**Unit Testing**

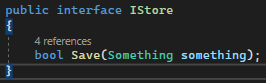
<https://github.com/tenPro4/NetCoreUnitTestBasic>

**Mock**

Override the provider service for testing purpose. It also provide verify tools for services testing.

**Exp1 Simple override**

1. have a interface class



2.



3. setup the interface to override the service. In the following case, it just simply return true.



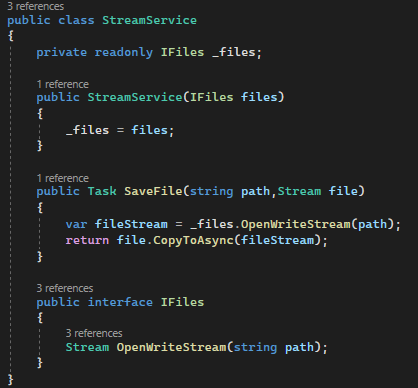
4. verify result



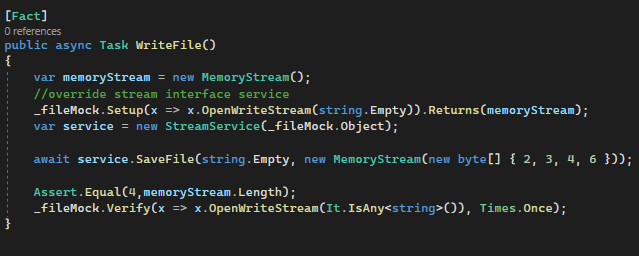


**Exp2 Mock I/O Service**

1.



2.



**Class fixture and Async Life Time**

1. class fixture is kind of class integrate to testing class so don’t need repeat setup code in every single testing class.

2. IAsyncLife time is the service where it provide initialize and dispose function. Tester can ask the what should do once app start and what should do once app is end.

Text

Description automatically generated





**Mock AuthenticationHandler**

1. override handler

Text

Description automatically generated

2. override provider

Text

Description automatically generated

3. register provider

Text

Description automatically generated

4.

Text

Description automatically generated

Graphical user interface, text, application

Description automatically generated

**Theory**

Text

Description automatically generated

**Test Api**

Graphical user interface, text, application

Description automatically generated

1. connect the select startup

2. enable dispose the test server once app is done.

3. to perform testserver, need install package Microsoft.AspNetCore.TestHost



Post request

Text

Description automatically generated

Make Db as singleton instance

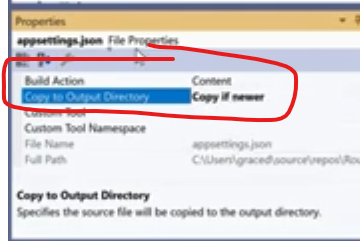
Graphical user interface, text, application, Word

Description automatically generated

Override appsettings file for test unit

1. create appsettings.json file to test project

2. Open appsettings.json properties, and change



3. register appsettings during the test unit setup

Text

Description automatically generated

4. debug app and put breakpoint to watch the data

Graphical user interface, application, Word

Description automatically generated

**Migrate db through dotnet ef or visual studio tool**

1.Dotnet ef

i. dotnet tool install --global dotnet-ef

ii. Execute migrate command,exp:

Graphical user interface, text, application, email

Description automatically generated

2. Visual Studio tool

Reference link: https://www.entityframeworktutorial.net/efcore/pmc-commands-for-ef-core-migration.aspx

i. Install two package:

Microsoft.EntityFrameworkCore.Tools

Microsoft.EntityFrameworkCore.Design

ii. Execute following command:

add-migration xxx -c XXXContext -s [project path] -o .\Migrations

update-database -Context XXXContext