Three ways to do WCF instance management

Shivprasad koirala
9 Jun 2010 CPOL

Three ways to do WCF instance management (Per call, Per session, and Single).

Download source code - 39.1 KB

Table of contents

Introduction

Very often we would like to control the way WCF service objects are instantiated on a WCF server. You would want to control how long the WCF instances should be residing on the server.

The WCF framework has provided three ways by which we can control WCF instance creation. In this article, we will first try to understand those three ways of WCF service instance control with simple code samples of how to achieve them. Finally, we will compare when to use them and under what situations.

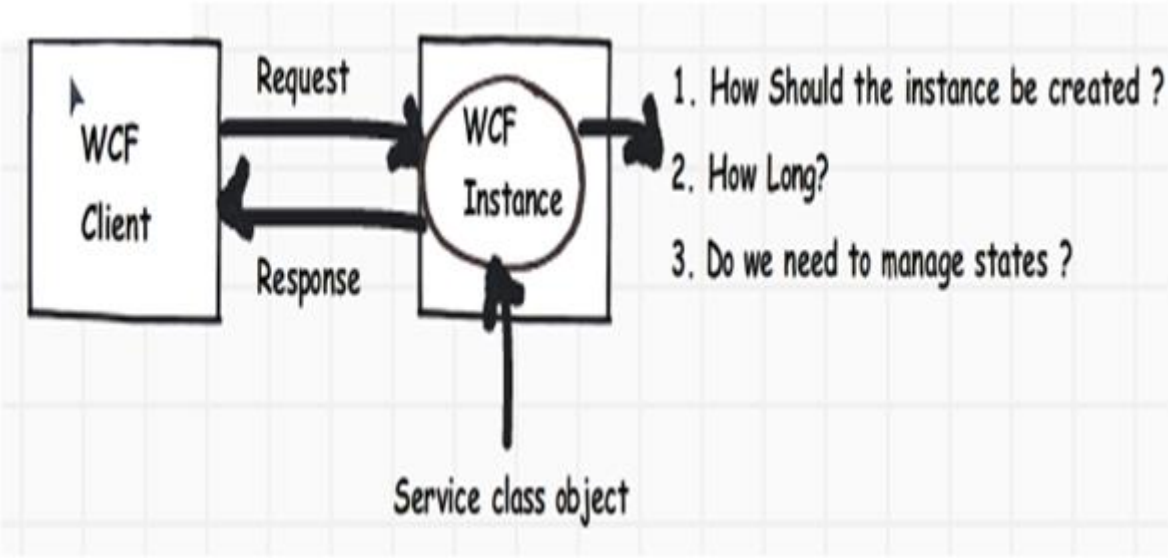
There is a small ebook for all my .NET friends which covers topics like WCF, WPF, WWF, AJAX, Core .NET, SQL, etc., which you can download from **here** or you can catch me on my daily free training from **here**.

WCF service object instancing basics

In normal WCF request and response communication, the following sequence of actions takes place:

- WCF client makes a request to a WCF service object.
- WCF service object is instantiated.
- WCF service instance serves the request and sends the response to the WCF client.

Following is a pictorial representation of how WCF requests and responses work.



Following are different ways by which you can create WCF instances:

- Create a new WCF service instance on every WCF client method call.
- Only one WCF service instance should be created for every WCF client session.
- Only one global WCF service instance should be created for all WCF clients.

To meet the above scenarios, WCF has provided three ways by which you can control WCF service instances:

