**Guide for Setting up Supabase for Hobbly**

📋 **Contents**

* Creating a Supabase Project
* Initializing the Database
* Setting up Authentication
* Getting API Keys
* Configuring the Project Environment
* Verifying Setup
* Test Data

**1. Creating a Supabase Project**

**Step 1.1: Registration on Supabase**

* Go to [https://supabase.com](https://supabase.com/)
* Click **"Start your project"**
* Log in with GitHub or create an account

**Step 1.2: Creating a New Project**

* Click **"New Project"**
* Fill out the form:
  + **Name**: hobbly (or any other name)
  + **Database Password**: Create a strong password (save it!)
  + **Region**: Choose the nearest region (e.g., West EU (Ireland))
  + **Pricing Plan**: Free tier is sufficient for development
* Click **"Create new project"**
* Wait 1–2 minutes while the project initializes

**2. Initializing the Database**

**Step 2.1: Opening the SQL Editor**

* In the left menu, select **"SQL Editor"**
* Click **"New query"**

**Step 2.2: Running the Initialization Script**

* Open the supabase\_init.sql file from the project root
* Copy its contents
* Paste into the Supabase SQL Editor
* Click **"Run"** (or Ctrl+Enter)
* Wait for the success message

**Step 2.3: Verifying Table Creation**

* Go to **"Table Editor"** in the left menu
* Ensure the following tables are created:
  + categories – Activity categories
  + tags – Tags
  + activities – Activities / Events
  + activity\_tags – Linking activities and tags
  + user\_profiles – User profiles

**3. Setting up Authentication**

**Step 3.1: Email Authentication Setup**

* Go to **"Authentication" → "Providers"**
* Ensure **"Email"** is enabled
* Configure:
  + Enable Email Confirmations: OFF (for testing)
  + Minimum password length: 8

**Step 3.2: Configuring Email Templates (optional)**

* Go to **"Authentication" → "Email Templates"**
* You can customize templates for:
  + Confirm signup
  + Reset password
  + Magic Link

**Step 3.3: Redirect URL Configuration**

* Go to **"Authentication" → "URL Configuration"**
* Add the following to Redirect URLs:
  + http://localhost:3000/\*
  + http://localhost:3000
* For production, add your domain

**4. Getting API Keys**

**Step 4.1: Copying Keys**

* Go to **"Settings" → "API"**
* Copy the following values:
  + **Project URL**: https://your-project.supabase.co
  + **anon public**: eyJhbGc... (long key)
  + **service\_role** (DO NOT use on client side!)

**Step 4.2: Saving Keys**

* Store the keys securely. They will be needed for project setup.

**5. Configuring the Project Environment**

**Step 5.1: Creating .env File**

* In the project root, create a .env file (if missing)
* Copy contents from .env.example
* Fill in values:

# Supabase Configuration

REACT\_APP\_SUPABASE\_URL=https://your-project-id.supabase.co

REACT\_APP\_SUPABASE\_ANON\_KEY=eyJhbGc...your\_anon\_key

# Optional Settings

PORT=3000

REACT\_APP\_ENV=development

**Step 5.2: Checking .gitignore**  
Ensure .env is included in .gitignore:

# env files

.env

.env.local

.env.production

**6. Verifying Setup**

**Step 6.1: Starting the Project**

npm start

**Step 6.2: Checking Console**

* Open browser dev console (F12)
* There should be no missing environment variable errors
* If errors appear → check .env values

**Step 6.3: Test Registration**

* Try registering a new user
* Check in Supabase Dashboard → **Authentication → Users**
* New user should appear in the list

**7. Test Data**

**Step 7.1: Creating a Test Admin**  
Run in SQL editor:

-- Create test admin

INSERT INTO auth.users (

id,

email,

encrypted\_password,

email\_confirmed\_at,

raw\_user\_meta\_data,

created\_at,

updated\_at

) VALUES (

gen\_random\_uuid(),

'admin@hobbly.fi',

crypt('Admin123!', gen\_salt('bf')),

now(),

'{"role": "admin", "fullName": "Admin User"}'::jsonb,

now(),

now()

);

**Step 7.2: Creating Test Activities**

-- Get admin ID

WITH admin\_user AS (

SELECT id FROM auth.users WHERE email = 'admin@hobbly.fi' LIMIT 1

),

-- Get category ID

sport\_category AS (

SELECT id FROM categories WHERE name = 'Sports and Physical Activity' LIMIT 1

)

-- Create test activity

INSERT INTO activities (

title,

description,

type,

category\_id,

location,

address,

price,

user\_id,

contact\_email,

contact\_phone

)

