

Documentation API – DigitalBank

1. Base URL

<https://ugyqyzerezxtnxutukcj.supabase.co.supabase.co/rest/v1/>

- Toutes les requêtes doivent inclure l'header **Authorization** avec le JWT :

Authorization: Bearer <ACCESS_TOKEN>

- Content-Type : application/json

2. Endpoints principaux

2.1 GET /customers

Description : Récupère la liste des clients.

Access : admin, customer_service (restreint par RLS pour customer).

Headers :

Authorization: Bearer <ACCESS_TOKEN>

Content-Type: application/json

Query Parameters :

- id=eq.<id> → filtrer un client spécifique
- email=eq.<email> → filtrer par email

Exemple de requête :

GET <https://ugyqyzerezxtnxutukcj.supabase.co.supabase.co/rest/v1/customers?id=eq.123>

Réponse :

```
[  
{  
  "id": 123,  
  "first_name": "Jean",  
  "last_name": "Doe",  
  "email": "jean.doe@example.com",  
  "phone": "+33 123 456 789",  
  "address": "123 Main Street",  
  "city": "Anytown",  
  "state": "CA",  
  "zip": "12345",  
  "country": "US",  
  "created_at": "2023-10-01T12:00:00Z",  
  "updated_at": "2023-10-01T12:00:00Z"}]
```

```
"last_name": "Dupont",
"email": "jean.dupont@email.fr",
"supabase_id": "uuid-123-456"
}
]
```

Codes d'erreur :

- 401 Unauthorized → JWT manquant ou invalide
- 403 Forbidden → rôle non autorisé par RLS

2.2 GET <https://ugyqyzerezxtnxutukcj.supabase.co.supabase.co/rest/v1/accounts>

Description : Liste des comptes bancaires.

Access : admin (tous), customer_service (clients limités), customer (seulement ses comptes)

Query Parameters :

- customer_id=eq.<id> → filtrer par client

Exemple de requête :

GET

https://ugyqyzerezxtnxutukcj.supabase.co.supabase.co/rest/v1/accounts?customer_id=eq.123

Réponse :

```
[
{
  "id": 456,
  "customer_id": 123,
  "account_number": "FR76 1234 5678 9101 1121 3141",
  "balance": 2500.50
}
```

Codes d'erreur :

- 401 Unauthorized
- 403 Forbidden → RLS empêche l'accès aux comptes d'un autre client

2.3 GET /transactions

Description : Récupère les transactions d'un compte.

Access : admin et analyst (tous les comptes), customer_service (clients limités), customer (seulement ses transactions)

Query Parameters :

- account_id=eq.<id> → filtrer par compte
- timestamp=gte.<date> → filtrer à partir d'une date

Exemple de requête :

GET

https://ugyqyzerezxtnxutukcj.supabase.co.supabase.co/rest/v1/transactions?account_id=eq.456

Réponse :

```
[  
 {  
   "id": 789,  
   "account_id": 456,  
   "amount": 120.75,  
   "merchant_category": "Supermarché",  
   "timestamp": "2026-01-22T10:30:00Z"  
 }  
]
```

Codes d'erreur :

- 401 Unauthorized
- 403 Forbidden → RLS

2.4 POST /transactions (optionnel, admin ou API interne)

Description : Crée une nouvelle transaction.

Access : admin uniquement

Body :

```
{  
  "account_id": 456,  
  "amount": 150.00,  
  "merchant_category": "Restaurant",  
  "timestamp": "2026-01-22T12:00:00Z"  
}
```

Réponse :

```
{  
  "id": 790,  
  "account_id": 456,  
  "amount": 150.00,  
  "merchant_category": "Restaurant",  
  "timestamp": "2026-01-22T12:00:00Z"  
}
```

Codes d'erreur :

- 401 Unauthorized
- 403 Forbidden
- 400 Bad Request → données manquantes ou mal formattées

3. Notes générales

- Toutes les tables sont protégées par **RLS**.
- Les rôles définis sont : admin, analyst, customer_service, customer.
- Les requêtes des clients (customer) ne peuvent accéder qu'à leurs propres données.
- Supabase REST utilise les **filtres query params** (eq, gte, lte, like, etc.) pour limiter les données.