

Documentation API – Supabase

1. Présentation générale

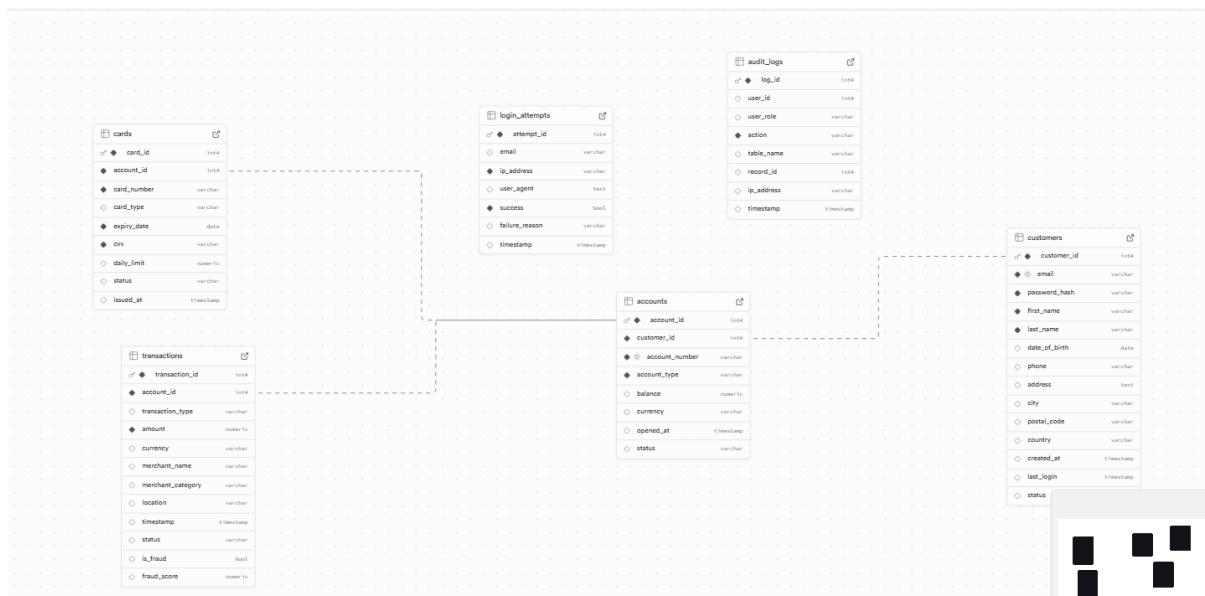
Dans cette phase, une API REST no-code a été mise en place à l'aide de Supabase, qui repose sur une base de données PostgreSQL.

Supabase génère automatiquement des endpoints REST sécurisés à partir des tables de la base de données, tout en intégrant des mécanismes avancés d'authentification et de contrôle d'accès (RLS – Row Level Security).

Cette API permet :

- La gestion des clients
- La consultation des comptes bancaires
- L'accès aux transactions
- L'application de règles de sécurité selon le rôle de l'utilisateur

restauration de la base



2. Authentification et sécurité

2.1 Méthode d'authentification

L'API utilise l'authentification **Email / Mot de passe** fournie par Supabase, avec :

- Support de la **Multi-Factor Authentication (MFA)**
- Gestion automatique des **JWT (JSON Web Tokens)**

Chaque requête à l'API nécessite :

- Un **token d'authentification** valide
- Les headers suivants :

Authorization: Bearer <SUPABASE_JWT_TOKEN>

apikey: <SUPABASE_PUBLIC_API_KEY>

Content-Type: application/json

3. Structure de l'API REST

- L'API est accessible via l'URL de base :

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

- Les endpoints sont générés automatiquement à partir des tables PostgreSQL.

4. Endpoints principaux

4.1 Transactions

- **Récupérer les transactions d'un compte**

Endpoint

GET [https://ugyqyzerezxtnxutukcj.supabase.co /rest/v1/transactions](https://ugyqyzerezxtnxutukcj.supabase.co/rest/v1/transactions)

Exemple

GET [https://ugyqyzerezxtnxutukcj.supabase.co /rest/v1/transactions?select=&account_id=eq.1?](https://ugyqyzerezxtnxutukcj.supabase.co/rest/v1/transactions?select=&account_id=eq.1?)

4.2 Comptes

- **Récupérer les comptes d'un client**

Endpoint

GET [https://ugyqyzerezxtnxutukcj.supabase.co /rest/v1/accounts](https://ugyqyzerezxtnxutukcj.supabase.co/rest/v1/accounts)

Paramètre : customer_id=eq.45678

Exemple

GET https://ugyqyzerezxtnxutukcj.supabase.co /rest/v1/accounts?customer_id=eq.45678

4.3 Clients

- Récupérer les informations d'un client

Endpoint

GET https://ugyqyzerezxtnxutukcj.supabase.co /rest/v1/customers

Exemple

GET https://ugyqyzerezxtnxutukcj.supabase.co /rest/v1/customers?customer_id=eq.45678

5. Sécurité – Row Level Security (RLS)

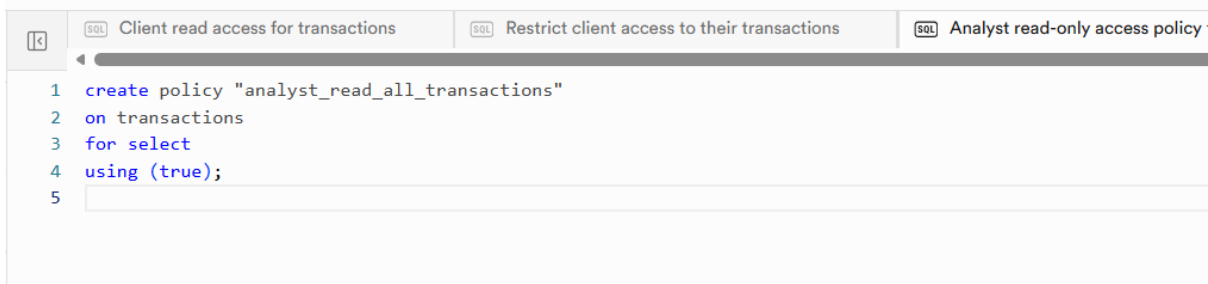
Des **politiques RLS** ont été mises en place afin de garantir la confidentialité des données :