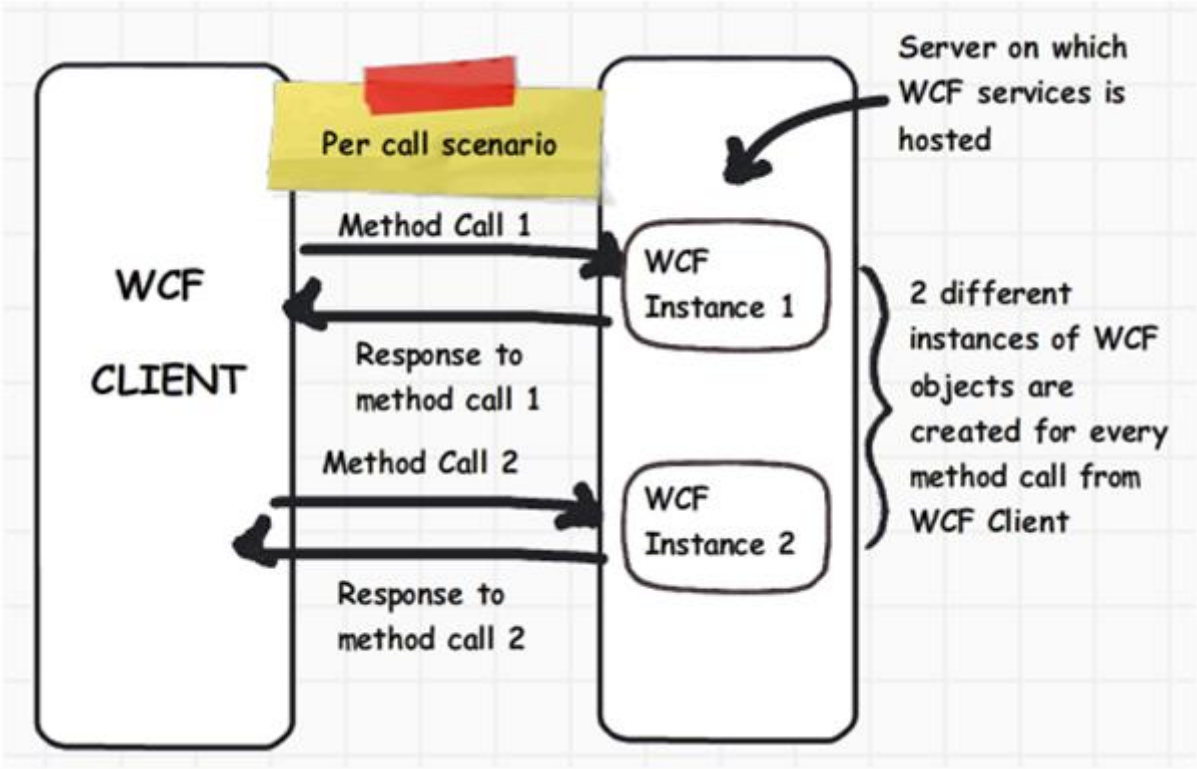
- Per call
- Per session
- Single instance

Per call instance mode

When we configure a WCF service as per call, new service instances are created for every method call you make via a WCF proxy client. The image below shows this in a pictorial format:

- The WCF client makes the first method call (method call 1).
- A new WCF service instance is created on the server for this method call.
- The WCF service serves the request and sends the response and the WCF instance is destroyed and given to the garbage collector for clean up.
- Now let's say the WCF client makes a second method call, a new instance is created, the request is served, and the WCF instance is destroyed.

In other words, for every WCF client method call, a WCF service instance is created, and destroyed once the request is served.



How to implement WCF per call instancing

In order to specify the instancing mode, we need to provide the `InstanceContextMode` value in the `ServiceBehavior` attribute as shown below. This attribute needs to specified on the `Service` class. In the below code snippet, we have specified `intCounter` as a class level variable and the class counter is incremented by one when the `Increment` method is called.

```
[ServiceBehavior(InstanceContextMode = InstanceContextMode.Percall)]
public class Service : IService
{
    private int intCounter;

    public int Increment()
    {
        intCounter++
        return intCounter;
    }
}
```

At the client, we consume the WCF client and we call the `Increment` method twice.

```
ServiceReference1.ServiceClient obj = new ServiceReference1.ServiceClient();
MessageBox.Show(obj.Increment().ToString());
MessageBox.Show(obj.Increment().ToString());
```

Even though we have called the `Increment` method twice, we get the value '1'. In other words, the WCF service instance is created for every method call made to the WCF service instance so the value will always be one.

