15,041,767 members

200

articles Q&A forums stuff lounge ?

Search for articles, questions,

Watch

Asynchronous web services call in ASP.NET

KK Kod

27 Jun 2013 CPOL

Rate me: 4.72/5 (15 votes)

This document describes how to use the web service method calls efficiently and effectively, by making use of Asynchronous calling Mechanism.

Introduction

I would like to write down a scenario before I dive into the topic.

In an application, I am consuming a web service method which does a huge process,
Takes around 3 to 4 minutes to complete,
Say an IO operation like writing into a text file or XML file or something like that,
And also after having written all these data into the file by the web service, the consuming application itself has to perform some sort of time consuming job like reading data from a file or fetching data ,
and processing the fetched data which also takes around 5 to 6 min to complete ...
So what will normally happen in our applications ....?
Like all other application service calls the thread which executes the task (the main thread) will wait till the web service call finishes and returns the output, right..?
What will happen if we can call the web service method in a separate thread?
Like a new thread executes the service and the main thread continues executing the rest of the program..
After completing the execution, the web service calls back the consuming application and says
"**I have completed my job ...**"
Seems nice, right?
Yes it's a really wonderful mechanism.
In ASP.NET we have a mechanism to call web services asynchronously.
This will help handling the requests in with a large number of hits by releasing threads without making them wait to process the requests.

How can we achieve this?

We are going to achieve this with the help of an asynchronous web service call mechanism in ASP.NET.
This mechanism was introduced in .NET Framework 2.0.

Here I am going to explain this with a simple example.

Sample project

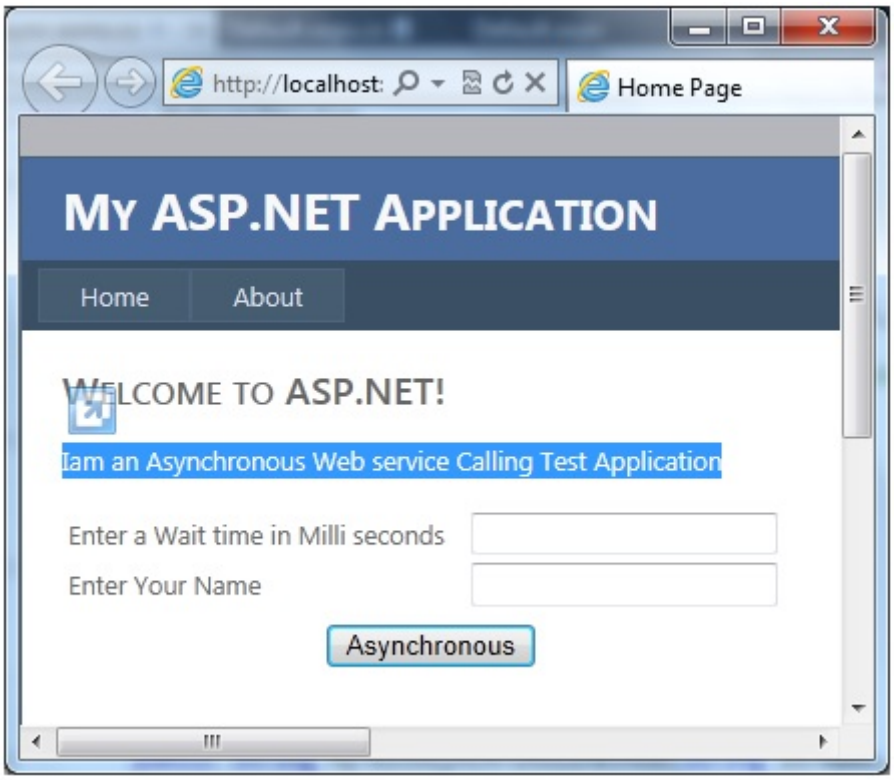
I am going to create a simple Web Service,With a service method called `MyTestAsynchronousMethod.` Which accepts two input parameters: `name` and `wait time`.
The output is a string, like `"Hei...XXXX..Iam Called Asynchronously"`
which will be returned after waiting for some time which you supply as a wait time parameter.
So the requirement is clear right...?

Let's start with the project..
Open Visual Studio...
It will create a Web service called *MyAsync.asmx with* a sample web method inside.
You can see the code snippet of the web method below:

C# Copy Code

```
[WebMethod]
public string MyTestAsynchronousMethod(string strName, int
waitTime)
{
  System.Threading.Thread.Sleep(waitTime);
  return "Hei..." + strName + "Iam Called Asynchronously";
}
```

By looking at *Snippet 1* you will understand what the web service method is doing, right?
And also we create a consuming ASP.NET web application that looks like the below screenshot.



