**Operation and EndPoint Behaviors in WCF**

****

**Rameshkartik.RS**

**Table of Contents**

[Operation Behavior 3](#_Toc389488394)

[ReleaseInstanceMode – SourceCode Explanation 4](#_Toc389488395)

[Endpoint Behaviors 7](#_Toc389488396)

[Attachment 9](#_Toc389488415)

[Summary 9](#_Toc389488419)

**WHAT’s IN THIS ARTicle?**

* Operation Behavior
* ReleaseInstanceMode
* Endpoint Behavior
* Attachment
* Summary

# Operation Behavior

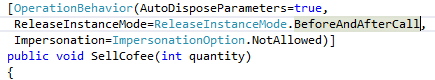
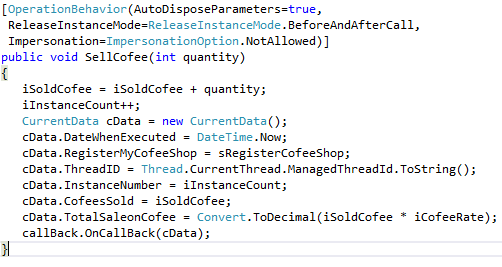
The Operation Behavior is used to apply behavior at the method. It allows you to control things such as

* Object Recycling
* Transactional
* Caller Identity

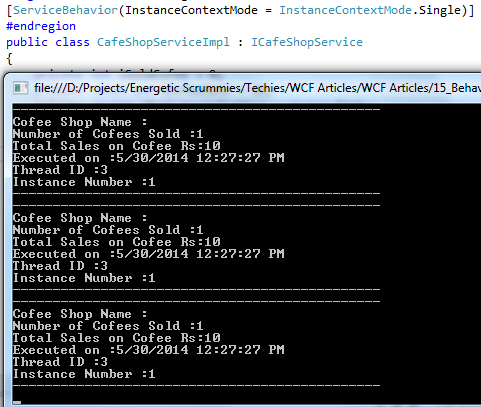
The above can be achieved by setting its properties

|  |  |  |
| --- | --- | --- |
| **Operation Behavior Properties** | | |
| **S.No** | **Property** | **What for?** |
| 1 | Auto Dispose Parameters | This property automatically disposes the input parameters, output and reference parameters. Default is True |
| 2 | TransactionAutoComplete | When the property is enabled and if the method is executed successfully without any errors ,it is marked as TransactionComplete, otherwise the transaction is aborted |
| 3 | TransactionScopeRequired | Used to set, if the transaction is required for this method |
| 4 | Impersonation | Sometimes we need to execute some method operations with the windows identity. For that we need to enable this property |
| 5 | ReleaseInstanceMode | This property overrides the InstanceContextMode. If the property is set to BeforeAndAfterCall it will create a new instance before a method call and it releases the memory after the call |

# ReleaseInstanceMode – SourceCode Explanation

As I told you earlier, if you want your service instance is Single, (*I.e. single instance will be used for all the clients*) and particular operation would be recycled(new instance on every time) on every method call, ReleaseInstanceMode is the perfect solution, because it creates the instance Before the Method call and it release the instance after the call. Let’s see this concept with the source code. To use the ReleaseInstanceMode for a particular operation, use the attribute ReleaseInstanceMode above the service interface method as it is mentioned in the above snapshot.

To understand the concept much better am just updating the iSoldCofee variable on every time the method gets called from client, to just count and maintain the number of coffees sold by a client. If you see the below attached output you can find that number of coffees sold will always be a value 1. Its means that when every time the method gets called; it creates a new service instance and it executes the method then it releases the instance since the Release InstanceMode we have chosen here is BeforeAndAfterCall.As like this you can also check it for BeforeCall , AfterCall separately . BeforeCall will create a new service instance before the service method gets executed. AfterCall will deactivate the instance after a call is made to the operation. Kindly refer the attached source code for further reference.



# Endpoint Behaviors

# Endpoint behaviors can be applied at both the services and the client level, Credential

# usage can be managed by applying specific behavior in the Endpoint. By using the config

# file you can specify and configure the endpoint behaviors. For Example, in the client side

# If you want to indicate the credentials that a client must use when calling a particular

# Endpoint, you can use the ClientCredential in the Endpoint behavior. Following are the

Steps to be followed in the client side to specify the endpoint behavior

1. Add an Endpoint behavior <endpointBehaviors>
2. Add a ClientCredentials to the Behavior Element
3. Add a ClientCertificate Element
4. Set The following Attributes with the values StoreLocation,StoreName,X509FindType and findValue

<behaviors>

<endpointBehaviors>

<behavior name="endpointCredentialBehavior">

<clientCredentials>

<clientCertificate findValue="CN=CafeClient"

storeLocation="CurrentUser"

storeName="My"

x509FindType="FindBySubjectDistinguishedName" />

</clientCredentials>

</behavior>

</endpointBehaviors>

</behaviors>

# The above is the one way (via Config) to mention the client credentials, But if you look

# the source code here I attached you can find that I have mentioned the client

# credentials using the source code. It’s says that client must use these credentials to

# access the service. What would be the next step? We need to create the service

# credentials in the service side,

<serviceBehaviors>

<behavior name="serviceBehavior">

<serviceCredentials>

<serviceCertificate findValue="CN=CafeServer" storeLocation="CurrentUser" storeName="My" x509FindType="FindBySubjectDistinguishedName"/>

</serviceCredentials>

</behavior>

</serviceBehaviors>

# As like above we need to create the service credentials in the service behavior. This

# actually sets that the service will be hosted if the certificates are available in the server

# system. So before hosting the server, kindly make sure the respective certificates are

# created in both server and the client. How you are going to create a certificate? Use

# Makecert.exe in the Visual studio command prompt to generate the certificate. Kindly

# refer my other articles where I have dictated specifically about the certificates, in this

# title you just need to know about only the Endpoint Contract. Refer the attached code

# for further details

# Attachment

# Please refer the Operation Behavior Code folder to see about the operation behavior

# attributes, To know about the Endpoint behavior please refer the Endpoint Behavior

# Code Folder

# Summary

# Operation behavior can be performed at the service level, where as the Endpoint

# behavior can be performed at both client and the service levels