

Credit Task 5.3 Stateful behavior retention

WCF services allow external systems or programs to communicate with each other in many ways. In enterprise system, WCF can be utilized to implement stateful and stateless business objects and they can be hosted inside a Windows Service, a console program or in IIS. In a design where Business Objects expose their services (or method call) to other system using WCF hosted on IIS, in order to retain stateful information stored in business objects, a number of configuration need to be done on the WCF services. First of all, the service needs to be configured to communicate using a session-enabled binding. This is required so that the communication channel between the client and the WCF service is maintained during the whole transaction. For example of some session enabled bindings on IIS are WsHttpBinding, WsDualHttpBinding, WsFederationHttpBinding. If hosted under Windows Service or self-host mode in a console program, the above binding plus NetTcpBinding can be used for session-enabled bindings. Such binding can be set under the configuration file as follow:

```
<system.serviceModel>
  <protocolMapping>
    <add scheme="http" binding="wsHttpBinding" />
  </protocolMapping>
  <!-- binding configuration - configures WSHttp binding for reliable sessions -->
  <bindings>
    <wsHttpBinding>
      <binding>
        <reliableSession enabled="true" />
      </binding>
    </wsHttpBinding>
  </bindings>
  <behaviors>
    <serviceBehaviors>
      <behavior>
        <serviceMetadata httpGetEnabled="True"/>
        <serviceDebug includeExceptionDetailInFaults="False" />
      </behavior>
    </serviceBehaviors>
  </behaviors>
  <serviceHostingEnvironment multipleSiteBindingsEnabled="true" />
</system.serviceModel>
```

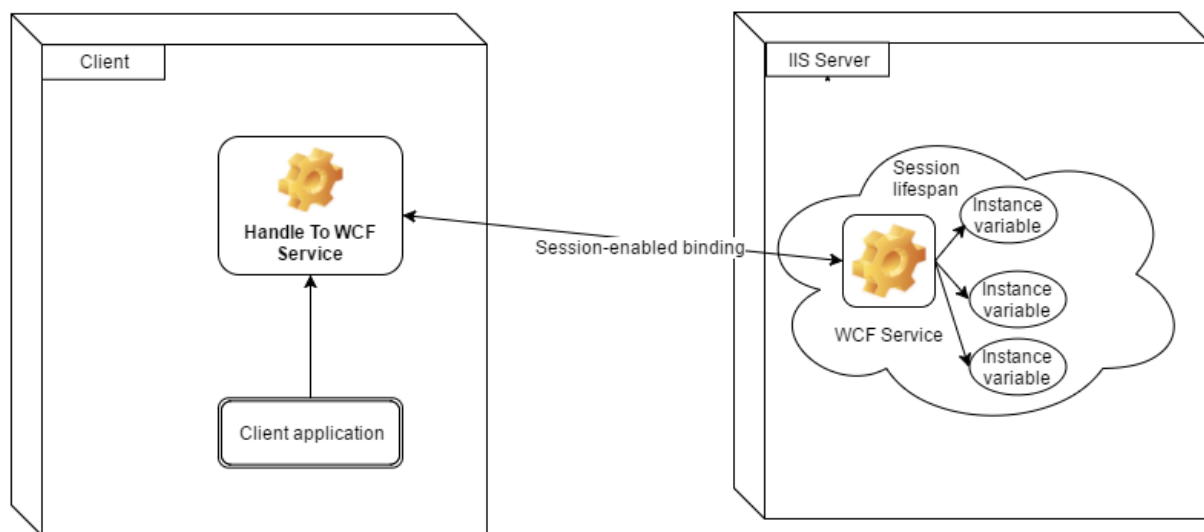
Secondly, the lifetime of WCF needs to be configured so that it will live during a whole session. This can be done by decorating the WCF service with ServiceBehaviour attribute:

```
[ServiceBehavior(InstanceContextMode = InstanceContextMode.PerSession)]
public class ShopCartB0 : IShopCartB0
{
    ...
}
```

The effect of this is that the first time a client calls a method from proxy object, WCF runtime will create an instance of Business Object (or effectively WCF service), then a handle to the service is created on the client site to keep a reference to the newly created

instance of the WCF service on the server side. It is important to note that if this handle is recreated in the client, a new WCF service will be created on the Server side and this new handle will point to it. Therefore, to keep the behavior of a stateless service, client must retain the handle and use this to call WCF service throughout the whole transaction. By doing so, instance variables that attached in that WCF service will be maintained as long as the handle is maintained on the client side.

In unlike ASP.NET WebForm where the stateful behavior of a page is achieved using VIEWSTATE, ASP.NET MVC is a stateless technology so any stateful behavior need to be developed manually. Therefore, in order to keep the WCF service as a stateful business object, one strategy is to use Server Session to maintain the handle to the WCF service for all consecutive calls from a client.



References:

- (1) 2016, *How to: Host a WCF Service in IIS*, Microsoft MSDN, viewed 31th Mar 2016, <[https://msdn.microsoft.com/en-us/library/ms733766\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/ms733766(v=vs.110).aspx)>
- (2) 2016, *Sessions, Instancing and Concurrency in WCF Services*, Microsoft MSDN, viewed 1st Jun 2016, <<https://msdn.microsoft.com/en-us/library/ff183865.aspx>>