**Руководство программиста для API Supabase**

**Содержание**

1. **Введение**
   * 1.1. Назначение руководства
   * 1.2. Что такое Supabase?
   * 1.3. Ключевые концепции Supabase
2. **Настройка проекта Supabase**
   * 2.1. Создание аккаунта Supabase
   * 2.2. Создание нового проекта
   * 2.3. Получение URL и ключа API
3. **Авторизация**
   * 3.1. Клиентская библиотека Supabase для .NET
   * 3.2. Регистрация пользователя
   * 3.3. Авторизация пользователя
   * 3.4. Выход из аккаунта
   * 3.5. Управление сессиями
4. **Работа с базой данных (Postgrest)**
   * 4.1. Описание Postgrest API
   * 4.2. Чтение данных (SELECT)
   * 4.3. Создание данных (INSERT)
   * 4.4. Обновление данных (UPDATE)
   * 4.5. Удаление данных (DELETE)
   * 4.6. Фильтрация и сортировка
   * 4.7. Реальное время (Realtime)
5. **Хранилище файлов (Storage)**
   * 5.1. Загрузка файлов (Upload)
   * 5.2. Получение URL файлов (Get URL)
   * 5.3. Удаление файлов (Delete)
   * 5.4. Управление политиками доступа
6. **Функции (Functions)**
   * 6.1. Описание Edge Functions
   * 6.2. Вызов функций из приложения
   * 6.3. Передача данных в функции
7. **Устранение неполадок**
   * 7.1. Общие проблемы и решения
   * 7.2. Получение помощи и обратная связь
8. **Приложение**
   * 8.1. Глоссарий

**1. Введение**

* **1.1. Назначение руководства**

Это руководство предназначено для программистов, желающих использовать API Supabase для разработки приложений. Оно содержит инструкции и примеры кода на C# для основных операций с Supabase.

* **1.2. Что такое Supabase?**

Supabase - это open-source альтернатива Firebase. Она предоставляет набор инструментов и сервисов для упрощения разработки современных веб- и мобильных приложений.

* **1.3. Ключевые концепции Supabase**
  + **Аутентификация (Auth):** Управление пользователями и их доступом к приложению.
  + **База данных (Postgrest):** Автоматически генерируемый RESTful API для PostgreSQL базы данных.
  + **Хранилище файлов (Storage):** Хранилище для файлов, интегрированное с системой аутентификации и политиками доступа.
  + **Функции (Functions):** Бессерверные функции, выполняемые на edge-серверах.
  + **Realtime:** Система реального времени для отслеживания изменений данных.

**2. Настройка проекта Supabase**

* **2.1. Создание аккаунта Supabase**
  1. Перейдите на сайт [https://supabase.com/](https://www.google.com/url?sa=E&q=https%3A%2F%2Fsupabase.com%2F)
  2. Нажмите кнопку "Sign Up" и создайте учетную запись, используя свой email или аккаунт Google/GitHub.
* **2.2. Создание нового проекта**
  1. Войдите в свою учетную запись Supabase.
  2. Нажмите кнопку "New Project".
  3. Выберите организацию, регион и придумайте название для своего проекта.
  4. Дождитесь завершения процесса создания проекта (это может занять несколько минут).
* **2.3. Получение URL и ключа API**
  1. В панели управления вашим проектом перейдите в раздел "Settings" -> "API".
  2. Скопируйте значения "Project URL" и "anon key". Они понадобятся для инициализации клиента Supabase в вашем приложении.

