

# AJAX based Web Services Invocation: Higher Performance & Bandwidth Utilization

Rohit Dhand <sup>+</sup>

Department of Computer Applications, Lovely Professional University (Phagwara –[PB]. India)

**Abstract.** Invoking Web Services is an integral part of Web 2.0 and SOA (Service Oriented Architecture). Web Services are the base mechanism used to transmit data between the client browser and server. But calling a web service asynchronously instead of performing a complete post back has several advantages, including less network traffic and generally elevated performance. Also with the support of DOM model and XMLHttpRequest Object in major popular browsers, numerous client side AJAX libraries are available to make these calls. The paper examines and compares the typical and Ajax based invocation of web services with remedies to improve the User Experience and Web Server Utilization.

**Keywords:** Web service, WSDL, callback, request-response

## 1. Introduction

Web Services have given Enterprise computing a new dimension for performing business activities. [1,2] They provide interoperability between numerous business processes and also brings automation of these tasks for various organizations. Typically web services were designed and created to provide data to Rich Internet Applications. They can be invoked or called by a number of web sites. Professional web services like Yahoo weather, Stock indexes etc are the best suited examples for the implementation of Web Services. Typically web services were planned to call with the synchronous behavior i.e. the client/user has to wait for the response from the server and cannot continue with his/her work. But with the growth of web based applications over a couple of years and implementation of standards like Web 2.0, Asynchronous invocation of the web services in distributed computing platform has seen a tremendous rise and acceptability in the web based programming. By asynchronous service model, we mean the client can continue working on other tasks without waiting for the result from the server after it calls a service; the service will inform the client about the results of the call at a later point in time.

In the conventional model of synchronous calls, Web Server works on a complete model of Form posting i.e. request-response cycle, which makes lots of pressure on Web server processing, making it more time consuming and degrading the performance of the web server. Over the years the communication speed of the Internet has increased a lot,[3,4] but even then most end users are still facing the pitfalls of twisted wires. So effective utilization of the network is one phenomena to be kept in mind while developing web based applications. This paper examines various such aspects of the web services calling, pitfalls of synchronous calls, Advantages of WS Call-Backs etc. Paper is divided into 4 sections. Part I gives a concise introduction to the web services and their invocation. Part II of the paper examines the meaning and background of the synchronous calls to the web services and how they are typically implemented in ASP.NET applications with C# language.

Part III throws light on the various frameworks which are AJAX based and how they help in reducing the web server load by avoiding complete post backs in the web forms. Part IV gives an insight to the

---

<sup>+</sup> Corresponding author.

*E-mail address:* rohit.dhand@lpu.co.in

implementation of the client side frameworks in web applications to make asynchronous calls.

## 2. Meaning and Background

**Over the years, web services have been called with the traditional methods of invocation i.e. sending** synchronous calls to the web services, in which the complete web form is posted to the server and user has to wait for the result to come back from web service. But, with the development of AJAX,[3,4] asynchronous calls to the web services are made making the user to continue with the work without waiting for the result to come from the invoked service. In simple words:

‘Asynchronous invocation of the web service helps the end user to continue with the work without waiting for the result to transfer back from the web server to the client. This helps in reducing the web server load and reducing the amount of data that travels through the network.’

Different languages and frame works like .NET, J2EE provide the methods to make synchronous calls to the web services that are a part of the Enterprise computing or Service Oriented Architecture. The following example demonstrates the traditional method of making, deploying and calling the web services with ASP.NET and C#.

### Step 1. Creation of the Web Service:

```
using System;
using System.Linq;
using System.Web;
using System.Web.Services;
using System.Web.Services.Protocols;
using System.Xml.Linq;
[WebService(Namespace = "http://tempuri.org/")]
[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
// To allow this Web Service to be called from script, using ASP.NET AJAX, uncomment the following
line.
// [System.Web.Script.Services.ScriptService]
public class Service : System.Web.Services.WebService
{
    public Service () {

        //Uncomment the following line if using designed components
        //InitializeComponent();
    }

    [WebMethod]
    public string HelloWorld() {
        return "Hello World";
    }
    [WebMethod]
    public int GetAllEmployees(String vCity)
    {
        //code to get all the employees from the database with parameter of City
    }
}
```

We can see that the web service contains methods that have attributes of [WebMethods] so that they can be invoked from the other web applications. And also all the web services contains the Business Logic and not the presentation components.

