

# DigitalBank France

## Documentation API REST

*Supabase PostgreSQL Backend*

## 1. Vue d'ensemble

Cette API REST fournit l'accès sécurisé à la base de données DigitalBank France via Supabase. Elle implémente une architecture de sécurité basée sur Row Level Security (RLS) avec authentification JWT et contrôle d'accès basé sur les rôles (RBAC).

### 1.1 URL de Base

**Base URL:** <https://tzoipnuhurrrqwghjpus.supabase.co/rest/v1>

### 1.2 Authentification

Toutes les requêtes nécessitent deux headers obligatoires :

Header	Description
apikey	Clé publique anon de Supabase
Authorization	Bearer <JWT_TOKEN>

## 2. Rôles et Permissions (RLS)

Le système implémente trois niveaux d'accès via Row Level Security :

Rôle	Permissions	Cas d'usage
admin	Accès complet sur toutes les tables	Gestion système, configuration, supervision
analyst	Lecture globale de toutes les transactions et comptes	Analyse de fraude, reporting, data science
client	Lecture uniquement de ses propres comptes et transactions	Application client, mobile banking

## 3. Endpoints Disponibles

### 3.1 Customers

**GET /customers**

Récupère la liste des clients.

**Permissions:**

- admin: Tous les clients
- analyst: tous les transactions
- client: Uniquement son profil (via email)

### 3.2 Accounts

**GET /accounts**

Récupère les comptes bancaires.

**Permissions:**

- admin: Tous les comptes
- analyst: Tous les comptes (lecture seule)
- client: Uniquement ses comptes

**Exemple de requête:**

<https://tzoipnuhurrrqwgjhpus.supabase.co/rest/v1/accounts>

**Réponse (200 OK) - Client:**

```
[{"account_id":12,"customer_id":9,"account_number":"FR7612345678901234567890134","account_type":"checking","balance":4500.25,"currency":"EUR","opened_at":"2026-01-21T10:54:03.740324","status":"active"}]
```

### 3.3 Transactions

**GET /transactions**

Récupère les transactions bancaires.

**Permissions:**

- admin: Toutes les transactions (30 dans l'exemple)
- analyst: Toutes les transactions (lecture seule)
- client: Uniquement les transactions de ses comptes

**Exemple de requête:**

<https://tzoipnuhurrrqwgjhpus.supabase.co/rest/v1/transactions>

## Réponse (200 OK) - Client:

```
[{"transaction_id":29,"account_id":12,"transaction_type":"payment","amount":-95.00,"currency":"EUR","merchant_name":"Cinema Gaumont","merchant_category":"Entertainment","location":"Bordeaux, France","timestamp":"2026-01-21T10:54:03.740324","status":"completed","is_fraud":false,"fraud_score":null}]
```

## 5. Codes d'Erreur

Code	Description
<b>200</b>	OK - Requête réussie
<b>401</b>	Unauthorized - Token JWT invalide ou expiré
<b>403</b>	Forbidden - Permissions insuffisantes (RLS bloque l'accès)
<b>404</b>	Not Found - Ressource inexistante
<b>500</b>	Internal Server Error - Erreur serveur

## 6. Considérations de Sécurité

### 6.1 Row Level Security (RLS)

RLS est activé sur toutes les tables sensibles. Les policies vérifient automatiquement l'identité de l'utilisateur via le JWT et filtrent les résultats selon son rôle.

### 6.2 Authentification

- Tokens JWT signés avec HS256
- Durée de validité: 3600 secondes (1 heure)
- Support MFA (Multi-Factor Authentication)

### 6.3 Recommandations

- Ne jamais exposer la clé service\_role en production
- Utiliser HTTPS uniquement
- Implémenter rate limiting côté client
- Chiffrer les données sensibles (cartes) avec pgcrypto
- Logger tous les accès dans audit\_logs

## 7. Tests avec Postman

### 7.1 Configuration

**Variables d'environnement à créer:**

- base\_url: <https://tzoipnuhurrrqwghjpus.supabase.co/rest/v1>
- anon\_key: Clé publique anon
- access\_token: JWT obtenu après login

### 7.2 Scénarios de Test

#### Test 1: Client accède à ses comptes

```
GET {{base_url}}/accounts Headers: apikey: {{anon_key}} Authorization:  
Bearer {{access_token}} Résultat attendu: 2 comptes (checking + savings)
```

#### Test 2: Analyst voit toutes les transactions

```
GET {{base_url}}/transactions?limit=10 Résultat attendu: 10 premières  
transactions (tous clients confondus)
```

#### Test 3: Admin filtre les fraudes

```
GET {{base_url}}/transactions?is_fraud=eq.true&select=* Résultat attendu:  
~10 transactions frauduleuses
```

#### Test 4: Client essaie d'accéder aux comptes d'un autre

```
GET {{base_url}}/accounts?customer_id=eq.5 Résultat attendu: 0 résultat (RLS  
bloque l'accès)
```

## 8. Conclusion

Cette API REST fournit un accès sécurisé et granulaire aux données bancaires de DigitalBank France. La protection par Row Level Security garantit que chaque utilisateur ne peut accéder qu'aux données qui lui sont autorisées.