**What is Supabase?**

**Supabase is an open-source alternative to Firebase** that provides backend-as-a-service using PostgreSQL instead of NoSQL. It's essentially a suite of tools that helps developers build applications faster by handling the backend infrastructure.

**Core Components**

**1. PostgreSQL Database**

* **Fully relational database** (unlike Firebase's NoSQL)
* **SQL support** - you can write complex queries
* **Row-level security** for data protection
* **Extensions** for additional functionality

**2. Auto-generated APIs**

javascript

// REST API example

// GET all users

fetch('https://your-project.supabase.co/rest/v1/users', {

headers: {

'apikey': 'your-anon-key'

}

})

// Realtime example

const subscription = supabase

.from('users')

.on('UPDATE', payload => {

console.log('User updated!', payload)

})

.subscribe()

**3. Authentication**

* Email/password auth
* Social logins (Google, GitHub, etc.)
* Magic links
* Phone authentication
* JWT tokens

**4. Storage**

* File storage and CDN
* Image transformations
* Access control

**5. Realtime Subscriptions**

* Listen to database changes in real-time
* Perfect for chat apps, live dashboards

**Key Features**

**🔥 Instant Setup**

bash

# Start locally in minutes

docker-compose up -d

# Or use cloud version

**🔄 Real-time by Default**

javascript

// Listen to database changes

supabase

.from('todos')

.on('INSERT', payload => {

console.log('New todo!', payload.new)

})

.subscribe()

**🔐 Built-in Security**

sql

-- Row Level Security policies

CREATE POLICY "Users can view own profile"

ON profiles FOR SELECT

USING (auth.uid() = id);

**📱 Multiple Client Libraries**

javascript

// JavaScript

const { data, error } = await supabase

.from('countries')

.select('name')

// Python

data = supabase.table('countries').select('name').execute()

// Flutter

final data = await supabase

.from('countries')

.select('name');

**Supabase vs Firebase**

| Aspect | Supabase | Firebase |
| --- | --- | --- |
| **Database** | PostgreSQL (SQL) | Firestore (NoSQL) |
| **API** | REST + GraphQL | Proprietary SDKs |
| **Realtime** | PostgreSQL replication | Proprietary |
| **Pricing** | Postgres-based | Usage-based |
| **Open Source** | ✅ Yes | ❌ No |

**Common Use Cases**

**1. Web Applications**

javascript

// Full-stack app example

const { data: user } = await supabase.auth.getUser()

const { data: posts } = await supabase

.from('posts')

.select(`

\*,

profiles (username, avatar\_url)

`)

.eq('user\_id', user.id)

**2. Mobile Apps**

dart

// Flutter example

await supabase

.from('todos')

.insert({'task': 'Buy milk', 'user\_id': userId});

**3. Real-time Dashboards**

javascript

// Live analytics

supabase

.from('analytics')

.on('INSERT', payload => {

updateChart(payload.new)

})

.subscribe()

**Architecture Overview**

text

Your Frontend App

↓

Supabase Client Library

↓

Supabase API Layer

↓

PostgreSQL Database ← Realtime ← Auth ← Storage

**Why Developers Love Supabase**

1. **Uses PostgreSQL** - familiar, powerful, SQL-standard
2. **Open Source** - self-hostable, transparent
3. **Real-time** - built-in, no extra setup
4. **Great DX** - excellent documentation and tools
5. **Generous Free Tier** - perfect for startups and side projects

**Getting Started Flow**

1. **Create Project** (cloud or local)
2. **Create Tables** in the SQL editor
3. **Enable Auth** if needed
4. **Connect Client** to your frontend
5. **Build Features** using the APIs

**In essence: Supabase takes the powerful PostgreSQL database and wraps it with easy-to-use APIs, authentication, and real-time capabilities, making backend development much faster and more accessible.**

**Question: What's the Difference Between Supabase and Firebase?**