SELECT

'Children’s Football Club',

'We invite children aged 7–12 to our football club. Professional coaches, modern equipment, friendly atmosphere.',

'club',

sport\_category.id,

'Helsinki',

'Sports Center, Mannerheimintie 15',

25.00,

admin\_user.id,

'football@hobbly.fi',

'+358 40 123 4567'

FROM admin\_user, sport\_category;

**Step 7.3: Adding Tags to Activity**

-- Link activity with tags

WITH activity AS (

SELECT id FROM activities WHERE title = 'Children’s Football Club' LIMIT 1

),

tags\_to\_add AS (

SELECT id FROM tags WHERE name IN ('Beginner-friendly', 'Kids-friendly', 'Recurring Event')

)

INSERT INTO activity\_tags (activity\_id, tag\_id)

SELECT activity.id, tags\_to\_add.id

FROM activity, tags\_to\_add;

**📝 Important Notes**

**Security**

* NEVER use the service\_role key on the client
* anon key is safe for browser usage
* Always enable Row Level Security (RLS)

**Free Plan Limits**

* 500MB database
* 1GB file storage
* 2GB transfer
* 50,000 monthly active users

**Useful Links**

* Supabase Docs
* PostgREST Operators
* Supabase JavaScript Client

**🆘 Troubleshooting**

* **Error: "Missing Supabase environment variables"**  
  → Check .env file exists and is correct
* **Error: "Invalid API key"**  
  → Make sure correct anon key is copied
* **Error: "relation does not exist"**  
  → Re-run supabase\_init.sql script
* **Error during user registration**  
  → Check Authentication → Email Provider settings

**✅ Readiness Checklist**

* Supabase project created
* SQL init script executed
* Required tables created
* Authentication configured
* API keys copied
* .env file created and configured
* Project runs successfully
* User registration works
* Test data created

After completing all steps, your Supabase backend is ready to use! 🎉

In this instruction, the **backend is actually created and configured at stages 2–7**:

* **Stage 2. Database Initialization** → tables, relations, and data structure are created. This is the foundation of the backend.
* **Stage 3. Authentication Setup** → login/registration logic is added.
* **Stage 4. Getting API Keys** → defines how the frontend will communicate with the backend.
* **Stage 5. Project Environment Setup (.env)** → connects the backend to the React app.
* **Stages 6–7. Verification and Test Data** → filling and testing the backend.

👉 So, the entire backend (database + API + authentication) is built through Supabase, starting from **Database Initialization (Stage 2)**.

**Инструкция по настройке Supabase для Hobbly**

**📋 Содержание**

1. [Создание проекта Supabase](https://github.com/mariacedersten/Ohjelmistokehitysprojekti-2/blob/main/hobbly-app/SUPABASE_SETUP.md#1-%D1%81%D0%BE%D0%B7%D0%B4%D0%B0%D0%BD%D0%B8%D0%B5-%D0%BF%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D0%B0-supabase)
2. [Инициализация базы данных](https://github.com/mariacedersten/Ohjelmistokehitysprojekti-2/blob/main/hobbly-app/SUPABASE_SETUP.md#2-%D0%B8%D0%BD%D0%B8%D1%86%D0%B8%D0%B0%D0%BB%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F-%D0%B1%D0%B0%D0%B7%D1%8B-%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85)
3. [Настройка аутентификации](https://github.com/mariacedersten/Ohjelmistokehitysprojekti-2/blob/main/hobbly-app/SUPABASE_SETUP.md#3-%D0%BD%D0%B0%D1%81%D1%82%D1%80%D0%BE%D0%B9%D0%BA%D0%B0-%D0%B0%D1%83%D1%82%D0%B5%D0%BD%D1%82%D0%B8%D1%84%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D0%B8)
4. [Получение ключей API](https://github.com/mariacedersten/Ohjelmistokehitysprojekti-2/blob/main/hobbly-app/SUPABASE_SETUP.md#4-%D0%BF%D0%BE%D0%BB%D1%83%D1%87%D0%B5%D0%BD%D0%B8%D0%B5-%D0%BA%D0%BB%D1%8E%D1%87%D0%B5%D0%B9-api)
5. [Настройка окружения проекта](https://github.com/mariacedersten/Ohjelmistokehitysprojekti-2/blob/main/hobbly-app/SUPABASE_SETUP.md#5-%D0%BD%D0%B0%D1%81%D1%82%D1%80%D0%BE%D0%B9%D0%BA%D0%B0-%D0%BE%D0%BA%D1%80%D1%83%D0%B6%D0%B5%D0%BD%D0%B8%D1%8F-%D0%BF%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D0%B0)
6. [Проверка настройки](https://github.com/mariacedersten/Ohjelmistokehitysprojekti-2/blob/main/hobbly-app/SUPABASE_SETUP.md#6-%D0%BF%D1%80%D0%BE%D0%B2%D0%B5%D1%80%D0%BA%D0%B0-%D0%BD%D0%B0%D1%81%D1%82%D1%80%D0%BE%D0%B9%D0%BA%D0%B8)
7. [Тестовые данные](https://github.com/mariacedersten/Ohjelmistokehitysprojekti-2/blob/main/hobbly-app/SUPABASE_SETUP.md#7-%D1%82%D0%B5%D1%81%D1%82%D0%BE%D0%B2%D1%8B%D0%B5-%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D0%B5)

