Service Role et JWT Authentication dans CrippleFN

Ce document explique comment utiliser les fonctionnalités de Service Role et d'authentification JWT dans l'API CrippleFN.

Configuration

Les variables d'environnement suivantes doivent être configurées dans le fichier `.env` :

```
SUPABASE_URL=https://your-project.supabase.co
SUPABASE_KEY=your-anon-key
SUPABASE_SERVICE_KEY=your-service-role-key
SUPABASE_JWT_SECRET=your-jwt-secret
```

- `SUPABASE_KEY` : Clé publique/anonyme pour les opérations standard
- `SUPABASE_SERVICE_KEY` : Clé service_role qui permet d'effectuer des opérations privilégiées
- `SUPABASE_JWT_SECRET` : Secret utilisé pour signer et vérifier les tokens JWT

Authentification

Création de compte

```
POST /auth/register
{
  "email": "utilisateur@example.com",
  "password": "motdepasse",
  "confirm_password": "motdepasse",
  "first_name": "Prénom",
  "last_name": "Nom"
}
```

Connexion

```
POST /auth/login
{
  "email": "utilisateur@example.com",
  "password": "motdepasse"
}
```

Les deux endpoints retournent un token JWT que vous devez inclure dans vos requêtes suivantes.

Utilisation du token

Incluez le token dans l'en-tête Authorization de vos requêtes :

```
Authorization: Bearer votre-token-jwt
```

Rafraîchissement du token

Si votre token est sur le point d'expirer, vous pouvez le rafraîchir :

```
POST /auth/refresh
Authorization: Bearer votre-token-actuel
```

Service Role

Le service role est un rôle privilégié qui permet d'effectuer des opérations administratives comme :

- Créer/modifier/supprimer des utilisateurs
- Accéder à des tables protégées
- Effectuer des opérations en masse

Le backend utilise le service role pour :

1. L'enregistrement des utilisateurs
2. Les opérations de fond sur les vérifications
3. Les accès privilégiés aux données

Utilisation dans le code

Obtenir un client avec privilèges service_role

```python
from api.dependencies.auth import get_service_role_client

async def my_function():
    # Obtenir un client avec privilèges service_role
    service_client = await get_service_role_client()

    # Effectuer des opérations privilégiées
    users = service_client.auth.admin.list_users()
```

Générer un token service_role

```python
from api.dependencies.auth import get_service_token

async def my_function():
    # Générer un token avec privilèges service_role
```

```
    service_token = await get_service_token()

    # Utiliser le token pour des requêtes à d'autres services
    headers = {"Authorization": f"Bearer {service_token}"}
    # ...
```

Sécurité

- Ne partagez jamais votre clé service_role ou votre secret JWT.
- Limitez l'utilisation du service_role aux opérations qui en ont réellement besoin.
- Vérifiez toujours les tokens JWT avant d'autoriser des opérations privilégiées.
- Les tokens JWT ont une durée de validité limitée (1 heure par défaut).

Exemple d'utilisation

Consultez le fichier `utils/service_role_example.py` pour voir des exemples concrets d'utilisation du service role et des JWT.