Per session instance mode

Very often we need to maintain state between method calls or for a particular session. For those kinds of scenarios, we will need to configure the service per session. In per session, only one instance of a WCF service object is created for a session interaction. The figure below explains this in pictorial format.

- The client creates the proxy of the WCF service and makes method calls.
- A WCF service instance is created which serves the method response.
- The client makes one more method call in the same session.
- The same WCF service instance serves the method call.
- When the client finishes its activity, the WCF instance is destroyed and served to the garbage collector for clean up.

How to implement per session instancing

To configure service as per session, we need to configure the `ServiceBehavior` attribute with a `PerSession` value in the `InstanceContextMode` object.

```
[ServiceBehavior(InstanceContextMode = InstanceContextMode.PerSession)]
public class Service : IService
{
    private int intCounter;
    public int Increment()
    {
        intCounter++
        return intCounter;
    }
}
```

At the client side, when we run the below client code, you should see the value '2' after the final client code is executed. We have called the method twice so the value will be seen as two.

```
ServiceReference1.ServiceClient obj = new ServiceReference1.ServiceClient();
MessageBox.Show(obj.Increment().ToString());
MessageBox.Show(obj.Increment().ToString());
```

Single instance mode

Often we would like to create one global WCF instance for all WCF clients. To create a single instance of a WCF service, we need to configure the WCF service as `Single` instance mode. Below is a simple pictorial notation of how the single instance mode will operate:

- WCF client 1 requests a method call on the WCF service.
- A WCF service instance is created and the request is served. The WCF service instance is not destroyed, the service instance is persisted to server other requests.
- Now let's say some other WCF client, e.g., client 2, requests a method call.
- The same WCF instance which was created by WCF client 1 is used to serve the request for WCF client 2. In other words, only one global WCF server service instance is created to serve all client requests.

How to implement single instance mode

In order to create a single instance of a WCF service, we need to specify `InstanceContextMode` as `Single`.

```
[ServiceBehavior(InstanceContextMode = InstanceContextMode.Single)]
public class Service : IService
{
}
```

If you call the WCF service from a different client, you will see the counter incrementing. The counter becomes a global variable.

When should you use per call, per session, and single mode?

Per call

- You want a stateless services.
- Your service holds intensive resources like connection objects and huge memory objects.
- Scalability is a prime requirement. You would like to have a scaled out architecture.
- Your WCF functions are called in a single threaded model.

Per session

- You want to maintain states between WCF calls.
- You a scaled up architecture.
- Light resource references.

Single

- You want share global data through your WCF service.
- Scalability is not a concern.

References

- MSDN link for WCF instances: http://msdn.microsoft.com/en-us/library/ms733040.aspx.
- Do not miss this post which covers end to end about WCF sessions: http://codeidol.com/csharp/wcf/Instance-Management/.
- Great blog by Rick rain on WCF instancing: http://blogs.msdn.com/b/rickrain/archive/2009/06/15/wcf-instancing-concurrency-and-throttling-part-1.aspx.

Source code

You can download the source code for this tutorial from **here**.

License

About the Author

Shivprasad koirala

Architect https://www.questpond.com
India

Do not forget to watch my Learn step by step video series.

Learn Azure Step by Step
Azure AZ-900 fundamentals certification
Angular Interview Questions and Answers
Learn Angular tutorials step by step for beginners
Learn MVC 5 step by step in 16 hours
Learn MVC Core step by step
Learn Design Pattern in 8 hours
C# tutorial for beginners(4 hrs)
Learn MSBI in 32 hours
Learn SQL Server in 16 hours series
Learn Power BI Mobile Step by Step
Learn Data structures and Algorithm Step by Step
Learn SharePoint Step by Step in 8 hours
Learn Python Step by Step
Learn Data Science Step by Step
Step by Step Mathematics for Data Science
Learn JavaScript in 2 hours series

Comments and Discussions

102 messages have been posted for this article Visit **https://www.codeproject.com/Articles/86007/3-ways-to-do-WCF-instance-management-Per-call-Per** to post and view comments on this article, or click **here** to get a print view with messages.