# „Users” Applications

Created by: Kornidesz Máté

<https://github.com/Kornimate>

<https://kornimate.github.io/introduction/>

List of content

[„Users” Applications 1](#_Toc181535673)

[1. Task 3](#_Toc181535674)

[1.1 System Requirements 3](#_Toc181535675)

[2. User’s documentation 3](#_Toc181535676)

[2.1. Aspire Web App Host 3](#_Toc181535677)

[2.2. API Swagger UI 4](#_Toc181535678)

[2.3. WPF Application 4](#_Toc181535679)

[2.3.1. Home Page 4](#_Toc181535680)

[2.3.2. 5](#_Toc181535681)

# 1. Task

TodoApi.7z contains a small .NET web API

Your tasks are:

1. Rename TodoItem to User and expand the User model to include fields like Address (street, house number, zip, city, country).

2. Write a console application that will call this API, allowing you to retrieve a User by a given ID and perform all other CRUD operations.

3. Create two requests that, for example, call GetUser(Id) 1000 times: one that executes sequentially and another that runs in parallel.

## 1.1 System Requirements

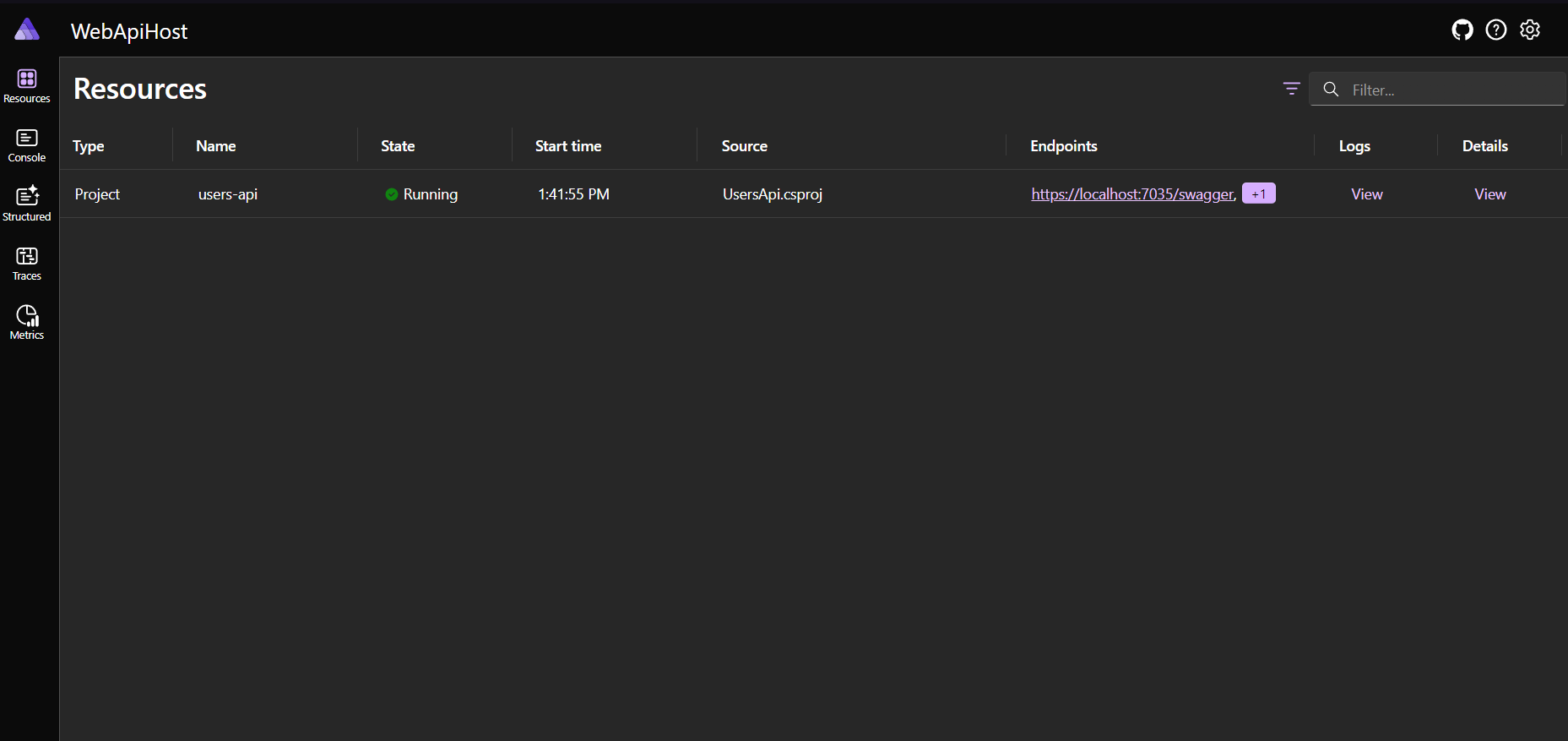
* Windows machine (Win 10 or later)
* 1GB of RAM
* .NET 8 runtime

# 2. User’s documentation

The application consits of a web API a WPF desktop application and an Aspire project for local monitoring of the web API. To start the application it is recommended to have multiple start up projects configured in Visual Studio which are the following: The Aspire project (which starts the web API and the WPF application.

