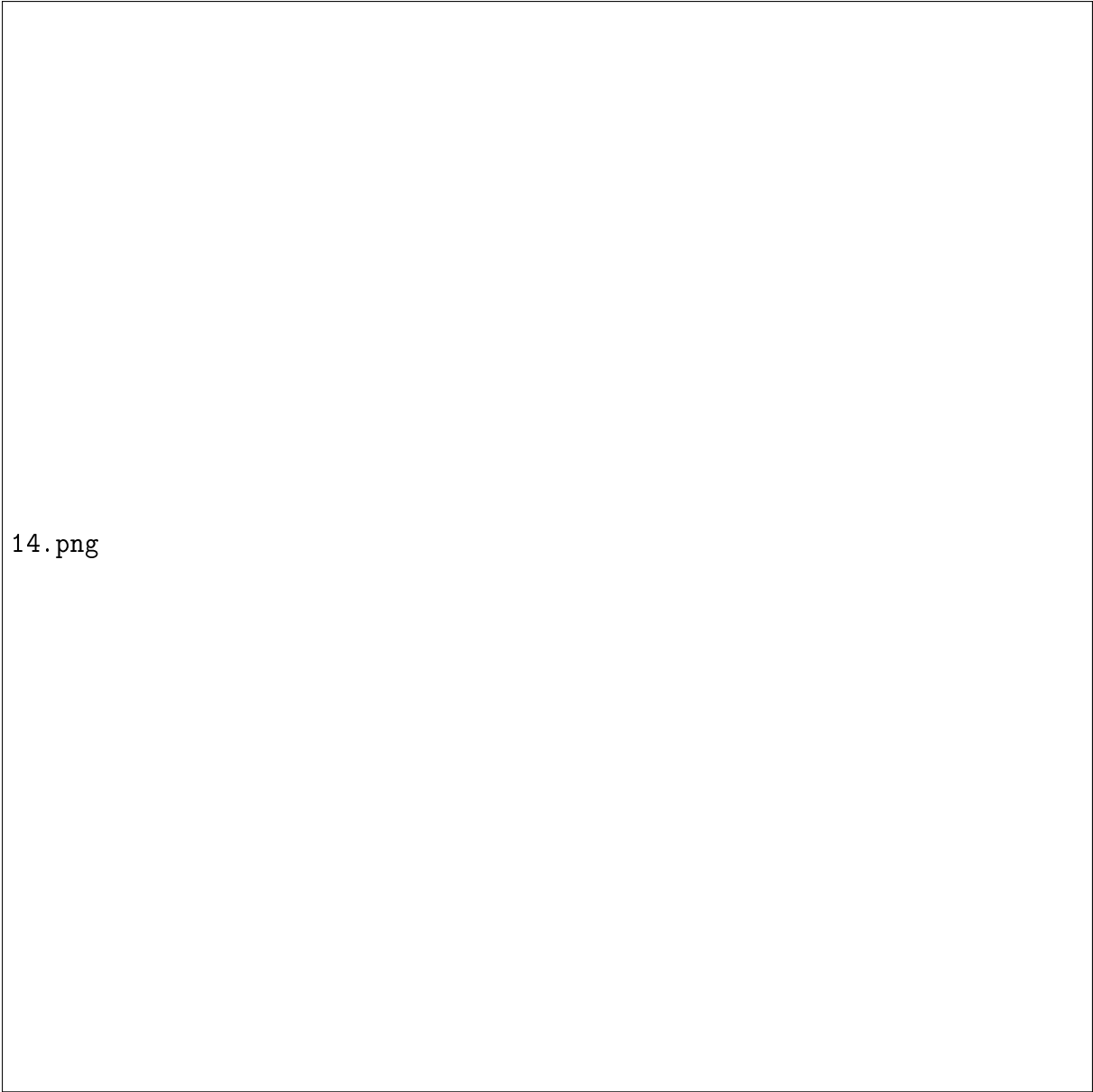
Déconnexion (authentifié requis).

Headers : Authorization: Bearer <accessToken>

4.3.4 **GET** /api/auth/me

Récupérer le profil utilisateur courant.

Headers : Authorization: Bearer <accessToken>



14.png

4.4 Événements

4.4.1 GET /api/events

Liste des événements avec pagination et filtres.

