

# Documentation API – Supabase

## 1. Présentation générale

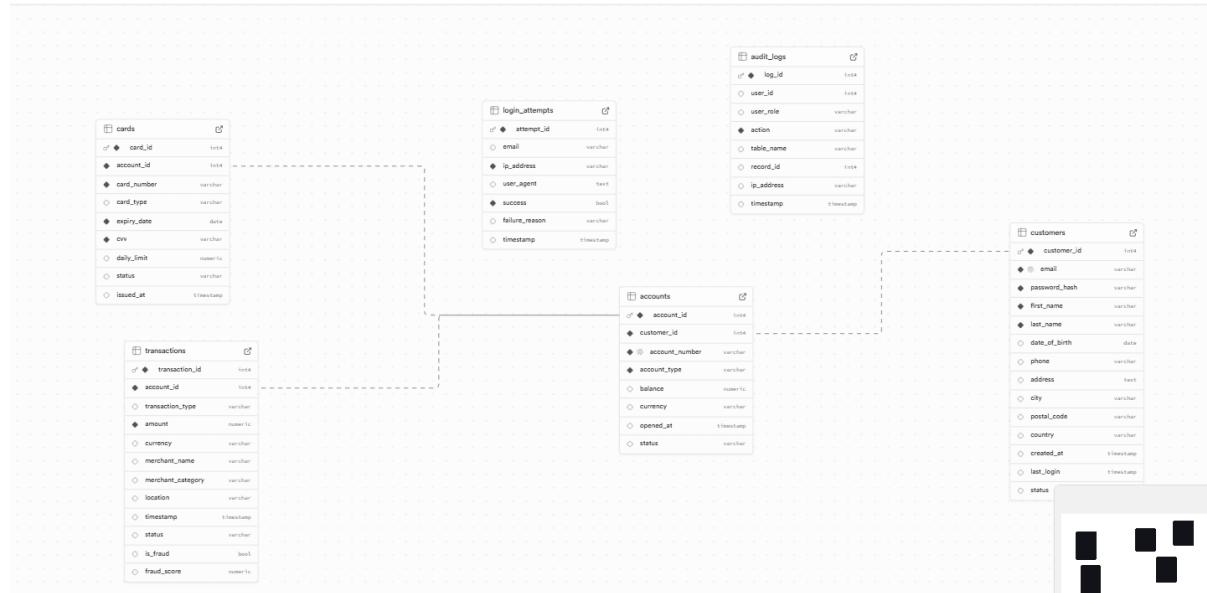
Dans cette phase, une API REST no-code a été mise en place à l'aide de Supabase, qui repose sur une base de données PostgreSQL.

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>

##### 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>

**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)

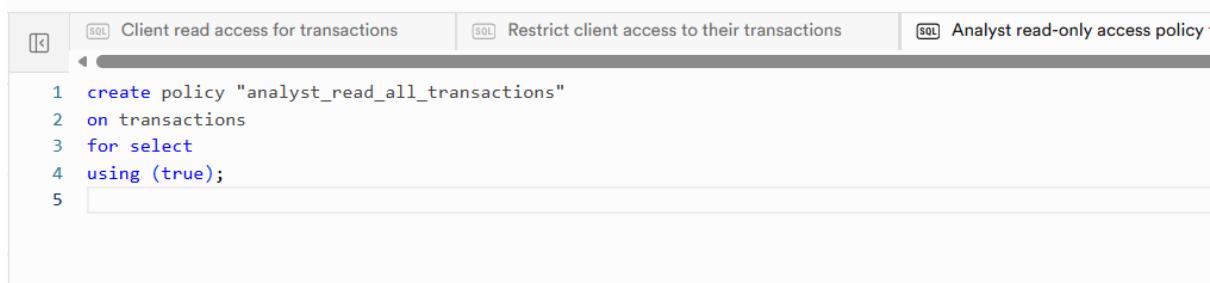
Des **policies RLS** ont été mises en place afin de garantir la confidentialité des données :

