

Documentation de la route : Création d'un Rescue Member

URL :

POST /create-rescue-member

Description :

Cette route permet de créer un membre de secours. Seuls les administrateurs peuvent accéder à cette route pour ajouter un membre associé à un service de secours existant.

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
email	string	Oui	Email du membre de secours.
password	string	Oui	Mot de passe pour le compte.
firstName	string	Oui	Prénom du membre de secours.
lastName	string	Oui	Nom du membre de secours.
phoneNumber	string	Oui	Numéro de téléphone du membre.
position	string	Oui	Poste du membre de secours (par ex., "Chef").
badgeNumber	string	Oui	Numéro de badge unique pour le membre.
rescueServiceId	string	Oui	ID du service de secours associé.

Exemple :

```
{
  "email": "john.doe@example.com",
  "password": "securePassword123",
  "firstName": "John",
  "lastName": "Doe",
  "phoneNumber": "0123456789",
  "position": "Chef",
  "badgeNumber": "CH12345",
}
```

```
"rescueServiceId": "uuid-service"
}
```

Réponses :

1. Succès :

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

Corps de la réponse :

Champ	Type	Description
id	string	ID unique du membre créé.
email	string	Email du membre.
firstName	string	Prénom du membre.
lastName	string	Nom du membre.
phoneNumber	string	Numéro de téléphone du membre.
role	string	Rôle de l'utilisateur (RESCUE_MEMBER).
position	string	Poste du membre de secours.
badgeNumber	string	Numéro de badge du membre.
rescueService	object	Détails du service de secours associé.
rescueService.id	string	ID du service de secours associé.
rescueService.name	string	Nom du service de secours associé.

Exemple :

```
{
  "message": "Membre de secours créé avec succès",
  "data": {
    "id": "uuid-user",
    "email": "john.doe@example.com",
    "firstName": "John",
    "lastName": "Doe",
    "phoneNumber": "0123456789",
    "role": "RESCUE_MEMBER",
    "position": "Chef",
    "badgeNumber": "CH12345",
    "rescueService": {
      "id": "uuid-service",
```

```
    "name": "Service Ambulancier"
  }
}
```

2. Champs requis manquants :

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

Exemple :

```
{
  "message": "Champs requis manquants",
  "data": {
    "missingFields": ["email", "password"]
  }
}
```

3. Email déjà utilisé :

- **Code HTTP :** 400 Bad Request
- **Description :** L'email fourni est déjà utilisé par un autre utilisateur.

Exemple :

```
{
  "message": "Cet email est déjà utilisé",
  "data": null
}
```

4. Badge déjà utilisé :

- **Code HTTP :** 400 Bad Request
- **Description :** Le numéro de badge est déjà utilisé par un autre membre de secours.

Exemple :

```
{
  "message": "Ce numéro de badge est déjà utilisé",
```

```
"data": null
}
```

5. Service introuvable :

- **Code HTTP :** 400 Bad Request
- **Description :** L'ID du service de secours fourni n'existe pas.

Exemple :

```
{
  "message": "Service de secours introuvable",
  "data": null
}
```

6. Accès non autorisé :

- **Code HTTP :** 401 Unauthorized
- **Description :** L'utilisateur authentifié n'a pas les permissions nécessaires (non-ADMIN).

Exemple :

```
{
  "message": "Accès interdit : seuls les administrateurs peuvent effectuer cette action",
  "data": null
}
```

7. 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 membre 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. Validation de l'unicité :

- Vérifie que l'email et le numéro de badge ne sont pas déjà utilisés.

3. Relation avec le service de secours :

- L'ID du service de secours (**rescueServiceId**) doit correspondre à un service existant.

4. Mot de passe sécurisé :

- Le mot de passe est haché avant d'être enregistré dans la base de données.

Commandes de test avec cURL :

```
curl -X POST http://localhost:3000/create-rescue-member \
-H "Content-Type: application/json" \
-H "Authorization: Bearer <token>" \
-d '{
  "email": "john.doe@example.com",
  "password": "securePassword123",
  "firstName": "John",
  "lastName": "Doe",
  "phoneNumber": "0123456789",
  "position": "Chef",
  "badgeNumber": "CH12345",
  "rescueServiceId": "uuid-service"
}'
```

Liens associés :

- [Route pour modifier un Rescue Member](#)
- [Route pour désactiver un Rescue Member](#)