**3. Авторизация**

* **3.1. Клиентская библиотека Supabase для .NET**

Для взаимодействия с API Supabase из C# необходимо использовать клиентскую библиотеку Supabase для .NET. Ее можно установить из NuGet:

Install-Package Supabase

* **3.2. Регистрация пользователя**
* using Supabase;
* // ...
* string supabaseUrl = "YOUR\_SUPABASE\_URL";
* string supabaseKey = "YOUR\_SUPABASE\_ANON\_KEY";
* Supabase.Client supabase = new Supabase.Client(supabaseUrl, supabaseKey, new SupabaseOptions());
* await supabase.InitializeAsync();
* try
* {
* var authResponse = await supabase.Auth.SignUp(email, password);
* if (authResponse.User != null)
* {
* Console.WriteLine($"User registered: {authResponse.User.Email}");
* }
* else
* {
* Console.WriteLine("Registration failed.");
* }
* }
* catch (Exception ex)
* {
* Console.WriteLine($"Error during registration: {ex.Message}");
* }

* **3.3. Авторизация пользователя**
* try
* {
* var authResponse = await supabase.Auth.SignIn(email, password);
* if (authResponse.User != null)
* {
* Console.WriteLine($"User signed in: {authResponse.User.Email}");
* }
* else
* {
* Console.WriteLine("Sign-in failed.");
* }
* }
* catch (Exception ex)
* {
* Console.WriteLine($"Error during sign-in: {ex.Message}");
* }

* **3.4. Выход из аккаунта**
* await supabase.Auth.SignOut();
* Console.WriteLine("User signed out.");

* **3.5. Управление сессиями**

//TODO (В следующих версиях)

**4. Работа с базой данных (Postgrest)**

* **4.1. Описание Postgrest API**

Postgrest позволяет вам работать с PostgreSQL базой данных Supabase, используя RESTful API. Каждая таблица в вашей базе данных становится доступной через API.

* **4.2. Чтение данных (SELECT)**
* using MegaSklad.Models; // Предполагаем, что у вас есть модель "Товары"
* try
* {
* var response = await supabase
* .From<Товары>()
* .Select("\*")
* .Get();
* if (response.Models != null)
* {
* foreach (var tovar in response.Models)
* {
* Console.WriteLine($"Товар: {tovar.название\_товара}, Цена: {tovar.цена}");
* }
* }
* else
* {
* Console.WriteLine("No data found.");
* }
* }
* catch (Exception ex)
* {
* Console.WriteLine($"Error fetching data: {ex.Message}");
* }

* **4.3. Создание данных (INSERT)**
* try
* {
* var newTovar = new Товары
* {
* название\_товара = "Новый товар",
* артикул\_товара = "ART123",
* цена = 100.00m
* };
* var response = await supabase
* .From<Товары>()
* .Insert(newTovar);
* if (response.Models != null)
* {
* Console.WriteLine($"New tovar added with ID: {response.Models.First().id}");
* }
* else
* {
* Console.WriteLine("Failed to add new tovar.");
* }
* }
* catch (Exception ex)
* {
* Console.WriteLine($"Error adding data: {ex.Message}");
* }

* **4.4. Обновление данных (UPDATE)**
* try
* {
* var tovarIdToUpdate = Guid.Parse("YOUR\_TOVAR\_ID"); // Замените на реальный ID
* var response = await supabase
* .From<Товары>()
* .Where(x => x.id == tovarIdToUpdate)
* .Set(x => x.цена, 120.00m)
* .Update();
* if (response.Models != null)
* {
* Console.WriteLine($"Tovar updated successfully.");
* }
* else
* {
* Console.WriteLine($"Failed to update tovar.");
* }
* }
* catch (Exception ex)
* {
* Console.WriteLine($"Error updating data: {ex.Message}");
* }

* **4.5. Удаление данных (DELETE)**
* try
* {
* var tovarIdToDelete = Guid.Parse("YOUR\_TOVAR\_ID"); // Замените на реальный ID
* var response = await supabase
* .From<Товары>()
* .Where(x => x.id == tovarIdToDelete)
* .Delete();
* if (response.ResponseMessage.IsSuccessStatusCode)
* {
* Console.WriteLine($"Tovar deleted successfully.");
* }
* else
* {
* Console.WriteLine($"Failed to delete tovar.");
* }
* }
* catch (Exception ex)
* {
* Console.WriteLine($"Error deleting data: {ex.Message}");
* }