It's quite simple with two text boxes.
One to enter the wait time and the other to enter your name.. with a button,
And also I have put a label to display the result of the web service call.

We are done with the consuming application and the service.
Now the only thing we have to do is call the web service in the button click event.

To call the application on the button click event we have to create a proxy.
As all of you know we can create a proxy using *WSDL.EXE* and also by adding a web reference to the consuming application.
Here I am going to use the second approach, i.e., using service reference for ease.

Right click the consuming web application,
go to the Add Service Reference option,
Then go ahead with adding the service reference in your consuming application.
I have added a web reference in the consuming application with name `MyProxy`.

So we are all set to call the service from our button click....

As the first step, in the asynchronous button click,
I am going to create the proxy class object and looking at the intelisense, we can see that there are mainly three things related to our web method which we have created.
1 event and two methods,one with an `async` augment.

Here we are going to use the event called `MyTestAsynchronousMethodCompleated` and the web method `MyTestAsynchronousMethodAsync`.

You might think that from where the web method `MyTestAsynchronousMethodAsync` does and the event `MyTestAsynchronousMethodCompleated` came from, right ?
The answer is it will come by default while creating the proxy of the web service.

Then question arises here is
What does this event do?
and
When will the event get fired?

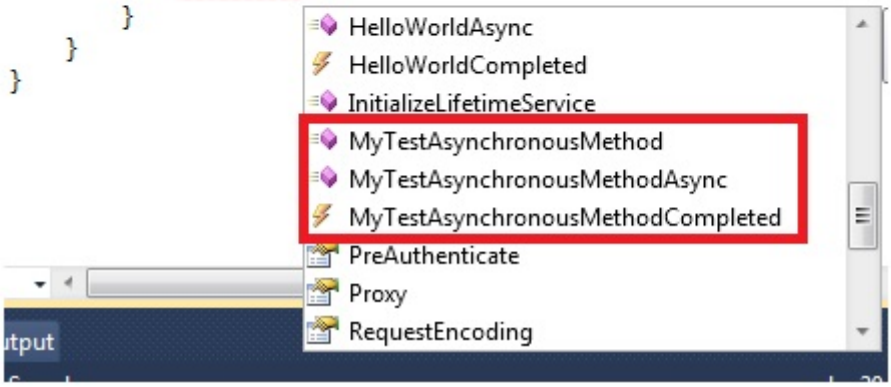
Yes that's a valid question here.
The web method `MyTestAsynchronousMethodAsync` is the hero who makes the asynchronous call happen and the event `MyTestAsynchronousMethodCompleated` is the supporting actor of the hero.
The event will get fired when the asynchronous web method call finishes its execution ...
Hope you get the point.

You can see the event and the method in the below snapshot.

```
namespace AsyncConsumer
{
    public partial class _Default : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void BtnAsync_Click(object sender, EventArgs
        {
            MyProxy.MyAsync objProxy = new MyProxy.MyAsync();
            objProxy.
        }
    }
}
```

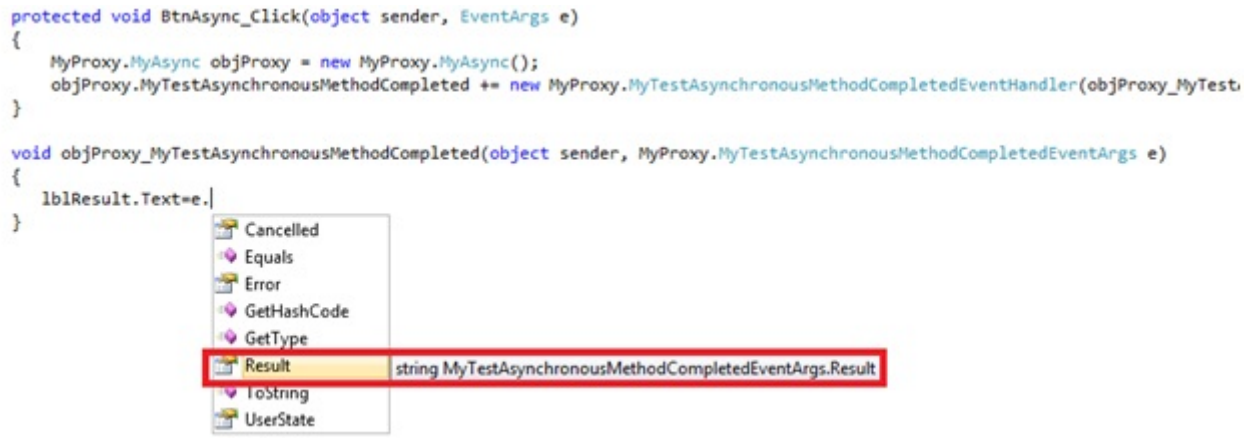


How's that..?

