

API Client Yoozak

Cette documentation décrit les endpoints API disponibles pour la gestion des clients dans l'application Yoozak.

Table des matières

- [Authentification](#)
- [Endpoints](#)
- [Gestion des tokens](#)
- [Exemples avec Postman](#)

Authentification

L'API utilise JWT (JSON Web Tokens) pour l'authentification. Deux types de tokens sont utilisés :

- **Access Token** : Durée de validité de 1 heure
- **Refresh Token** : Durée de validité de 1 jour

Format du Header d'authentification

```
Authorization: Bearer <votre_token>
```

Endpoints

1. Inscription Client

- **URL** : `/api/clients/auth/register/`
- **Méthode** : `POST`
- **Auth requise** : Non
- **Corps de la requête** :

```
{
  "username": "votre_username",
  "password": "votre_password",
  "email": "votre@email.com",
  "first_name": "Prénom",
  "last_name": "Nom"
}
```

- **Réponse réussie** (201 Created) :

```
{
  "refresh": "token_refresh",
  "access": "token_access",
  "user": {
    "id": 1,
    "username": "votre_username",
    "email": "votre@email.com",
    "first_name": "Prénom",
    "last_name": "Nom"
  },
  "token_info": {
    "expires_at": "2024-06-05T13:56:56.000Z",
    "issued_at": "2024-06-05T12:56:56.000Z",
    "type": "Bearer"
  }
}
```

2. Connexion Client

- **URL** : `/api/clients/auth/login/`
- **Méthode** : `POST`
- **Auth requise** : Non
- **Corps de la requête** :

```
{
  "username": "votre_username",
  "password": "votre_password"
}
```

- **Réponse réussie** (200 OK) : Même format que l'inscription

3. Rafraîchir Token

- **URL** : `/api/clients/auth/token/refresh/`
- **Méthode** : `POST`
- **Auth requise** : Non
- **Corps de la requête** :

```
{
  "refresh": "votre_refresh_token"
}
```

- **Réponse réussie** (200 OK) :

```
{
  "access": "nouveau_token_access"
}
```

4. Profil Client

- **URL** : `/api/clients/clients/me/`
- **Méthode** : `GET`
- **Auth requise** : Oui
- **Réponse réussie** (200 OK) :

```
{
  "id": 1,
  "username": "votre_username",
  "email": "votre@email.com",
  "first_name": "Prénom",
  "last_name": "Nom"
}
```

Gestion des tokens

Le système de tokens fonctionne comme suit :

1. Lors de la connexion/inscription, vous recevez :
 - Un access token (validité : 1 heure)
 - Un refresh token (validité : 1 jour)
2. Utilisation :
 - Utilisez l'access token pour toutes les requêtes authentifiées
 - Si l'access token expire, utilisez le refresh token pour en obtenir un nouveau
 - Si le refresh token expire, l'utilisateur doit se reconnecter
3. Détection d'expiration :
 - Le serveur renvoie une erreur 401 si le token est invalide
 - L'en-tête `X-Token-Refresh-Required: true` est ajouté si le token expire bientôt

Exemples avec Postman

Configuration de l'environnement

1. Créez un environnement "Yoozak Local"
2. Ajoutez les variables :
 - `base_url` : `http://127.0.0.1:8000`
 - `access_token` : (vide)

- `refresh_token` : (vide)

Tests des endpoints

1. Inscription :

- Envoyez une requête POST à `{{base_url}}/api/clients/auth/register/`
- Les tokens seront automatiquement sauvegardés dans les variables d'environnement

2. Connexion :

- Envoyez une requête POST à `{{base_url}}/api/clients/auth/login/`
- Les tokens seront mis à jour

3. Accès au profil :

- Ajoutez l'en-tête : `Authorization: Bearer {{access_token}}`
- Envoyez une requête GET à `{{base_url}}/api/clients/clients/me/`

4. Rafraîchissement du token :

- Si vous recevez une erreur 401
- Envoyez une requête POST à `{{base_url}}/api/clients/auth/token/refresh/`
- Le nouveau token d'accès sera automatiquement sauvegardé