Paramètres de requête :

| Param | Type | Défaut | Description |
|---------------|--------|------------|------------------------------|
| page | number | 1 | Numéro de page |
| limit | number | 10 | Éléments par page (max 100) |
| sort | string | -createdAt | Tri (ex : -startDate, title) |
| search | string | - | Recherche textuelle |
| category | string | - | Filtrer par catégorie |
| status | string | - | Filtrer par statut (admin) |
| city | string | - | Filtrer par ville |
| startDateFrom | date | - | Date début minimum |
| startDateTo | date | - | Date début maximum |
| priceMin | number | - | Prix minimum |
| priceMax | number | - | Prix maximum |

TABLE 3 – Paramètres de filtrage des événements

Exemple :

15.png

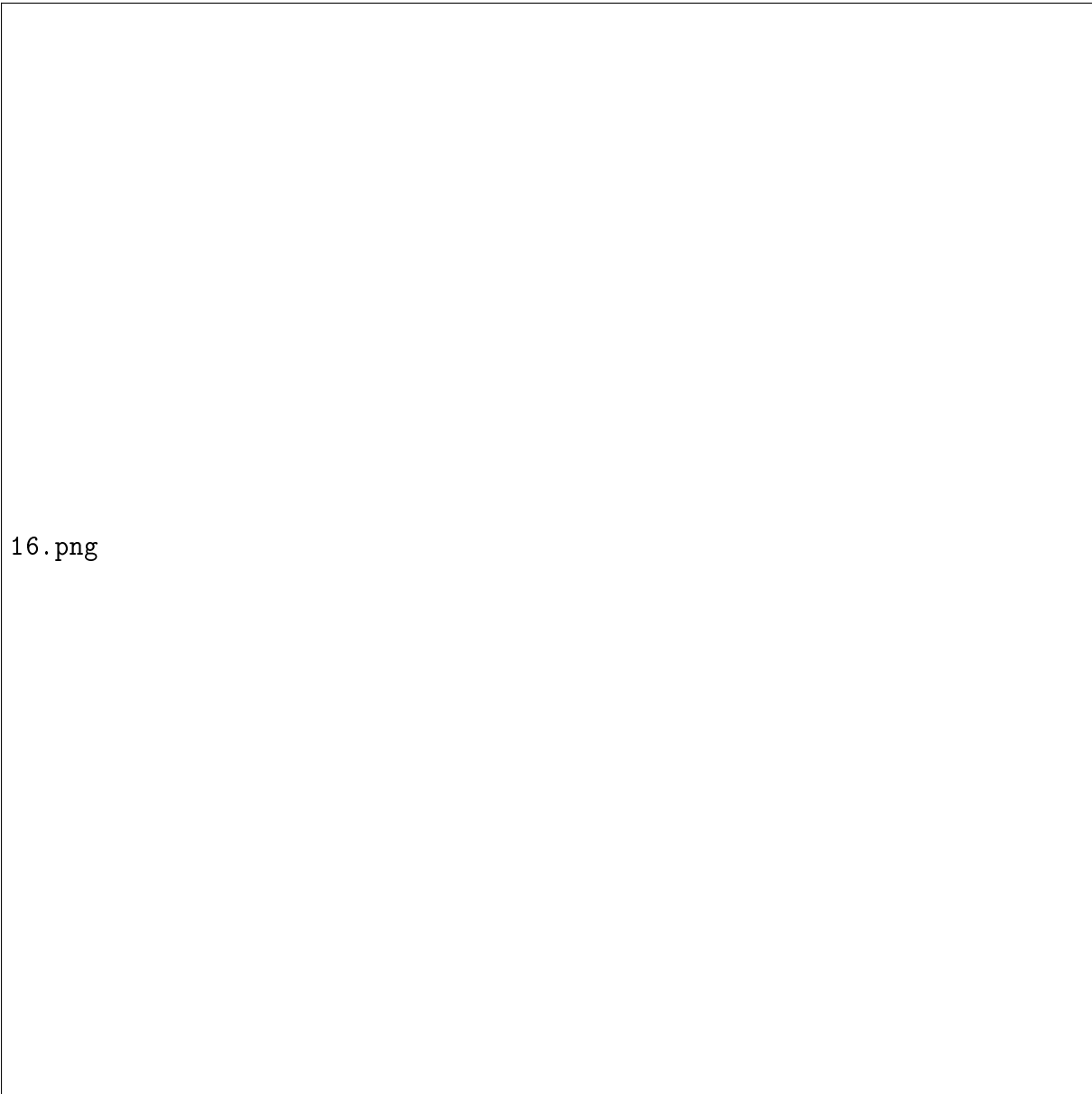
4.4.2 GET /api/events/ :id

Détail d'un événement.