### Step 2: Invoking the Web Service by Synchronous calls

The web service can be invoked by adding the reference of this service in any other web application and making an object to call the same[5]. The reference can be added as a Web Reference. Code to invoke the web service in any Web Form:

```
<WebService Class> obj=new <Web ServiceClass>;
```

```
obj.<MethodName(>;
```

The above method makes the full call to the web service and the user has to wait for the response to come back. This also sends complete data post back to the web server and degrades the performance of the web server. All the data travels to the network, whether the partial update has happened. The following figure shows the Synchronous method of calling web service:

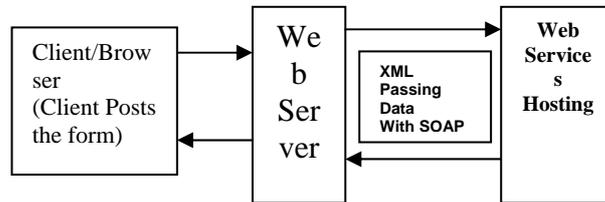


Figure 1. Synchronous Calling of Web Services

The above example shows process of calling web services synchronously putting load on the web server and also sending more of traffic on the net. Also the user is in lock mode i.e. wait till the response comes from the web service.

### 3. Frameworks for Asynchronous Calls

Making synchronous calls to web services can be problematic on occasion, because they have the potential to cause considerable delay. To overcome the necessity of waiting the web service response, we can call web services asynchronously.[5,6] The asynchronous call mechanism does not need any additional functionality for the web service to handle it competently. Numerous frameworks are available like Dojo Toolkit, MooTools, Ajax.NET Professional, ASP.NET AJAX, Saaris, PHPLiveX etc. for the implementation of the asynchronous calls. One of the most vital method of performing asynchronous calls is “Callbacks”[7]. In this approach we create a delegate which handles the results immediately when the client calls the web service. This means the client can continue with the application without any delay. The following code shows how to make the client side calls:

#### //OnClick Event in JavaScript

```
function WebService_Call()
{
//Create instance of Web Service
WebService obj=new WebService();
//Create a delegate that handles the asynchronous result make asynchronous call through the proxy class
obj.Method(param1,CallBackSampleMethod);
//Get the result of the Web Method by defining the delegate used to get the result
}
function CallBackSampleMethod(result)
{
//display the result in the label
if(null!=result)
{
$get("Label1").value=result;
}
}
```

### 4. Implementation

In order to implement the Call Back mechanism, we need some framework that helps to create a proxy of the web service on the client side, so that we can invoke the same through Java Script. Normally the web services are also embedded in the same web project because of the security concerns, but even then we can resort to calling of web service that is not the part of the same web project.

We will take an example of Web Service that helps to retrieve all the customers by the virtue of their city and returns the DataSet. The example is performed on ASP.NET Ajax Extensions with C# with SQL Server

as backend[7]. The following steps describe the complete solution of asynchronous implementation of web services:

### Step 1: Creating the Asynchronous Web Service

```
using System;
using System.Linq;
using System.Web;
using System.Web.Services;
using System.Web.Services.Protocols;
using System.Xml.Linq;
[WebService(Namespace = "http://tempuri.org/")]
[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
// To allow this Web Service to be called from script, using ASP.NET AJAX, uncomment the following
line.
[System.Web.Script.Services.ScriptService]
public class Service : System.Web.Services.WebService
{
    public Service () {
    }
    [WebMethod]
    [ScriptMethod()] >_
    public DataSet GetEmployees(String vCity)
    {
        //code to get all the employees from the database parameter of City
    }
}
```

The above Web Service class has an attribute of [System.Web.Script.Services.ScriptService], that makes the total difference. This helps in making the proxy class when compiled, so that it can be invoked from the client side. Also any method of web service to be invoked from client side in ASP.NET Ajax must have [ScriptMethod()] attribute that helps in invocation of the same from the client without web form postback.[7]

### Step 2: Setting the Configuration to use Proxy in Javascript

```
<asp:ScriptManager ID="ScriptManager1" runat="server">
    <Services>
        <asp:ServiceReference path="~/Service.asmx" />
    </Services>
</asp:ScriptManager>
```

The above code helps in setting a proxy registration of the web service class in the web form, so that it can be invoked from JavaScript asynchronously. Numerous frameworks are available to do the same with little bit of difference in the coding style.