## 2.1. Aspire Web App Host

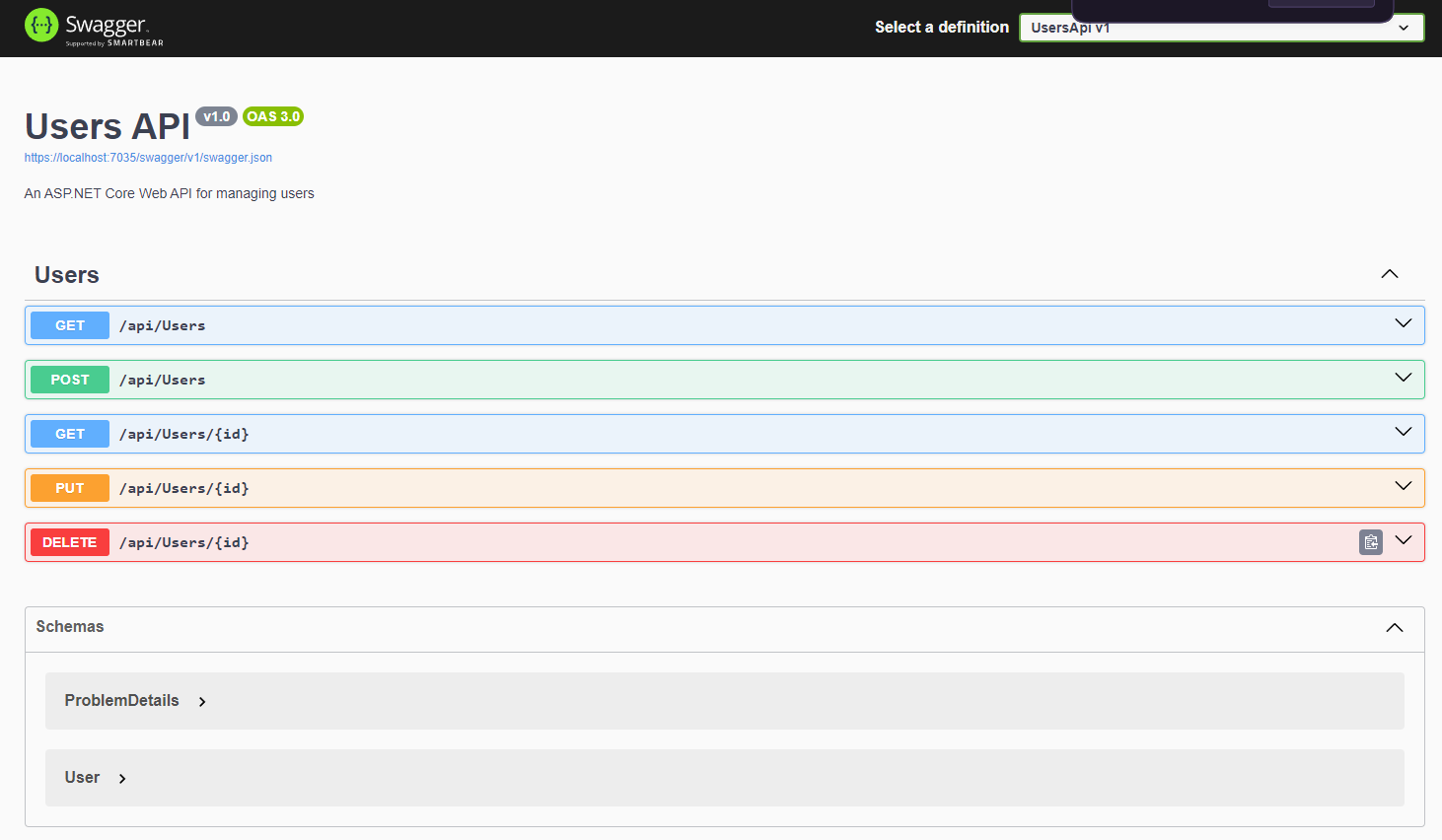
This application provides a UI for monitoring the registered services. The web API is registered so it will be monitored through Aspire’s dashboard. It also provides navigation to the wep API’s UI.



