Overall the sample codebase is well written. It has a very clear separation of concerns, splitting up the logic intelligently between commands, command handlers and view model builders. There could have been further abstraction which I have done, such as create a generic base command handler class which contains the base constructor as well as an abstract Handle method. Another abstraction that could have been made was for the view model builders which also contained a common constructor. That way it’s written only once and if it ever needs to change, there’s a centralized place to do it.

There were a couple of other small things that I changed which follows best practices. One example was the handling of roles. Before, the role names were string literals everywhere. This is a concern for the same reason as above, if any of the role names needed to be changed, they would have to be changed everywhere. I created a RoleConstants class to hold the role names and used those variables to authenticate against the roles. Another minor change was in the OffersOnPropertyViewModel file where there were multiple classes in one file. It’s best practice to have one class per file to avoid confusion.

The sample codebase made it easy to both extend the offer functionality as well as easy to add the appointment functionality.