5.1 Clients

- Peuvent consulter uniquement :
 - Leurs propres comptes
 - Leurs propres transactions

5.2 Analystes

- Accès en lecture à l'ensemble des transactions
- Accès aux données clients à des fins d'analyse



```
1 create policy "analyst_read_all_transactions"
2 on transactions
3 for select
4 using (true);
5
```

5.3 Administrateurs

- Accès complet en lecture et écriture
- Gestion des utilisateurs et des données

Ces règles sont implémentées directement au niveau de PostgreSQL via les **policies Supabase**, assurant une sécurité indépendante de l'application cliente.

6. Tests de l'API

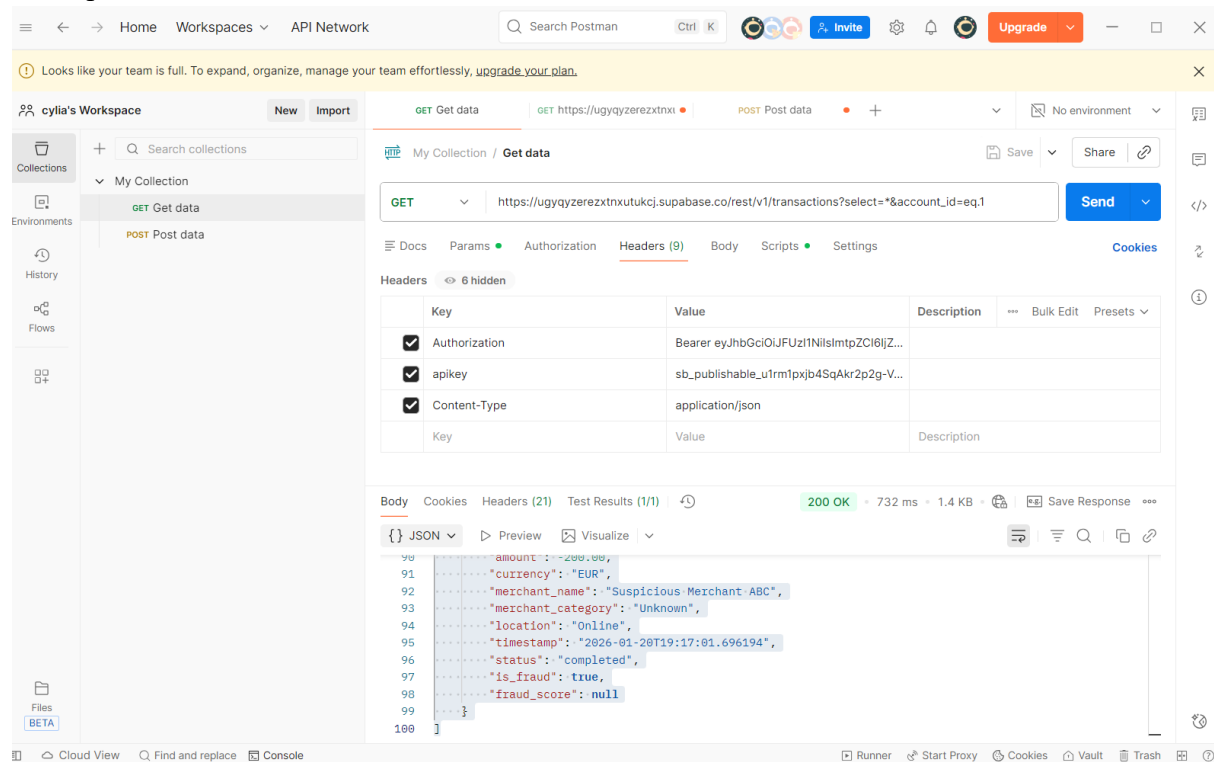
Les endpoints ont été testés avec :

- **Postman**

Creation d'un token

[illegible]

La requête



The screenshot shows the Postman interface with a workspace named 'cylia's Workspace'. A collection named 'My Collection' contains two requests: 'GET Get data' and 'POST Post data'. The 'GET Get data' request is selected, showing the URL: `https://ugyqyzereztuxutkcj.supabase.co/rest/v1/transactions?select=*%&account_id=eq.1`. The request headers are configured with the following values:

| Key | Value | Description |
|---------------|--|-------------|
| Authorization | <code>Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXZW40In0=</code> | |
| apikey | <code>sb_publishable_u1rm1pxjb4SqAkr2p2g-V...</code> | |
| Content-Type | <code>application/json</code> | |

The response body is a JSON array of transaction records, with the following structure:

```
{
  "amount": -200.00,
  "currency": "EUR",
  "merchant_name": "Suspicious Merchant ABC",
  "merchant_category": "Unknown",
  "location": "Online",
  "timestamp": "2026-01-20T19:17:01.696194",
  "status": "completed",
  "is_fraud": true,
  "fraud_score": null
}
```

7. Conclusion

L'utilisation de Supabase a permis de déployer rapidement une **API REST sécurisée, scalable et no-code**, connectée à une base PostgreSQL robuste.