The next thing we have to do is to plumb the event.
As I mentioned earlier, the event is responsible to return the result after the execution of the web service method.
So I am going to set a label in our consuming application in such a way that it will display the returned string from the web service method call..
Ok let's take a look at how it's done..

For that purpose I have added an event handler for that event.
Visual Studio will create the event handler stub automatically for us.
Just type += next to the event which we have and hit Tab key twice in your keyboard ... simple,
You have got your event handler stub with the arguments and the parameters set..
The only thing you have to do is write the business logic inside the event handler stub... That Visual Studio doesn't know, I mean your business logic.

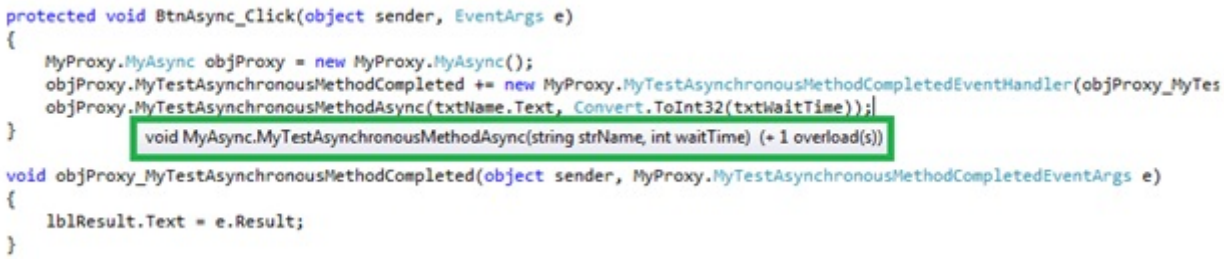
So we have to code in this stub which assigns the result to the label...



You can see a property called `Result` in the event argument.
That's the property that gives you the web method executed `Result`, `Error` etc., in string format, it's not the service method itself as before.

So let's go and implement the logic in the stub by assigning the result to the label text property (you can see this in the below snapshot).
OK now we have even assigned the result..
Did we call the method....?
Noooo right..?

Let's go and call the method..
Which method are we going to call ...?
We are going to call the second web method which is generated automatically.



One thing I would like you to notice is,
we are not assigning the web method result to any variable or object as we usually do..
Did you saw the above screenshot?
The return type of the web method is `void`.
How come we don't assign it to any variable or object..?
It's not possible.
That's why we are getting the result in the web method call completed event as discussed before, in the result property of the event args.

Here a question can arise in your mind, why is it like that..?

The answer is.. we are getting the output only after the execution of the method is completed.
In this scenario once the method is called the main thread will transfer control internally to a child thread,
And it will take care of the execution of the web service method and the main thread will continue with the execution process in our consuming application.

So after completing the web service method execution the event will get fired, and that we will get the result finally..

Hope everything is clear..

C# Shrink ▲ Copy Code

```
namespace AsyncConsumer
{
    public partial class _Default : System.Web.UI.Page
    {
        Stopwatch objSW = new Stopwatch();
```

```
        protected void Page_Load(object sender, EventArgs e)
        {
        }
        protected void BtnAsync_Click(object sender, EventArgs e)
        {
            objSW.Reset();
            objSW.Start();
            MyProxy.MyAsync objProxy = new MyProxy.MyAsync();
            objProxy.MyTestAsynchronousMethodCompleted +=
              new MyProxy.MyTestAsynchronousMethodCompletedEventHandler(
              objProxy_MyTestAsynchronousMethodCompleted);
            objProxy.MyTestAsynchronousMethodAsync(txtName.Text,
              Convert.ToInt32(txtWaitTime.Text));
            DoSomeLongJob();
            lblExecTime.Text = "Total Execution Time :" +
              objSW.ElapsedMilliseconds.ToString() + " Milliseconds";
        }
        void objProxy_MyTestAsynchronousMethodCompleted(object sender,
             MyProxy.MyTestAsynchronousMethodCompletedEventArgs e)
        {
            lblResult.Text = e.Result;
        }
        private void DoSomeLongJob()
        {
            System.Threading.Thread.Sleep(5000);
        }
    }
}
```

Also I have added a method called `DoSomeLongJob().` Which makes the thread to wait for 5 seconds for the consuming application and a Label to display the execution time.
The above is the complete code snippet which we have done..

