**Hello World API – WCF or WebAPI**

Since this API that retrieves the “Hello World” message is going to be potentially used by Mobile/Web applications, it has to be exposed through HTTP over REST protocol. There is a design decision to be made on whether it should be implemented in WCF or WebAPI. WCF supports REST protocol, but it requires a lot more configuration and hence additional maintenance than WebAPI. So I’m going with WebAPI.

**Message Factory**

Currently the message is hard-coded in HelloWorldMessageFactory. But the code is written with dependency on abstraction (IHelloWorldMessageFactory) so that the current factory can be replaced with another implementation.

**Unity Inversion of Control**

The dependencies should be interchangeable to allow for future enhancement of these two scenarios mainly:

1. Swap out current implementation of HelloWorldMessageFactory to get the message from the Database instead.
2. Currently the message is written to console by ConsoleMessageHandler. This can be changed to another implementation like write the message to DB, or raise an event by creating a concrete implementation of IHelloWorldMessageHandler and configuring it in the Unity.config, configuration file.

**Unit Testing**

I’m using Microsoft Fakes framework for mocking objects. I could write more tests cases to get complete coverage though.