

Technique for centralized web service authentication, logging, etc.

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.aspnet/2006-08/msg02411.html>

- *From:* rshekhtm@xxxxxxxxxx
 - *Date:* 22 Aug 2006 12:13:31 -0700
-

Hi everyone,

I would like to get your opinion on a technique I came up with when faced with the problem of redundant code in every web method (authentication, logging, exception handling). Normally, my web methods would look something like this:

```
[WebMethod][SoapHeader("AuthHeader")]
public ReturnType GetSomeData(SomeType param1)
{
    try
    {
        // Authentication

        // Implementation
    }
    catch(Exception ex)
    {
        // Exception handling and logging
    }
    finally
    {
        // Request logging
    }
}
```

```
[WebMethod][SoapHeader("AuthHeader")]
public ReturnTypeArray GetMoreData(Type1 param1, Type2 param2, Type3
param3)
{
    try
    {
        // Authentication

        // Implementation
    }
    catch(Exception ex)
```

Technique for centralized web service authentication, logging, etc.

```
{  
// Exception handling and logging  
}  
finally  
{  
// Request logging  
}  
}
```

It gets rather tiresome to copy–paste all this non–implementation code, never mind the fact that the DRY principle (Don't Repeat Yourself) is violated. The only suggestions to avoid this that I have seen include writing HTTP handlers or SOAP extensions. Yikes – very ugly, IMHO.

I have come up with a method that uses delegates coupled with a centralized ExecuteRequest function that takes care of this non–implementation functionality and works with any number of parameters (FYI – in my case, web methods always return a value, but I'm sure it wouldn't be too hard to make void methods work).

It reminds me of the Template Method pattern, except instead of overridden methods, we have delegates to perform the work inside of a method that provides the code template. Have a look:

```
private delegate ReturnType GetSomeDataImplDelegate(SomeType param1);  
private delegate ReturnTypeArray GetMoreDataImplDelegate(Type1 param1,  
Type2 param2, Type3 param3);  
private MulticastDelegate _func;
```

```
[WebMethod][SoapHeader("AuthHeader")]  
public ReturnType GetSomeData(SomeType param1)  
{  
_func = new GetSomeDataImplDelegate(GetSomeDataImpl);  
return (ReturnType)ExecuteRequest(param1);  
}
```

```
[WebMethod][SoapHeader("AuthHeader")]  
public ReturnTypeArray GetMoreData(Type1 param1, Type2 param2, Type3  
param3)  
{  
_func = new GetMoreDataImplDelegate(GetMoreDataImpl);  
return (ReturnTypeArray)ExecuteRequest(param1, param2, param3);  
}
```

```
private ReturnType GetSomeDataImpl(SomeType param1)  
{  
// Implementation  
}
```

```
private ReturnTypeArray GetMoreDataImp
```