Let's see the execution of our application..

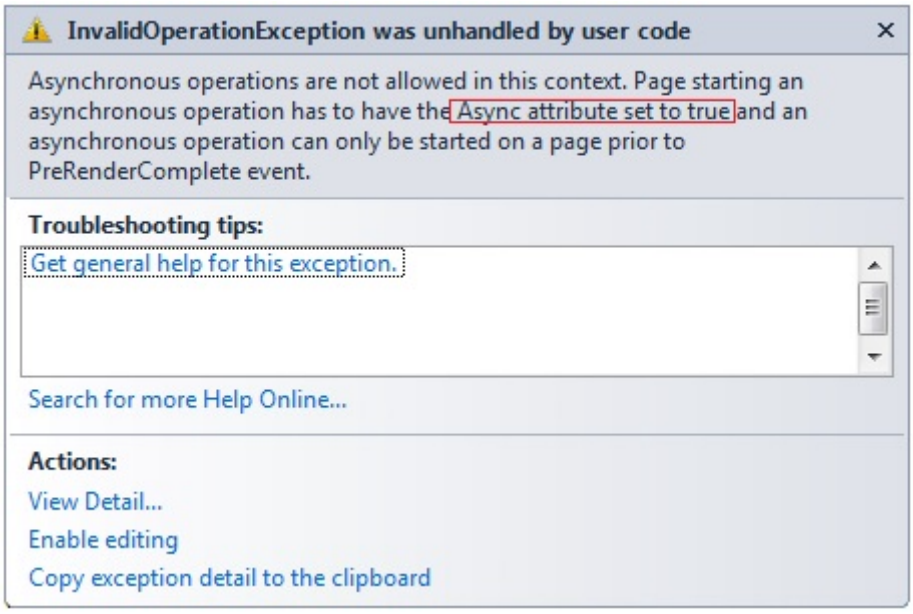
Before that let's try predict the output and execution time...

If we are consuming this as a normal web method and if I am giving input like

What would be the result and execution time?
In a normal case this will take approximately 7 seconds delay, that is 2 seconds which we have given, and the 5 seconds of the `DoSomeLongJob();`
right?

Let's see how long will it take.

Ooopsss!!! Unfortunately or fortunately I got an exception like below while clicking the button:

