Developer Guide

This documentation is to guide other developers on the framework and standards I have implemented.

Visual Studio Projects

1. **SingleSignOn** : This project holds the authentication web service. It is a Web API projects.
2. **SingleSignOn.Core**: This is a library project which holds all the backend processing & algorithm for SingleSignOn Authentication Service.
3. **SampleHost**: This is a simple MVC project which allows user to login and create user. This site project authenticates through SingleSignOn Web Services
4. **SingleSignOn.Test**: This is a Ms-Test projects which I wrote some unit test on the work I have done.

Technology used & purpose

1. **Castle Windsor :** This technology is used to perform dependency injection. For more regarding how to setup dependencies injection using Castle Windsor, please refer to "Installer" folder under "SingleSignOn.Core" project.
2. **Mediator:** This project uses "Mediator" framework where by the backend Web Api controller receives message from front end, and it will pass the message to the relevant class/services (I named it Handler) listening/subscribing to the message. One example would be "SignInHandler.cs", which listen to login request and process the request.
3. **FluentValidation**: This library was used to store and create validation for all operation. For more description on how to use it, please refer to "SignInValidator.cs" file under "Validators" folder in "SingleSignOn.Core" project.
4. **NLog**: This library was used to store logs and errors happened within the system. I used it mostly to log database error within this project.
5. **Shouldly & FakeItEasy**: I used these unit testing framework in my test project. I used Shouldly for easier assertion, and FakeItEasy for Mocks and Stubs. The example can be seen on "SingleSignOn.Test" project.