4.4.3 POST /api/events

Créer un événement (authentifié).

Headers : Authorization: Bearer <accessToken>



16.png

4.4.4 PUT /api/events/ :id

Modifier un événement (propriétaire ou admin).

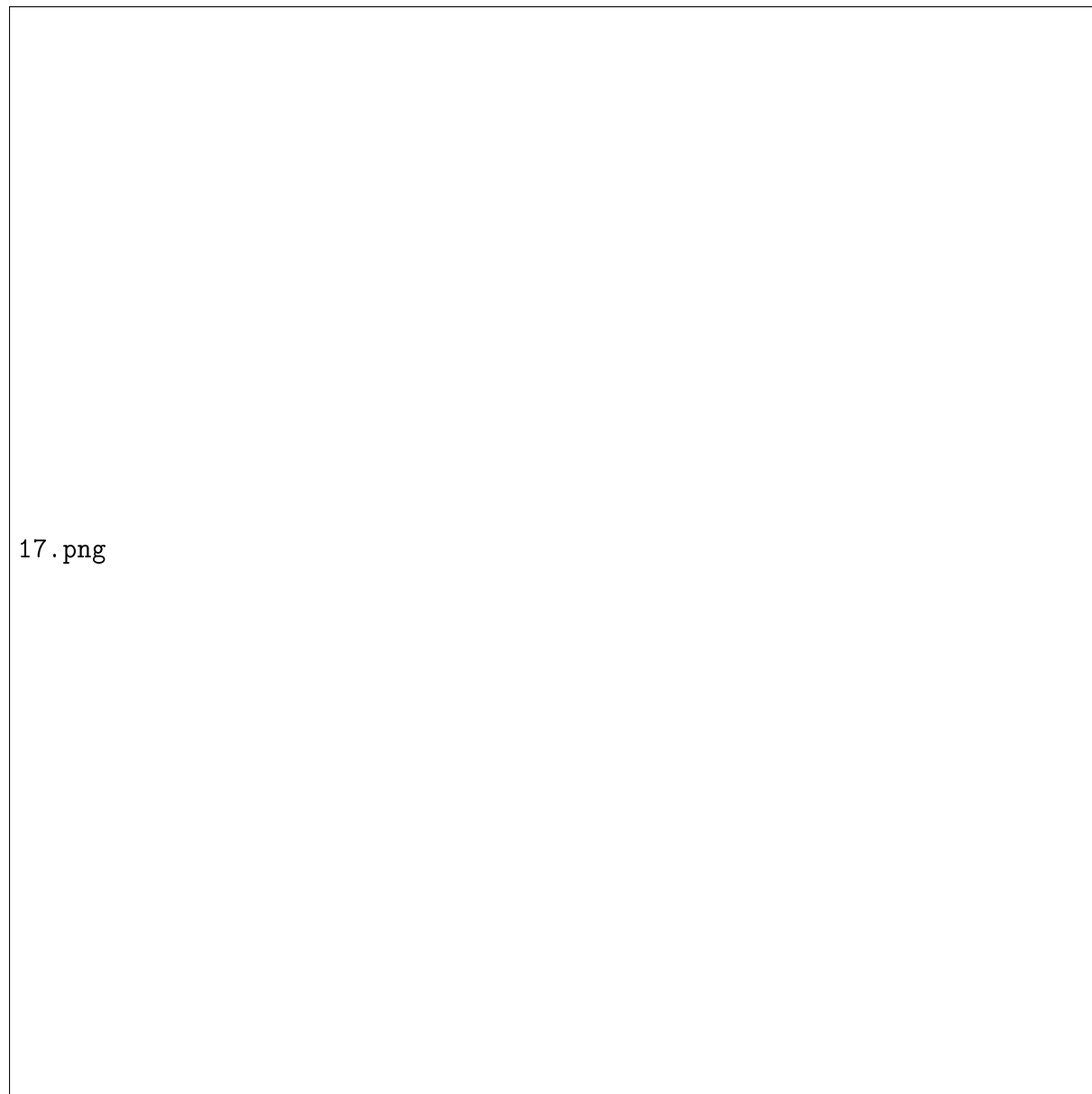
4.4.5 DELETE /api/events/ :id

Supprimer un événement (propriétaire ou admin).

