



*Resource Software Solution*

# **Flutter Dev On the Job Training**

# HƯỚNG DẪN SỬ DỤNG SUPABASE CHO NGƯỜI MỚI

## 1. Tạo Project Supabase

1. Vào: <https://supabase.com/>
2. Đăng ký (Sign up) hoặc Đăng nhập (Sign in) bằng tài khoản Google/GitHub/email.
3. Chọn **New Project**.
  - Nhập **Project name** (ví dụ: personal-finance-tracker)
  - Nhập **Database Password** (ghi nhớ để dùng kết nối)
  - Chọn **Region** gần bạn nhất.
  - Bấm **Create new project**.
4. Chờ ~2-3 phút, Supabase tạo database xong sẽ tự động vào dashboard.

## 2. Tạo Table trong Supabase (Database > Tables)

### Bảng & Cột đề xuất (Personal Finance Tracker):

#### a) users

(Bạn có thể dùng sẵn Supabase Auth, nhưng vẫn nên có bảng users mở rộng nếu cần)

Column	Type	Constraint	Note
id	uuid	Primary Key, Default: gen_random_uuid()	
email	text	Unique	
name	text		
avatar_url	text	nullable	

#### b) categories

Column	Type	Constraint	Note
id	int8	Primary Key, auto inc	
name	text		(Food, Salary, Shopping...)
type	text		('INCOME'/'EXPENSE')
icon	text	nullable	(icon name or url)

### c) transactions

Column	Type	Constraint	Note
id	int8	Primary Key, auto inc	
user_id	uuid	Foreign Key -> users.id	NOT NULL
category_id	int8	Foreign Key -> categories.id	NOT NULL
type	text		INCOME/'EXPENSE'
amount	float8	NOT NULL	
note	text	nullable	
date	date	NOT NULL	

### d) budgets

Column	Type	Constraint	Note
id	int8	Primary Key, auto inc	
user_id	uuid	Foreign Key -> users.id	NOT NULL
category_id	int8	Foreign Key -> categories.id	NOT NULL
amount	float8	NOT NULL	
month	int4	NOT NULL	(1-12)
year	int4	NOT NULL	(YYYY)

### Cách tạo Table:

- Trong Supabase Dashboard: Chọn **Database** → **Tables** → **New Table**.
- Nhập tên bảng, thêm từng cột với type và constraint đúng như trên.
- Tick “Enable Row Level Security (RLS)” sau khi tạo bảng.

### Tạo Foreign Key:

- Khi thêm cột user\_id (transactions), chọn type là uuid, rồi bấm nút “Add foreign key” → chọn bảng users, cột id.

### Seed Data cho categories

Bạn có thể vào bảng **categories** > **Insert Row** để thêm các dòng ví dụ:

- name: Food, type: EXPENSE, icon: food
- name: Shopping, type: EXPENSE, icon: shopping\_cart
- name: Salary, type: INCOME, icon: money
- name: Freelance, type: INCOME, icon: work

### 3. Tạo Policy cho bảng (Bảo mật truy cập đúng user)

#### Ví dụ cho bảng transactions:

- Chọn bảng → Policies → New Policy.
- Policy Name: "Users can CRUD own transactions"
- Action: All
- Expression: `user_id = auth.uid()`
- Bấm Save.

Làm tương tự với bảng **budgets**, **transactions**.

### 4. Lấy API URL, API Key cho Postman

- Vào **Project Settings > API**
  - **Project URL:** (ví dụ: `https://your-project-id.supabase.co`)
  - **Anon Public API Key:** (dùng cho Postman)
- Header cần dùng khi gọi:
  - `apikey: [API_KEY]`
  - `Authorization: Bearer [API_KEY]` (hoặc Bearer JWT token nếu cần bảo mật từng user)

### 5. Tạo Postman Collection (CRUD) cho từng bảng

#### A. Example: transactions

##### 1. GET all transactions

- **Method:** GET
- **URL:** `https://your-project-id.supabase.co/rest/v1/transactions?user_id=eq.{USER_ID}`
- **Headers:**
  - `apikey: [API_KEY]`
  - `Authorization: Bearer [API_KEY]`
  - `Content-Type: application/json`

##### 2. GET by id

- **Method:** GET
- **URL:** `.../transactions?id=eq.1` (hoặc filter khác)

### 3. INSERT (Create)

- **Method:** POST
- **URL:** .../transactions
- **Headers:** như trên
- **Body (raw/JSON):**

```
{  
  "user_id": "uuid-cua-ban",  
  "category_id": 1,  
  "type": "EXPENSE",  
  "amount": 500,  
  "note": "Ăn sáng",  
  "date": "2024-07-14"  
}
```

### 4. UPDATE

- **Method:** PATCH
- **URL:** .../transactions?id=eq.1
- **Body (raw/JSON):**

```
{  
  "amount": 600,  
  "note": "Ăn trưa"  
}
```

### 5. DELETE

- **Method:** DELETE
- **URL:** .../transactions?id=eq.1
- **Headers:** như trên

B. Mẫu Collection cho các bảng khác (chỉ đổi endpoint):

Bảng	Endpoint
users	/rest/v1/users
categories	/rest/v1/categories
budgets	/rest/v1/budgets
transactions	/rest/v1/transactions

## 6. Export & Import Postman Collection

- Sau khi tạo đủ request trong Postman, bấm chuột phải vào Collection → Export → Chọn phiên bản 2.1 (recommended)
- Gửi file .json cho team, Fresher chỉ cần Import là dùng ngay.

## 7. Checklist Postman Collection cho Fresher

- GET all (từng bảng)
- GET by id
- POST/INSERT (từng bảng)
- PATCH/UPDATE (từng bảng)
- DELETE (từng bảng)
- Test request trả về đúng dữ liệu, báo lỗi đúng policy

Đây là **file mẫu Postman Collection (JSON)** dùng để test **CRUD** cho 4 bảng chính: users, categories, transactions, budgets trên Supabase REST API.

Bạn chỉ cần:

- Import file này vào Postman
- Thay đổi giá trị `{{SUPABASE_URL}}`, `{{API_KEY}}` và các biến body theo project của bạn
- Dùng ngay để test!

**File Postman Collection (supabase\_finance\_tracker.postman\_collection.json)** cho Supabase Personal Finance Tracker, bạn có thể tải về và import trực tiếp vào Postman. Chỉ cần thay biến `SUPABASE_URL` và `API_KEY` theo project của bạn, rồi dùng ngay nhé!