



ASP.NET Technique: Invoking Page Methods with jQuery

By Daniele Bochicchio, Stefano Mostarda, and Marco De Sanctis, authors of *ASP.NET 4.0 in Practice*

Sometimes you need a method only in one page. For these situations, placing it in a web service might be useless. In this article, based on chapter 12 of ASP.NET 4.0 in Practice, the authors show you how you can use Ajax with jQuery to create a method in the page that requires it and then expose it to the client.

It's quite likely that you need the total orders amount calculation only in the page that shows the customer. Placing the method that calculates this amount in a web service is perfectly valid, but placing it only in the page that requires it might be a good idea, too. In ASP.NET terminology, such a method is called a *page method*.

Problem

We need to create a method that calculates the total orders amount. Such a method must not live in a web service, but only in the page that uses it. This method must be available to the client.

Solution

A page method is a method, just like all the others. It lives in the page class, and it *must* be static and marked with the `WebMethod` attribute (`System.Web.Services` namespace). The following listing shows the code for this method.

Listing 1 Invoking the server using jQuery API for Ajax

```
C#:
[WebMethod]
public static decimal GetOrdersAmount(string CustomerId)
{
    using (var ctx = new NorthwindEntities())
    {
        return ctx.Orders.Where(o => o.CustomerID == CustomerId)
            .Sum(o => o.Order_Details.Sum(d => d.UnitPrice * d.Quantity));
    }
}

VB:
<WebMethod> _
Public Function GetOrdersAmount(CustomerId As String) As Decimal
    Using ctx = New NorthwindEntities()
        Return ctx.Orders.Where(Function(o) _
            o.CustomerID = CustomerId).Sum(Function(o) _
            o.Order_Details.Sum(Function(d) d.UnitPrice * d.Quantity))
    End Using
End Function
```

The method is marked with the proper attribute and is static. The method is now available to the client and can be invoked using JavaScript code:

```
$.ajax({
    url: "page.aspx/GetOrdersAmount",
```

For source code, sample chapters, the Online Author Forum, and other resources, go to <http://mannings.com/bochicchio/>

```
data: '{ "CustomerId": "ALFKI" }',
type: "POST",
contentType: "application/json",
dataType: "json",
success: function (result) {
    //code
}
});
```

The `url` parameter consists of the page name, plus the `/` character and the method name (it's similar to the web service URL); the rest remains identical. This code wasn't difficult at all!

Discussion

The decision to place spare methods only where they belong is a good one. The problems occur when such a method must be used in other pages, too. The method can be duplicated in each page class, the client code for all pages can invoke the method in the original page, or you can move the method into a web service. The last choice is definitely our favorite because each time you have a *common* method, it's best to place it in a *common* place.

Here are some other Manning titles you might be interested in:



[Windows 8 Apps with HTML5 and JavaScript](#)

Dan Shultz, Joel Cochran, and James Bender



[Fast ASP.NET Websites](#)

Dean Alan Hume



[Windows Store App Development: C# and XAML](#)

Pete Brown

Last updated: August 2, 2013

For source code, sample chapters, the Online Author Forum, and other resources, go to <http://manning.com/groves/>