4.4.6 **POST** /api/events/ :id/register

S'inscrire à un événement (authentifié).

Réponse (201 Created) :



17.png

4.4.7 **DELETE** /api/events/ :id/register

Se désinscrire d'un événement.

4.4.8 **GET** /api/events/ :id/registrations

Liste des inscrits (propriétaire ou admin).

4.4.9 **POST** /api/events/ :id/cover

Upload image de couverture (multipart/form-data).

Form Data : cover: <fichier image>

4.5 Utilisateur

4.5.1 **GET** /api/users/me/events

Mes événements créés.

4.5.2 **GET** /api/users/me/registrations

Mes inscriptions.

4.5.3 **POST** /api/users/me/avatar

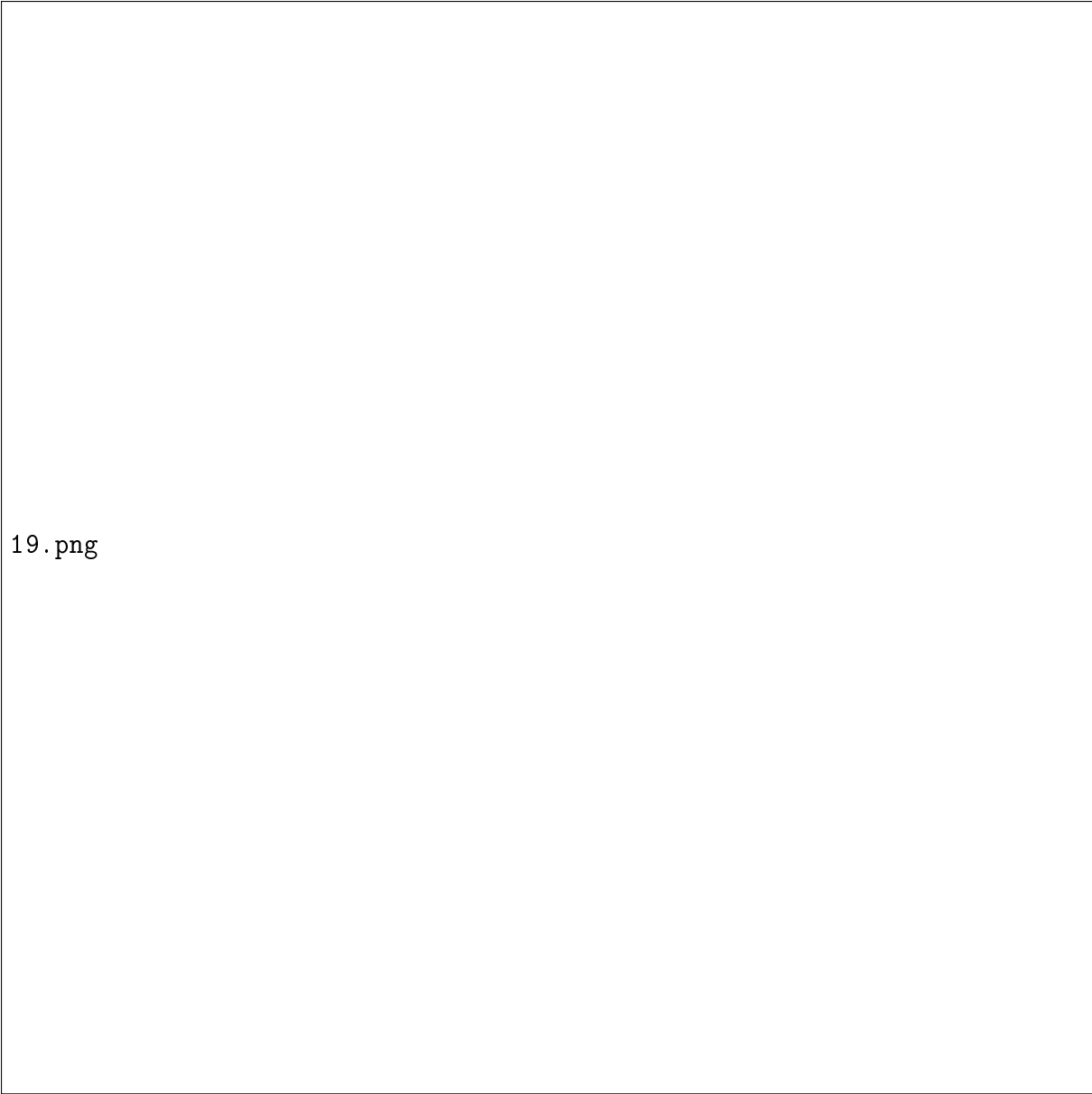
Upload avatar (multipart/form-data).