**1. Создание проекта Supabase**

**Шаг 1.1: Регистрация на Supabase**

1. Перейдите на [https://supabase.com](https://supabase.com/)
2. Нажмите "Start your project"
3. Войдите через GitHub или создайте аккаунт

**Шаг 1.2: Создание нового проекта**

1. Нажмите "New Project"
2. Заполните форму:
   * **Name**: hobbly (или любое другое название)
   * **Database Password**: Создайте надежный пароль (сохраните его!)
   * **Region**: Выберите ближайший регион (например, West EU (Ireland))
   * **Pricing Plan**: Free tier подойдет для разработки
3. Нажмите "Create new project"
4. Подождите 1-2 минуты пока проект инициализируется

**2. Инициализация базы данных**

**Шаг 2.1: Открытие SQL редактора**

1. В левом меню выберите "SQL Editor"
2. Нажмите "New query"

**Шаг 2.2: Выполнение скрипта инициализации**

1. Откройте файл supabase\_init.sql из корня проекта
2. Скопируйте весь содержимое файла
3. Вставьте в SQL редактор Supabase
4. Нажмите "Run" (или Ctrl+Enter)
5. Дождитесь сообщения об успешном выполнении

**Шаг 2.3: Проверка создания таблиц**

1. Перейдите в "Table Editor" в левом меню
2. Убедитесь, что созданы следующие таблицы:
   * categories - Категории активностей
   * tags - Теги
   * activities - Активности/мероприятия
   * activity\_tags - Связь активностей и тегов
   * user\_profiles - Профили пользователей

**3. Настройка аутентификации**

**Шаг 3.1: Настройка email аутентификации**

1. Перейдите в "Authentication" → "Providers"
2. Убедитесь, что "Email" включен
3. Настройте параметры:
   * **Enable Email Confirmations**: OFF (для тестирования)
   * **Minimum password length**: 8

**Шаг 3.2: Настройка шаблонов писем (опционально)**

1. Перейдите в "Authentication" → "Email Templates"
2. Можете настроить шаблоны для:
   * Confirm signup
   * Reset password
   * Magic Link

**Шаг 3.3: Настройка URL перенаправления**

1. Перейдите в "Authentication" → "URL Configuration"
2. Добавьте в "Redirect URLs":
3. http://localhost:3000/\*
4. http://localhost:3000
5. Для продакшена добавьте ваш домен

**4. Получение ключей API**

**Шаг 4.1: Копирование ключей**

1. Перейдите в "Settings" → "API"
2. Скопируйте следующие значения:
   * **Project URL**: https://your-project.supabase.co
   * **anon public**: eyJhbGc... (длинный ключ)
   * **service\_role** (НЕ используйте на клиенте!)

**Шаг 4.2: Сохранение ключей**

Сохраните ключи в безопасном месте. Они понадобятся для настройки проекта.

**5. Настройка окружения проекта**

**Шаг 5.1: Создание .env файла**

1. В корне проекта создайте файл .env (если его нет)
2. Скопируйте содержимое из .env.example
3. Заполните значения:

# Supabase Configuration

REACT\_APP\_SUPABASE\_URL=https://your-project-id.supabase.co

REACT\_APP\_SUPABASE\_ANON\_KEY=eyJhbGc...ваш\_anon\_ключ

# Опциональные настройки

PORT=3000

REACT\_APP\_ENV=development

**Шаг 5.2: Проверка .gitignore**

Убедитесь, что .env добавлен в .gitignore:

# env files

.env

.env.local

.env.production

**6. Проверка настройки**

**Шаг 6.1: Запуск проекта**

npm start

**Шаг 6.2: Проверка консоли**

1. Откройте браузер и консоль разработчика (F12)
2. Не должно быть ошибок о отсутствующих переменных окружения
3. Если есть ошибки - проверьте правильность ключей в .env