* **4.6. Фильтрация и сортировка**
* // Фильтрация по условию
* var response = await supabase
* .From<Товары>()
* .Where(x => x.цена > 50.00m)
* .Get();
* // Сортировка по цене (по возрастанию)
* var response = await supabase
* .From<Товары>()
* .Order(x => x.цена)
* .Get();
* // Сортировка по цене (по убыванию)
* var response = await supabase
* .From<Товары>()
* .Order(x => x.цена, ascending: false)
* .Get();

Use code [with caution](https://support.google.com/legal/answer/13505487).C#

* **4.7. Реальное время (Realtime)**  
  //TODO (В следующих версиях)

**5. Хранилище файлов (Storage)**

* **5.1. Загрузка файлов (Upload)**
* try
* {
* byte[] fileData = File.ReadAllBytes("path/to/your/file.jpg"); // Прочитайте данные из файла
* string fileName = "your-image.jpg";
* var response = await supabase.Storage
* .From("your-bucket-name") // Укажите имя бакета
* .Upload(fileData, fileName, new Supabase.Storage.FileOptions { ContentType = "image/jpeg" });
* Console.WriteLine($"File uploaded successfully: {response}");
* }
* catch (Exception ex)
* {
* Console.WriteLine($"Error uploading file: {ex.Message}");
* }

* **5.2. Получение URL файлов (Get URL)**
* try
* {
* string fileUrl = await supabase.Storage
* .From("your-bucket-name") // Укажите имя бакета
* .GetPublicUrl("your-image.jpg"); // Укажите имя файла
* Console.WriteLine($"File URL: {fileUrl}");
* }
* catch (Exception ex)
* {
* Console.WriteLine($"Error getting file URL: {ex.Message}");
* }

Use code [with caution](https://support.google.com/legal/answer/13505487).C#

* **5.3. Удаление файлов (Delete)**
* try
* {
* var response = await supabase.Storage
* .From("your-bucket-name") // Укажите имя бакета
* .Remove(new List<string> { "your-image.jpg" }); // Укажите имя файла
* Console.WriteLine($"File removed successfully.");
* }
* catch (Exception ex)
* {
* Console.WriteLine($"Error removing file: {ex.Message}");
* }

* **5.4. Управление политиками доступа**  
  //TODO (В следующих версиях)

**6. Функции (Functions)**

* **6.1. Описание Edge Functions**  
  //TODO (В следующих версиях)
* **6.2. Вызов функций из приложения**  
  //TODO (В следующих версиях)
* **6.3. Передача данных в функции**  
  //TODO (В следующих версиях)

**7. Устранение неполадок**

* **7.1. Общие проблемы и решения**
  + **Проблема:** Ошибка аутентификации.
    - **Решение:** Проверьте правильность email и пароля. Убедитесь, что учетная запись подтверждена (если требуется).
  + **Проблема:** Не удается получить данные из базы данных.
    - **Решение:** Проверьте правильность URL и ключа API. Убедитесь, что у вас есть права доступа к таблице.
  + **Проблема:** Не удается загрузить файл в хранилище.
    - **Решение:** Проверьте, что у вас есть права на запись в бакет. Убедитесь, что тип контента (ContentType) указан верно.
  + **Проблема:** Неправильная работа в Realtime
    - **Решение:** Убедитесь, что вы правильно обработали функцию.
* **7.2. Получение помощи и обратная связь**
  + Обратитесь к документации Supabase: [https://supabase.com/docs](https://www.google.com/url?sa=E&q=https%3A%2F%2Fsupabase.com%2Fdocs)
  + Задайте вопрос на Stack Overflow с тегом "supabase".

**8. Приложение**

* **8.1. Глоссарий**
  + **API:** Application Programming Interface - интерфейс программирования приложений.
  + **REST:** Representational State Transfer - архитектурный стиль взаимодействия между компонентами распределенного приложения в сети.
  + **JSON:** JavaScript Object Notation - текстовый формат обмена данными.
  + **URL:** Uniform Resource Locator - унифицированный указатель ресурсов.
  + **Edge Functions:** Edge Functions - функции для расширения функциональности приложения.