

Documentation de l'API de Gestion des Clients

Introduction

Cette API permet la gestion des clients, incluant la création, la lecture, la mise à jour et la suppression de clients. L'API utilise JWT pour la sécurisation des routes sensibles.

Base URL

<http://localhost:3000/api>

Endpoints

1. Authentification

POST `/login`

Description : Authentifie un client et renvoie un token JWT.

- Corps de la requête :

```
{  
  "email": "string",  
  "mot_de_passe": "string"  
}
```

Réponse :

- Code : `200 OK`
- Corps de la réponse :

```
{  
  "id": "int",  
  "email": "string",  
  "accessToken": "string"  
}
```

Codes d'erreur :

- `404 Not Found` : Client non trouvé.
- `401 Unauthorized` : Mot de passe invalide.

2. Créer un client

POST `/customers`

Description : Crée un nouveau client dans la base de données.

- Corps de la requête :

```
{  
  "nom": "string",  
  "prenom": "string",  
  "date_naissance": "date",  
  "email": "string",  
  "mot_de_passe": "string",  
  "adresse": "string"  
}
```

Réponse :

- Code : `200 Created`
- Corps de la réponse :

```
{  
  "id": "int",  
  "nom": "string",  
  "prenom": "string",  
  "date_naissance": " date ",  
  "email": "string",  
  "adresse": "string"  
}
```

Codes d'erreur :

- `400 Bad Request` : Requête mal formée ou données manquantes.
- `500 Internal Server Error` : Erreur lors de la création du client.

3. Récupérer tous les clients

GET `/customers`

Description : Récupère la liste de tous les clients.

Headers :

`x-access-token` : Token JWT pour l'authentification.

Réponse :

- Code : `200 OK`
- Corps de la réponse :

```
[  
  {  
    "id": "int",  
    "nom": "string",  
    "prenom": "string",  
    "date_naissance": "date",  
    "email": "string",  
    "adresse": "string"  
  },  
  ...  
]
```

Codes d'erreur :

- `401 Unauthorized` : Token JWT manquant ou invalide.
- `500 Internal Server Error` : Erreur lors de la récupération des clients.

4. Récupérer un client par ID

GET `/customers/:clientId`

Description : Récupère un client spécifique en utilisant son ID.

Paramètres d'URL : ‘ *clientId* ’ : ID du client (int)

Réponse :

- Code : `200 OK`
- Corps de la réponse :

```
{  
  "id": "int",  
  "nom": "string",  
  "prenom": "string",  
  "date_naissance": "date",  
  "email": "string",  
  "adresse": "string"  
}
```

Codes d'erreur :

- `404 Not Found` : Client non trouvé.
- `500 Internal Server Error` : Erreur lors de la récupération du client.

5. Mettre à jour un client

PUT `/customers/:clientId`

Description : Met à jour les informations d'un client existant.

Paramètres d'URL : ‘ *clientId* ’ : ID du client (int)

Headers : `x-access-token` : Token JWT pour l'authentification.

Corps de la requête :

```
{  
  "nom": "string",  
  "prenom": "string",  
  "date_naissance": " date ",  
  "email": "string",  
  "mot_de_passe": "string",  
  "adresse": "string"  
}
```

Réponse :

- Code : `200 OK`
- Corps de la réponse :

```
{  
  "id": "int",  
  "nom": "string",  
  "prenom": "string",  
  "date_naissance": " date ",  
  "email": "string",  
  "adresse": "string"  
}
```

Codes d'erreur :

- `400 Bad Request` : Requête mal formée ou données manquantes.
- `404 Not Found` : Client non trouvé.
- `500 Internal Server Error` : Erreur lors de la mise à jour du client.

6. Supprimer un client

DELETE `/customers/:clientId`

Description : Supprime un client de la base de données.

Paramètres d'URL : `clientId` : ID du client (int)

Headers : `x-access-token` : Token JWT pour l'authentification.

Réponse :

- Code : `200 OK`
- Corps de la réponse :

```
{  
  "message": "Le client a été supprimé avec succès !"  
}
```

Codes d'erreur :

- `404 Not Found` : Client non trouvé.
- `500 Internal Server Error` : Erreur lors de la suppression du client.

Sécurité

Certaines routes de l'API (comme la récupération, la mise à jour, et la suppression des clients) sont protégées par des tokens JWT.

Assurez-vous d'inclure le token d'accès dans les en-têtes de requête sous `x-access-token` pour ces routes.

Endpoint	Méthode HTTP	Description	Corps de la requête	Réponse	Token
/login	POST	Authentifie un client et renvoie un token JWT	{ "email": "string", "mot_de_passe": "string" }	{ "id": "int", "email": "string", "accessToken": "string" }	Non
/customers	POST	Crée un nouveau client	{ "nom": "string", "prenom": "string", "date_naissance": "date", "email": "string", "mot_de_passe": "string", "adresse": "string" }	{ "id": "int", "nom": "string", "prenom": "string", "date_naissance": "date", "email": "string", "adresse": "string" }	Non
/customers	GET	Récupère la liste de	N/A	[{ "id": "int", "nom": "string", "prenom":	Oui

		tous les clients		"string", "date_naissance": "string", "email": "string", "adresse": "string" }, ...]	
/customers/:clientId	GET	Récupère un client spécifique par son ID	N/A	{ "id": "int", "nom": "string", "prenom": "string", "date_naissance": "string", "email": "string", "adresse": "string" }	Non
/customers/:clientId	PUT	Met à jour un client existant par son ID	{ "nom": "string", "prenom": "string", "date_naissance": ": "string", "email": "string", "mot_de_passe": "string", "adresse": "string" }	{ "id": "int", "nom": "string", "prenom": "string", "date_naissance": "string", "email": "string", "adresse": "string" }	Oui
/customers/:clientId	DELETE	Supprime un client par son ID	N/A	{"message": "Le client a été supprimé avec succès !" }	Oui