### 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);
```

## 5.3 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

The screenshot shows the Postman interface for creating a token. The URL is https://ugyqyzerezxntxutukc.supabase.co/auth/v1/token?grant\_type=password. The Headers tab is selected, showing two checked fields: 'apikey' with value 'sb\_publishable\_u1rm1pxjb4SqAkr2p2g-V...' and 'Content-Type' with value 'application/json'. The Body tab is selected, showing a JSON response with an 'access\_token' field containing a long string of characters.

Key	Value	Description	Bulk Edit	Presets
apikey	sb_publishable_u1rm1pxjb4SqAkr2p2g-V...			
Content-Type	application/json			
Key	Value	Description		

Body:

```
{ } JSON ▾ ▶ Preview ▶ Visualize ▶
```

200 OK 870 ms 1.99 KB ⏺ Save Response

```
1 "access_token": | eyJhbGciOiJFUzI1NiIiImtpZCI6IjZhYzEzMmU2LTm5NdctNDcwNS04ZjM4LTY30WyyNTRlZTg4ZCIsInR5cCI6IkpxVCJ9. eyJpc3MiOiJodHRwczovL3VneXF5emVyZXp4dG54dXR1a2NqLnN1cGFjYXNlNmNvL2F1dGgvdjEiLCJzdWiiOijjNWfhYzEZYSo2NTI0LTRjOGUtODZmYSo4YzYzWU2N2ZjMTM1Lc3hdWq1o1JhdXRoZw50awNndGVKliwiZXhwIjoxNy40TUyNjMSLCJpYXo1oje3Njg5NDkwMzsInvtWlsjioiY2xpZw50OHRLc3Quy29tIiwiGhbmu1o1iILCJchHbfbwVOYWRhdGEiOnsicHJvdmlkZXIoiJlbWFpbCisInByb3ZpZGVycyI6WyJlbWFpbCJdfSwidXN1c19tZXrhZGF0YSI6eyJlbWFpbF92ZXJpZmllZC160HJ1ZX0sInjvbGUjoiJhdXRoZw50awNndGVKliwiYWfsIjoiYWFsMSIsImFtcii6w3sibwV0aG9kIjoicGFzc3dvcmQilCJ0aw1lC3rbxA10jE3Njg5NDkwMz19XSwic2Vzc2lvb19pZCI6IjEwZGRjZGQXLWQwMTgtNDAzNi1hMGZkLTcwY2FlYWFl7TowMCTsTml7Y?Fihh255h91rvTA7mFcr?u9
```

## La requête

The screenshot shows the Postman application interface. On the left, there's a sidebar with sections for Collections, Environments, History, Flows, and Files (BETA). The main area has tabs for 'GET Get data' and 'POST Post data'. Under 'GET Get data', it says 'GET https://ugyqyzerezxnxutukcj.supabase.co/rest/v1/transactions?select=\*&account\_id=eq.1'. Below this, there's a 'Headers' section with three checked items: 'Authorization' (Value: Bearer eyJhbGciOiJFUzI1NlslsImtpZC10jZ...), 'apikey' (Value: sb\_publ...), and 'Content-Type' (Value: application/json). At the bottom, the response is shown as a JSON object:

```
{ } JSON ▾ ▶ Preview ▾ Visualize ▾  
y0 |   "amount": 200.00,  
91 |   "currency": "EUR",  
92 |   "merchant_name": "Suspicious Merchant ABC",  
93 |   "merchant_category": "Unknown",  
94 |   "location": "Online",  
95 |   "timestamp": "2026-01-20T19:17:01.696194",  
96 |   "status": "completed",  
97 |   "is_fraud": true,  
98 |   "fraud_score": null  
99 | }  
100 ]
```

At the bottom right, there are buttons for 'Runner', 'Start Proxy', 'Cookies', 'Vault', 'Trash', and a help icon.

## 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.