

Documentation de la route : Création d'un Rescue Service

URL :

POST /create-rescue-service

Description :

Cette route permet aux administrateurs de créer un service de secours. Elle vérifie si les informations nécessaires sont fournies, empêche les doublons de noms, et enregistre un nouveau service actif.

Headers :

Champ	Type	Obligatoire	Description
Authorization	string	Oui	Le token JWT au format Bearer <token> pour l'authentification.

Corps de la requête (Request Body) :

Champ	Type	Obligatoire	Description
name	string	Oui	Nom du service de secours.
service_type	string	Oui	Type du service de secours (par ex., "Médical").
contact_number	string	Oui	Numéro de contact du service de secours.
description	string	Oui	Brève description du service de secours.

Exemple :

```
{
  "name": "Service Ambulancier",
  "service_type": "Médical",
  "contact_number": "123456789",
  "description": "Fournit une assistance médicale d'urgence."
}
```

Réponses :

1. Succès :

- Code HTTP : 201 Created
- Description : Le service de secours a été créé avec succès.

Corps de la réponse :

Champ	Type	Description
id	string	ID unique du service de secours créé.
name	string	Nom du service de secours.
serviceType	string	Type du service de secours.
contactNumber	string	Numéro de contact du service de secours.
description	string	Description du service de secours.
isActive	boolean	Indique si le service est actif (true par défaut).

Exemple :

```
{
  "message": "Service de secours créé avec succès",
  "data": {
    "id": "uuid-service",
    "name": "Service Ambulancier",
    "serviceType": "Médical",
    "contactNumber": "123456789",
    "description": "Fournit une assistance médicale d'urgence.",
    "isActive": true
  }
}
```

2. Champs requis manquants :

- **Code HTTP :** 400 Bad Request
- **Description :** Les champs obligatoires ne sont pas fournis.

Exemple :

```
{
  "message": "Champs requis manquants",
  "data": {
    "missingFields": ["name", "service_type", "contact_number"]
  }
}
```

3. Nom de service en double :

- **Code HTTP :** 400 Bad Request
- **Description :** Un service de secours avec le même nom existe déjà.

Exemple :

```
{
  "message": "Un service de secours avec ce nom existe déjà",
  "data": null
}
```

4. Erreur serveur :

- **Code HTTP :** 500 Internal Server Error
- **Description :** Une erreur s'est produite sur le serveur.

Exemple :

```
{
  "message": "Erreur lors de la création du service de secours",
  "data": "Détail de l'erreur technique."
}
```

Notes techniques :

1. Vérification des permissions :

- Seuls les utilisateurs avec le rôle **ADMIN** peuvent accéder à cette route.

2. Vérification des doublons :

- La route vérifie si un service de secours avec le même **name** existe déjà avant de créer un nouveau service.

3. Service actif par défaut :

- Le service de secours créé est actif par défaut (**is_active: true**).
-

Commandes de test avec cURL :

```
curl -X POST http://localhost:3000/create-rescue-service \
-H "Content-Type: application/json" \
-H "Authorization: Bearer <token>" \
-d '{
```

```
"name": "Service Ambulancier",  
"service_type": "Médical",  
"contact_number": "123456789",  
"description": "Fournit une assistance médicale d'urgence."  
'
```

Liens associés :

- [Route pour désactiver un RescueService](#)
- [Route pour activer un RescueService](#)