Нагрузочное тестирование (API).

Нагрузочное тестирование проводилось при помощи разработанной программы «NagryzkaTest». Данная программа позволяет одновременно авторизоваться при помощи API определенное пользователем количество раз.

В ходе тестирования было выяснено, что бесплатный план supabase позволяет одновременно отправлять запрос на авторизацию 30 пользователям, при этом 25 из 30 пользователей успешно проходят авторизацию, а остальным выходит сообщение о лимите одновременных запросов (Рисунок 1).

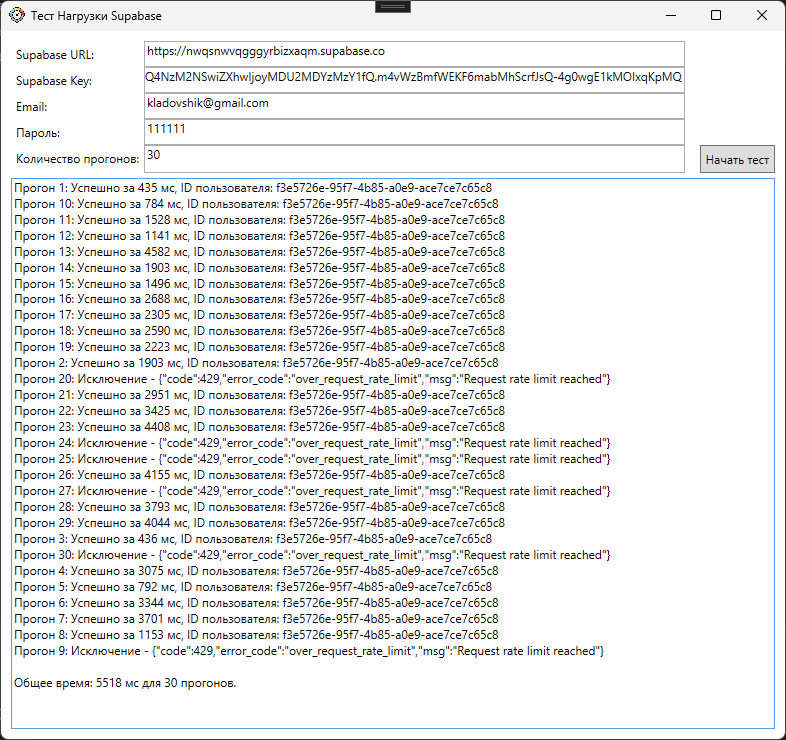


Рисунок 1 – Тестирование 30 авторизаций одновременно

Код программы:

<Window x:Class="NagryzkaTest.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:NagryzkaTest"

mc:Ignorable="d"

Title="Тест Нагрузки Supabase" Height="500" Width="800" WindowStartupLocation="CenterScreen" Icon="/logo.png">

<Grid Margin="10">

<Grid.RowDefinitions>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="Auto"/>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="Auto"/>

<ColumnDefinition Width="Auto"/>

</Grid.ColumnDefinitions>

<Label Grid.Row="0" Grid.Column="0" Content="Supabase URL:"/>

<TextBox Grid.Row="0" Grid.Column="1" Name="UrlTextBox" Text="YOUR\_SUPABASE\_URL" Margin="0,0,10,0"/>

<Label Grid.Row="1" Grid.Column="0" Content="Supabase Key:"/>

<TextBox Grid.Row="1" Grid.Column="1" Name="KeyTextBox" Text="YOUR\_SUPABASE\_ANON\_KEY" Margin="0,0,10,0"/>

<Label Grid.Row="2" Grid.Column="0" Content="Email:"/>

<TextBox Grid.Row="2" Grid.Column="1" Name="EmailTextBox" Text="test@example.com" Margin="0,0,10,0"/>

<Label Grid.Row="3" Grid.Column="0" Content="Пароль:"/>

<TextBox Grid.Row="3" Grid.Column="1" Name="PasswordTextBox" Text="password" Margin="0,0,10,0"/>

<Label Grid.Row="4" Grid.Column="0" Content="Количество прогонов:"/>

<TextBox Grid.Row="4" Grid.Column="1" Name="RunsTextBox" Text="10" Margin="0,0,10,0"/>

<Button Grid.Row="4" Grid.Column="2" Content="Начать тест" Name="StartButton" Click="StartButton\_Click" Padding="5" Margin="5,0,0,0"/>

<TextBox Grid.Row="5" Grid.Column="0" Grid.ColumnSpan="4" Name="ResultsTextBox" IsReadOnly="True" VerticalScrollBarVisibility="Auto" FontSize="12" Margin="0,5,0,0"/>

</Grid>

</Window>

using System;

using System.Collections.Generic;

using System.Threading.Tasks;

using System.Windows;

using Supabase;

using System.Diagnostics;

using System.Threading;

using System.Collections.Concurrent;

using System.Linq;

namespace NagryzkaTest

{

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

private async void StartButton\_Click(object sender, RoutedEventArgs e)

{

string supabaseUrl = UrlTextBox.Text;

string supabaseKey = KeyTextBox.Text;

string email = EmailTextBox.Text;

string password = PasswordTextBox.Text;

int numberOfRuns;

if (!int.TryParse(RunsTextBox.Text, out numberOfRuns) || numberOfRuns <= 0)

{

MessageBox.Show("Пожалуйста, введите корректное положительное число для количества прогонов.");

return;

}

ResultsTextBox.Clear();

await RunLoadTest(supabaseUrl, supabaseKey, email, password, numberOfRuns);

}

private async Task RunLoadTest(string supabaseUrl, string supabaseKey, string email, string password, int numberOfRuns)

{

var results = new ConcurrentBag<string>();

var stopwatch = Stopwatch.StartNew();

var tasks = new List<Task>();

for (int i = 1; i <= numberOfRuns; i++)

{

int runNumber = i;

tasks.Add(Task.Run(async () =>

{

try

{

var client = new Supabase.Client(supabaseUrl, supabaseKey, new SupabaseOptions

{

AutoRefreshToken = false,

});

await client.InitializeAsync();

var start = Stopwatch.StartNew();

var authResponse = await client.Auth.SignIn(email, password);

start.Stop();

if (authResponse?.User != null)

{

results.Add($"Прогон {runNumber}: Успешно за {start.ElapsedMilliseconds} мс, ID пользователя: {authResponse.User.Id}");

}

else

{

results.Add($"Прогон {runNumber}: Неудачно за {start.ElapsedMilliseconds} мс, Ошибка: Неизвестно (проверьте исключения)");

}

}

catch (Exception ex)

{

results.Add($"Прогон {runNumber}: Исключение - {ex.Message}");

}

finally

{

}

}));

}

await Task.WhenAll(tasks);

stopwatch.Stop();

Application.Current.Dispatcher.Invoke(() =>

{

foreach (var result in results.OrderBy(r => r))

{

ResultsTextBox.AppendText(result + Environment.NewLine);

}

ResultsTextBox.AppendText($"\nОбщее время: {stopwatch.ElapsedMilliseconds} мс для {numberOfRuns} прогонов.");

});

}

}

}