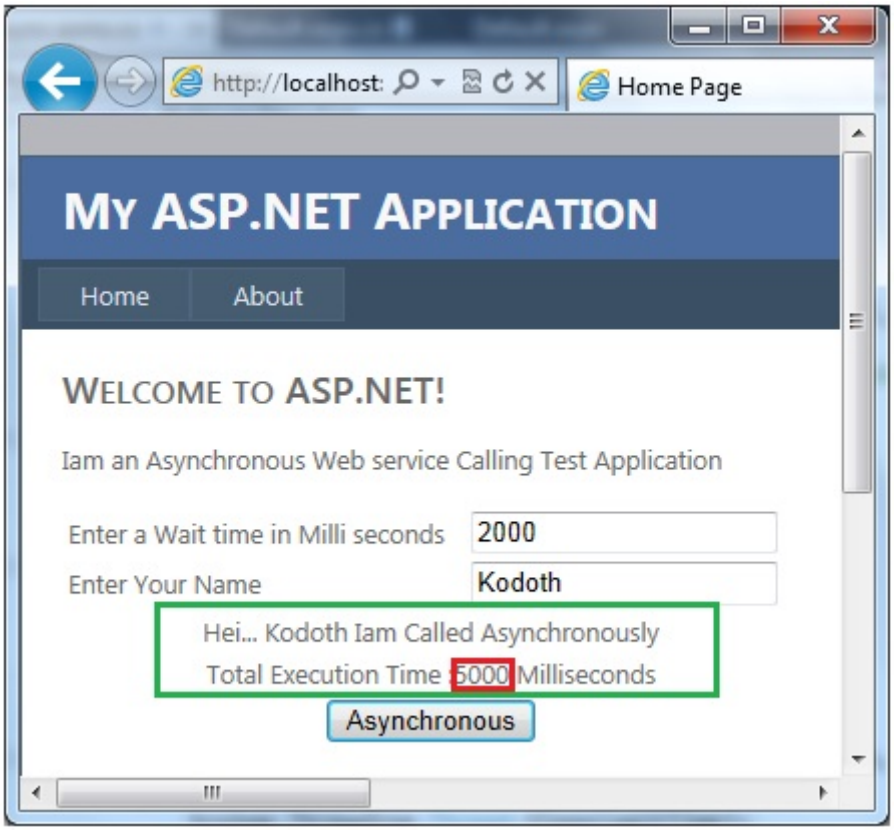


Did you notice the marked portion in the exception?
Yes, we have to add and set the `Async` attribute to `true` in the `Page` directive of the page in which we are planning to have any asynchronous web service method call.
Otherwise it's not possible. Let's go and set it.

After setting the `Async` attribute the directive will be like:

```
<%@ Page Title="Home Page" Language="C#" MasterPageFile="~/Site.master" AutoEventWireup="true"
    CodeBehind="Default.aspx.cs" Inherits="AsyncConsumer._Default" Async="true" %>
```

OK, let's try again ... this time hope we will get the result without any exception..



Wooowwww!!!
See the above snapshot...
Our application executed successfully and look at the execution time...
We achieved this in 5000 milliseconds instead of 7000 milliseconds.
Have saved around 2000 milliseconds execution time and that's really awesome.

In this case if we had used a normal web method calling mechanism definitely it will take 7000 milliseconds and plus ...

Conclusion

We can save a lot of execution time by implementing mechanisms like asynchronous web service calls.
This will be really useful for applications that call lots of web services and do long running jobs.
Time is Money save time save money ...by using asynchronous web method calls.

License

This article, along with any associated source code and files, is licensed under The Code Project Open License (CPOL)

Share

About the Author

KK Kod
Software Developer (Senior)
United States
Watch this Member

Currently working as Application Development Consultant in USA
I love writing what i know about and learning from the mistakes which i make...

Comments and Discussions

Add a Comment or Question | Email Alerts | Search Comments

First Prev Next

Great explanation ... very very clear
Kayun Chantarasathaporn 24-Jun-21 11:06

Problem with an application
Member 15063528 31-Mar-21 16:34

Any thoughts on the sequence of execution?
veenaamburle 15-Dec-15 12:03

How to implement it for ajax calling static webmethod
souvikcode 1-Jul-15 11:35

Re: How to implement it for ajax calling static webmethod
KK Kod 12-Oct-15 16:13

Nice Job
K K Allentown 18-Jul-14 15:18

Sample code??
marc11h 13-May-14 16:39

Re: Sample code??
KK Kod 13-May-14 18:34

Re: Sample code??
humayoun kabir 12-Oct-15 13:57

Re: Sample code??
KK Kod 12-Oct-15 16:10

I created this article in 2013 ,if you are asking me know ...No answer !!!
and ive explained line by line there in this article ,it would be really easy for you to recreate it .
Sorry !!!

Reply · Email · View Thread

How to start a session on .net server from a iOS or Android application?
Member 10046387 26-Feb-14 8:37

Getting Response from Web service
Member 9496050 26-Feb-14 5:01

Helpful
Member 9496050 26-Feb-14 4:56

My vote of 5
coder52 9-Sep-13 21:27

VB Code?
Member 10241176 29-Aug-13 23:33

Re: VB Code?
KK Kod 31-Aug-13 7:58

Re: VB Code?
Member 10241176 3-Sep-13 17:53

Re: VB Code?
Member 10241176 3-Sep-13 20:56

My vote of 5
Hitesh07 2-Aug-13 8:25

Re: My vote of 5
KK Kod 2-Aug-13 9:01

Refresh 1

General News Suggestion Question Bug Answer Joke Praise Rant Admin

Use Ctrl+Left/Right to switch messages, Ctrl+Up/Down to switch threads, Ctrl+Shift+Left/Right to switch pages.

Permalink
Advertise
Privacy
Cookies
Terms of Use

Layout: fixed | fluid

Article Copyright 2013 by KK Kod
Everything else Copyright © CodeProject, 1999-2021

Web03 2.8.20210901.1