**Шаг 6.3: Тестовая регистрация**

1. Попробуйте зарегистрировать нового пользователя
2. Проверьте в Supabase Dashboard → Authentication → Users
3. Новый пользователь должен появиться в списке

**7. Тестовые данные**

**Шаг 7.1: Создание тестового администратора**

В SQL редакторе выполните:

-- Создание тестового администратора

INSERT INTO auth.users (

id,

email,

encrypted\_password,

email\_confirmed\_at,

raw\_user\_meta\_data,

created\_at,

updated\_at

) VALUES (

gen\_random\_uuid(),

'admin@hobbly.fi',

crypt('Admin123!', gen\_salt('bf')),

now(),

'{"role": "admin", "fullName": "Admin User"}'::jsonb,

now(),

now()

);

**Шаг 7.2: Создание тестовых активностей**

-- Получаем ID администратора

WITH admin\_user AS (

SELECT id FROM auth.users WHERE email = 'admin@hobbly.fi' LIMIT 1

),

-- Получаем ID категории

sport\_category AS (

SELECT id FROM categories WHERE name = 'Спорт и физическая активность' LIMIT 1

)

-- Создаем тестовую активность

INSERT INTO activities (

title,

description,

type,

category\_id,

location,

address,

price,

user\_id,

contact\_email,

contact\_phone

)

SELECT

'Футбольная секция для детей',

'Приглашаем детей 7-12 лет в нашу футбольную секцию. Профессиональные тренеры, современное оборудование, дружеская атмосфера.',

'club',

sport\_category.id,

'Хельсинки',

'Спортивный центр, ул. Маннергейма 15',

25.00,

admin\_user.id,

'football@hobbly.fi',

'+358 40 123 4567'

FROM admin\_user, sport\_category;

**Шаг 7.3: Добавление тегов к активности**

-- Связываем активность с тегами

WITH activity AS (

SELECT id FROM activities WHERE title = 'Футбольная секция для детей' LIMIT 1

),

tags\_to\_add AS (

SELECT id FROM tags WHERE name IN ('Подходит для начинающих', 'Подходит для детей', 'Постоянное событие')

)

INSERT INTO activity\_tags (activity\_id, tag\_id)

SELECT activity.id, tags\_to\_add.id

FROM activity, tags\_to\_add;

**📝 Важные заметки**

**Безопасность**

* **НИКОГДА** не используйте service\_role ключ на клиенте
* anon ключ безопасен для использования в браузере
* Всегда используйте Row Level Security (RLS) для защиты данных

**Лимиты бесплатного плана**

* 500MB база данных
* 1GB хранилище файлов
* 2GB трансфер
* 50,000 активных пользователей в месяц

**Полезные ссылки**

* [Документация Supabase](https://supabase.com/docs)
* [PostgREST операторы](https://postgrest.org/en/stable/api.html#operators)
* [Supabase JavaScript Client](https://supabase.com/docs/reference/javascript/introduction)

**🆘 Решение проблем**

**Ошибка: "Missing Supabase environment variables"**

**Решение**: Проверьте наличие и правильность .env файла

**Ошибка: "Invalid API key"**

**Решение**: Убедитесь, что скопировали правильный anon ключ

**Ошибка: "relation does not exist"**

**Решение**: Выполните supabase\_init.sql скрипт заново

**Ошибка при регистрации пользователя**

**Решение**: Проверьте настройки Email провайдера в Authentication

**✅ Чеклист готовности**

* Создан проект в Supabase
* Выполнен SQL скрипт инициализации
* Созданы все необходимые таблицы
* Настроена аутентификация
* Скопированы API ключи
* Создан и настроен .env файл
* Проект успешно запускается
* Работает регистрация пользователей
* Созданы тестовые данные

После выполнения всех шагов ваш Supabase backend готов к работе! 🎉

В этой инструкции **backend фактически создаётся и настраивается на этапах 2–7**:

* **Этап 2. Инициализация базы данных** → создаются таблицы, связи, структура данных. Это основа backend.
* **Этап 3. Настройка аутентификации** → добавляется логика входа/регистрации пользователей.
* **Этап 4. Получение ключей API** → то, как frontend будет обращаться к backend.
* **Этап 5. Настройка окружения проекта (.env)** → подключение backend к React-приложению.
* **Этап 6–7. Проверка и тестовые данные** → наполнение и тестирование backend.

👉 То есть весь backend (база данных + API + аутентификация) строится через Supabase, начиная с **инициализации базы данных (Этап 2)**.