WCF

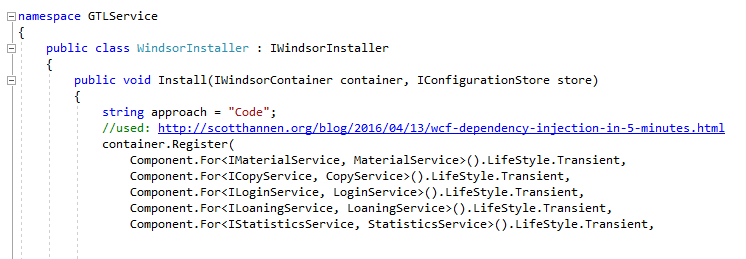
Defining an endpoint is necessary as it is the only way of communicating with it. To define an endpoint we need its address, its binding and its contract (the ABCs of WCF)

* Address defines where a service is hosted, an address can be an IP address, a server name or even a URL
* Binding defines how messages are handled in both the client and server side. There are multiple binging types each with their own strengths and weaknesses. We chose to use the HTTP binding as its very well suited for connecting web services.
* Contract is the agreement between the client and the server that specifies the structure and the contents of the messages sent and received. The data which is the agreement of the structure of the message and the message contract is about the content.

To communicate with the service establishing communication is necessary, meaning a client proxy needs to be created, this is done by adding a service reference to the client which will be using it.

Dependency injection in WCF

Dependency injection is great as it means code becomes less coupled and more flexible as the dependencies of classes are injected during runtime and it simplifies the running of tests by mocking dependencies in places of real classes. For our project we used Castle Windsor, a dependency container which is extensively used by the developer community and comes preloaded with WCF integration facilities.



Figure

Using Windsor we can instantiate an object required in the parameter of the constructor of class. This allowed us to use the dependency injection, even though the default constructor of WCF service should be parameter less. In the already shown snippet of the code (Figure 1) we can observe how it was done (If IMaterialService was required, the new MaterialService would be returned). This approach, although not being the only possible solution to this problem, allowed us to keep code simple and organized, not requiring multiple constructors for each class, while also providing improved testability.