

Système de Gestion des Visiteurs - Backend API

Table des matières

- [Technologies utilisées](#)
- [Configuration](#)
- [Authentification](#)
- [API Visiteurs](#)
- [API Visites](#)
- [API Services](#)

Technologies utilisées

- Java 21
- Spring Boot 3.4.3
- JWT pour l'authentification
- Base de données : H2 (dev) / MySQL (prod)
- Maven
- Swagger/OpenAPI

Configuration

Base de données (Développement - H2)

```
spring.datasource.url=jdbc:h2:file:./data/visitorsdb
spring.datasource.username=sa
spring.datasource.password=
```

Base de données (Production - MySQL)

```
spring.datasource.url=jdbc:mysql://localhost:3306/gvdb
spring.datasource.username=root
spring.datasource.password=****
```

Authentification

Connexion

Requête :

```
POST /api/login
```

Corps de la requête :

```
{
  "username": "admin",
  "password": "admin",
  "role": "ROLE_ADMIN"
}
```

Réponse :

```
{
  "token": "eyJhbGciOiJIUzI1NiJ9...",
  "role": "ROLE_ADMIN"
}
```

API Visiteurs

Obtenir tous les visiteurs

Requête :

```
GET /api/visitors
```

Réponse :

```
[
  {
    "id": 1,
    "nom": "Dupont",
    "prenom": "Jean",
    "numeroId": "VIS123",
    "satisfactionMoy": 4.5,
    "nbVisit": 3
  }
]
```

Rechercher des visiteurs

Requête :

```
GET /api/visitors/search?query=jean dupont
```

Obtenir un visiteur par ID

Requête :

```
GET /api/visitors/{id}
```

Modifier un visiteur

Requête :

```
PUT /api/visitors/{id}
```

Corps de la requête :

```
{
  "nom": "Dupont",
  "prenom": "Jean",
  "numeroId": "VIS123",
  "satisfactionMoy": 4.5,
  "nbVisit": 3
}
```

Supprimer un visiteur

Requête :

```
DELETE /api/visitors/{id}
```

API Visites

Créer une nouvelle visite

Requête :

```
POST /api/visits
```

Corps de la requête :

```
{
  "nom": "Dupont",
```

```
"prenom": "Jean",  
"numeroId": "VIS123",  
"heureArrivee": "2024-05-25T09:00:00",  
"serviceId": 1,  
"satisfaction": null  
}
```

Obtenir toutes les visites

Requête :

```
GET /api/visits
```

Réponse :

```
[  
  {  
    "id": 1,  
    "nom": "Dupont",  
    "prenom": "Jean",  
    "numeroId": "VIS123",  
    "heureArrivee": "2024-05-25T09:00:00",  
    "heureSortie": "2024-05-25T10:30:00",  
    "serviceVisite": "Service Technique",  
    "serviceId": 1,  
    "statut": "PRESENT",  
    "satisfaction": 5  
  }  
]
```

Mettre à jour le statut d'une visite

Requête :

```
PUT /api/visits/{id}/statut?statut=PRESENT
```

Mettre à jour la satisfaction

Requête :

```
PUT /api/visits/{id}/satisfaction?satisfaction=5
```

Statistiques des visites du jour

Requête :

```
GET /api/visits/count/today
```

Réponse :

```
{
  "total": 10,
  "presents": 5,
  "pending": 3
}
```

API Services

Créer un nouveau service

Requête :

```
POST /api/services
```

Corps de la requête :

```
{
  "nomService": "Service Technique",
  "responsable": "Marie Martin",
  "telephone": "0123456789",
  "statut": "libre"
}
```

Obtenir tous les services

Requête :

```
GET /api/services
```

Réponse :

```
[
  {
    "id": 1,
    "nomService": "Service Technique",
    "responsable": "Marie Martin",
    "telephone": "0123456789",
    "statut": "libre"
  }
]
```

Rechercher des services

Requête :

```
GET /api/services/search?query=technique
```

Modifier un service

Requête :

```
PUT /api/services/{id}
```

Corps de la requête :

```
{
  "nomService": "Service Technique",
  "responsable": "Marie Martin",
  "telephone": "0123456789",
  "statut": "occupe"
}
```

Supprimer un service

Requête :

```
DELETE /api/services/{id}
```

États des visites

Les visites peuvent avoir l'un des états suivants :

- **EN_ATTENTE** : Le visiteur attend d'être reçu
- **PRESENT** : Le visiteur est en cours de visite
- **CLOTURE** : La visite est terminée

Sécurité

- Authentification requise via JWT pour tous les endpoints (sauf **/api/login**)
- Format du header : **Authorization: Bearer <token>**
- Deux rôles disponibles :
 - **ROLE_ADMIN** : Accès complet
 - **ROLE_USER** : Accès en lecture seule

Documentation API

Interface Swagger disponible à : **<http://localhost:8060/swagger-ui.html>**

Installation et Démarrage

```
# Compiler le projet
./mvnw clean package

# Démarrer en mode développement
./mvnw spring-boot:run -Dspring.profiles.active=dev

# Démarrer en mode production
./mvnw spring-boot:run -Dspring.profiles.active=prod
```