4.6 Statistiques

4.6.1 **GET** /api/stats/dashboard

Statistiques générales du dashboard.

Réponse :



19.png

4.6.2 **GET** /api/stats/events/category

Répartition des événements par catégorie.

4.6.3 **GET** /api/stats/events/status

Répartition par statut.

4.6.4 **GET** /api/stats/events/trend

Tendance de création des événements (12 mois).

4.6.5 **GET** /api/stats/registrations/trend

Tendance des inscriptions (12 mois).

4.6.6 **GET** /api/stats/events/top

Top 5 événements par inscriptions.

4.6.7 **GET** /api/stats/events/upcoming

Événements à venir (30 jours).

4.6.8 **GET** /api/stats/admin

Statistiques admin globales (admin uniquement).

4.7 Fichiers

4.7.1 **GET** /api/files

Liste des fichiers de l'utilisateur.

4.7.2 **GET** /api/files/:id

Détail d'un fichier.

4.7.3 **GET** /api/files/:id/download

Télécharger un fichier.

4.7.4 **GET** /api/files/:id/preview

Prévisualiser un fichier (inline).

4.7.5 **DELETE** /api/files/:id

Supprimer un fichier.

4.8 Codes d'Erreur

| Code | HTTP | Description |
|--------------------------|------|---------------------------------|
| VALIDATION_ERROR | 400 | Données invalides |
| INVALID_CREDENTIALS | 401 | Email ou mot de passe incorrect |
| TOKEN_EXPIRED | 401 | Token JWT expiré |
| TOKEN_INVALID | 401 | Token JWT invalide |
| TOKEN_MISSING | 401 | Token manquant |
| FORBIDDEN | 403 | Accès refusé |
| INSUFFICIENT_PERMISSIONS | 403 | Permissions insuffisantes |
| NOT_FOUND | 404 | Ressource non trouvée |
| EVENT_NOT_FOUND | 404 | Événement non trouvé |
| USER_NOT_FOUND | 404 | Utilisateur non trouvé |
| USER_EXISTS | 409 | Email déjà utilisé |
| ALREADY_REGISTERED | 409 | Déjà inscrit |
| EVENT_FULL | 409 | Événement complet |
| FILE_TOO_LARGE | 400 | Fichier trop volumineux |
| FILE_TYPE_NOT_ALLOWED | 400 | Type de fichier non autorisé |
| INTERNAL_ERROR | 500 | Erreur serveur |

TABLE 4 – Codes d'erreur de l'API

5 Annexes

5.1 Structure du Projet

```

eventhub/
|-- backend/
|   |-- src/
|   |   |-- config/           # Configuration (DB, erreurs)
|   |   |-- controllers/      # Controleurs HTTP
|   |   |-- middlewares/      # Middlewares (auth, upload)
|   |   |-- models/           # Modeles Mongoose
|   |   |-- routes/           # Definitions des routes
|   |   |-- services/         # Logique metier
|   |   |-- seeds/            # Scripts de seed
|   |   |-- server.js         # Point d'entree
|   |-- uploads/              # Fichiers uploades
|   |-- package.json
|
|-- frontend/
|   |-- src/
|   |   |-- components/       # Composants React
|   |   |-- contexts/         # Contextes (Auth)
|   |   |-- pages/            # Pages de l'application
|   |   |-- services/         # Services API
|   |   |-- utils/            # Utilitaires
|   |   |-- App.jsx           # Composant principal
|   |   |-- main.jsx          # Point d'entree
|   |-- index.html
|   |-- package.json
|   |-- vite.config.js
|   |-- tailwind.config.js
|
|-- docs/                     # Documentation
|-- README.md

```

5.2 Commandes Utiles

| Commande | Description |
|---------------|------------------------------|
| npm run dev | Lancer en mode développement |
| npm start | Lancer en production |
| npm run build | Build frontend production |
| npm run seed | Peupler la base de données |
| npm test | Exécuter les tests |

TABLE 5 – Commandes NPM principales

5.3 Types MIME Autorisés

— image/jpeg

- image/png
- image/gif
- image/webp

Taille maximale : **5 MB**