1. picture: Aspire Dashboard

## 2.2. API Swagger UI

The API provides a swagger UI for interactive interaction with the API itself and it is the documentation for the API usage too as well as for the DTO present in it.



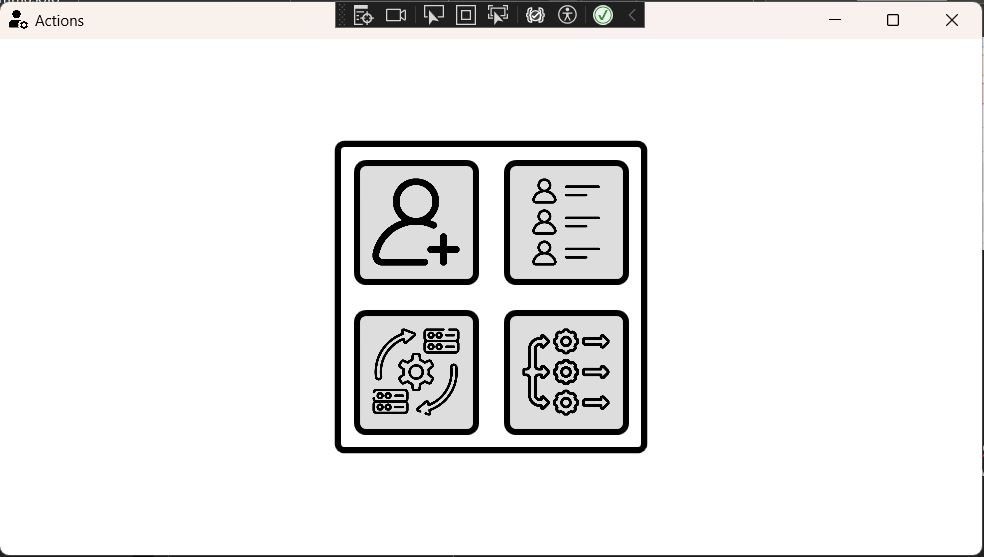
2. picture: Web API swagger UI

## 2.3. WPF Application

The WPF application is simulating the CRUD operations to the web API through a desktop UI.

### 2.3.1. Home Page

The home page contains 4 buttons. One for creating new user, one for listing existing users, one for simulating API calls synchrouously, one for simiulating API calls in a parallel way.



4.

3.

2.

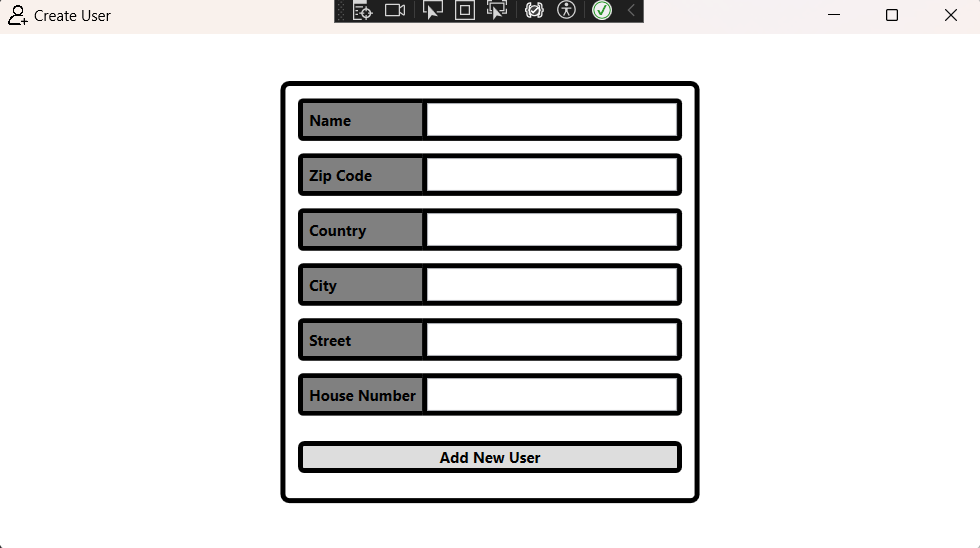
1.

3. picture: WPF app home page

1. Button to open create user window
2. Button to open list users window
3. Button to start synchronuous API calls simulation
4. Button to start parallel API calls simulation

### 2.3.2. Create User Page

This page enables the user to create new user and save it to the database through the API. This page has some restrictions as empty data is not allowed, house number and zip number fields only allow numbers. By clicking the add user page the app will post data to the endpoint and will notify the user about the success of the action.



4. picture: Create user page