### Step 3: Writing Delegate Code in JavaScript to handle Asynchronous calls

The third step includes the writing of delegate that will handle Callback. This is primarily done at the JavaScript level that makes the client calls without complete request-response cycle.

```
<script language="javascript">
    function CallWebService()
    {
        Service.GetEmploees("XYZ",OnCallBack);
    }
    function OnCallBack(result)
    {
        if(null!=result)
        {
            $get("DataGrid").DataSource=result;
            $get("DataGrid").DataBind();
        }
    }
}
```

</script>

The code contains two methods i.e. CallWebService which is called on the clicking of the button and also initialises a Callback delegate known as “OnCallback”. This delegate is defined as a separate function in JavaScript by the name of OnCallback(result) and it takes one argument called ‘result’. This receives the data from the web service method that has been invoked. The web service is invoked with the client end function and data is returned in the form of DataSet that can be manipulated in any form. The user does not have to wait for the form to post back i.e. complete request-response cycle rather the user continues with the work while the partial update is transferred from the web server to the client. The above implementation explains the web services and their invocation with help of asynchronous calls. This helps in the following: [6,7]

- a. Elevated performance of the web server because the complete web form posting is not done and only partial updates are retrieved from the web server.
- b. Results in higher bandwidth and optimal utilization of the same which enhances the user experience.

## 5. Conclusion

Web services and a service-oriented style of architecture are widely seen as the basis for a new generation of distributed applications and system management tools. This paper introduces the key concepts, benefits of asynchronous calls and methods to invoke web services to improve the overall user experience and higher performance. Paper gives the implementation of the asynchronous based frameworks and implementation example in ASP.NET to justify the concept. With the rising popularity of Web 2.0 and Rich Internet Based Applications, callbacks/asynchronous calls are making the implementation of the web services more powerful and help in improving the functioning of the web server.

## 6. References

- [1] J-Y Chung, K-J Lin, R.G. Mathieu. “Web Services Computing”, In IEEE Computer Vol. 36, Issue 10, pp 38-44, Oct. 2003.
- [2] M. Turner, et al. “Using Web Service Technologies to create an Information Broker”. In Proceedings of the IEEE International Conference on Software Engineering (ICSE), pp. 552-563, 2004.
- [3] Y. Goland, M. Nottingham, D. Orchard (BEA). “WSCallBack Protocol (WS-Callback) 0.91.”, [http://dev2dev.bea.com/technologies/webservices/WSCallBack0\\_9.jsp](http://dev2dev.bea.com/technologies/webservices/WSCallBack0_9.jsp)
- [4] M. Gudgin, M. Hadley. “Web Services Addressing 1.0 – Core”, <http://www.w3.org/TR/ws-addr-core>, Feb. 2005. N. Kavantzias, et al. “Web Services Choreography Description Language Version 1.0”, W3C Working Draft 17 December 2004.
- [5] T. Andrews, et al. “BPEL4WS Specification”, <http://www-106.ibm.com/developerworks/webservices/library/ws-bpel/>, May 2003.
- [6] S. Dalal, et al. “BTP, Version 1.0.9.5”, [http://docs.oasisopen.org/business\\_transaction](http://docs.oasisopen.org/business_transaction), Nov. 2004. [8] K. Brown, et al, “Web Service Polling (WS-Polling)”, W3C Member Submission, <http://www.w3.org/Submission/ws-polling/>, Oct. 26, 2005.
- [7] M. Ruth, F. Lin and S. Tu. “A Client-side Framework Enabling Callbacks from Web Services”, In Proceedings of the European Conference on Web Services (ECOWS), pp. 105-116, 2005.