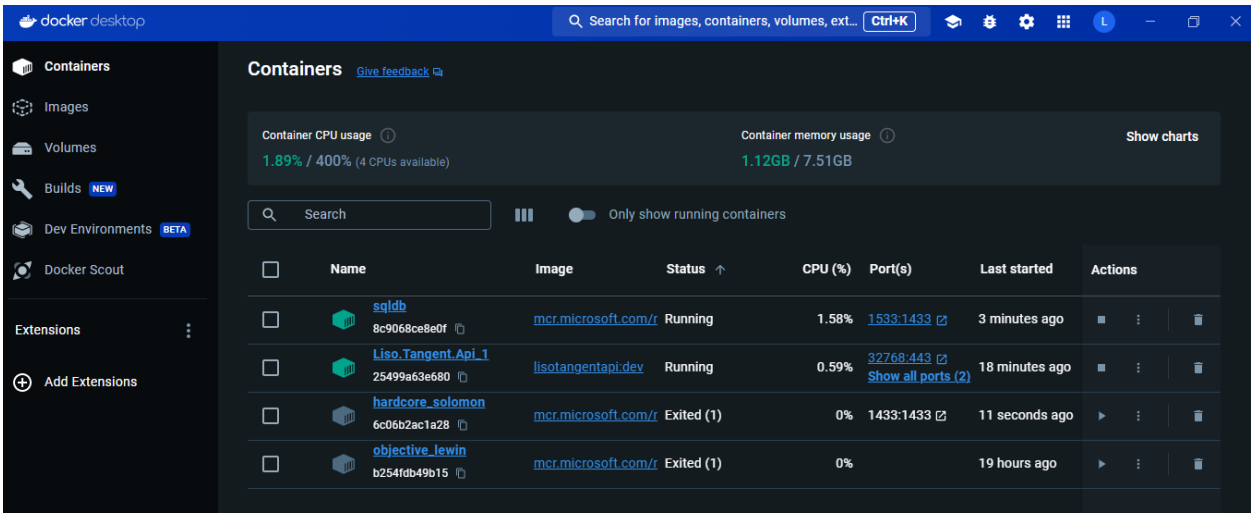
Approach

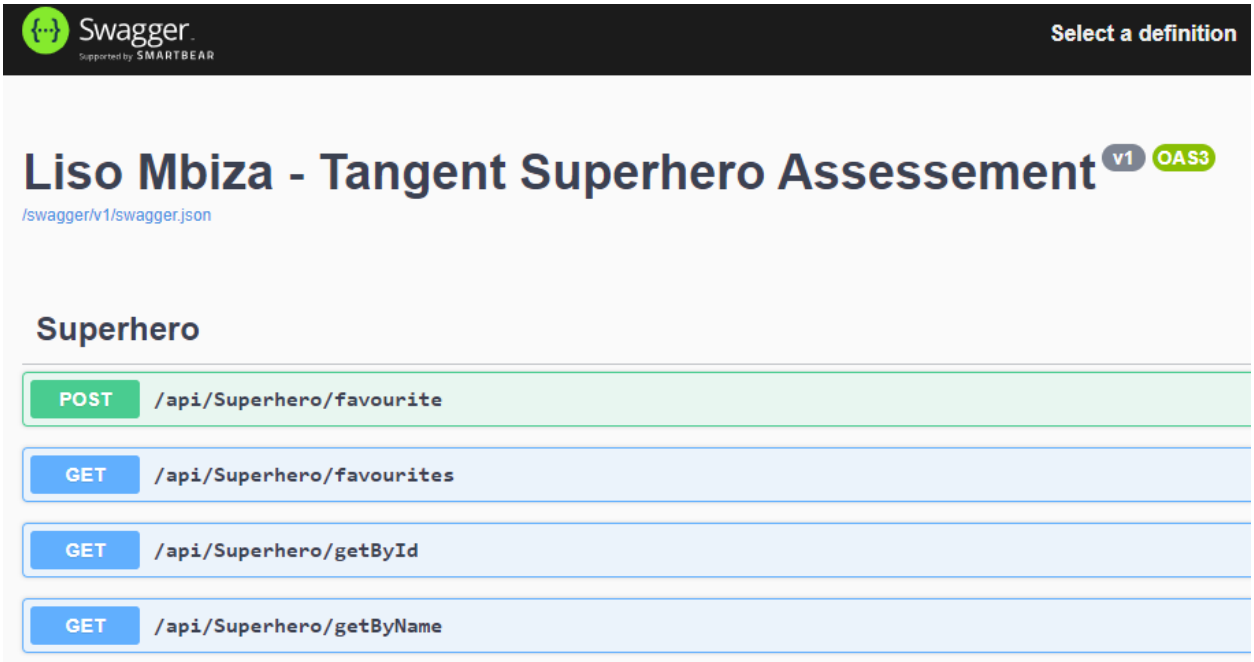
I decided to take the TDD approach. I create a REST api and started from the ground up. By that I mean I created the repo project to save and read the favourites (Open-Closed & Liskov Substitution). I created the Service/Business logic project to consume the repository project. I then created the unit test for the services (making use of Interfaces for each process/component and not have 1 huge interface file “Interface Segragation”). I then created a library to call the superhero api. I decide to create a library for this to make it easy to create unit tests for the outgoing calls. I then hooked in the Rest Api to the services. I have this decision to separate the processes to make it easy to just plugin/remove any component. (Single Responsibility). These interfaces would be injected (Dependency Inversion) by making use of the built in functionality that comes with .net5.

Tech

I have written the assessment making use of:-

- .net5
- SQL Server Express for testing locally
- Added docker file for deploying to the docker container
- Unit Test (MSTest framework) for Services and Api
- Migration scripts are created in the repository project




Swagger
Supported by SMARTBEAR
Select a definition
Liso Mbiza - Tangent Superhero Assessement
v1
OAS3
/swagger/v1/swagger.json
Superhero
POST
/api/Superhero/favourite
GET
/api/Superhero/favourites
GET
/api/Superhero/getById
GET
/api/Superhero/getByName