



Experience a new world of interaction

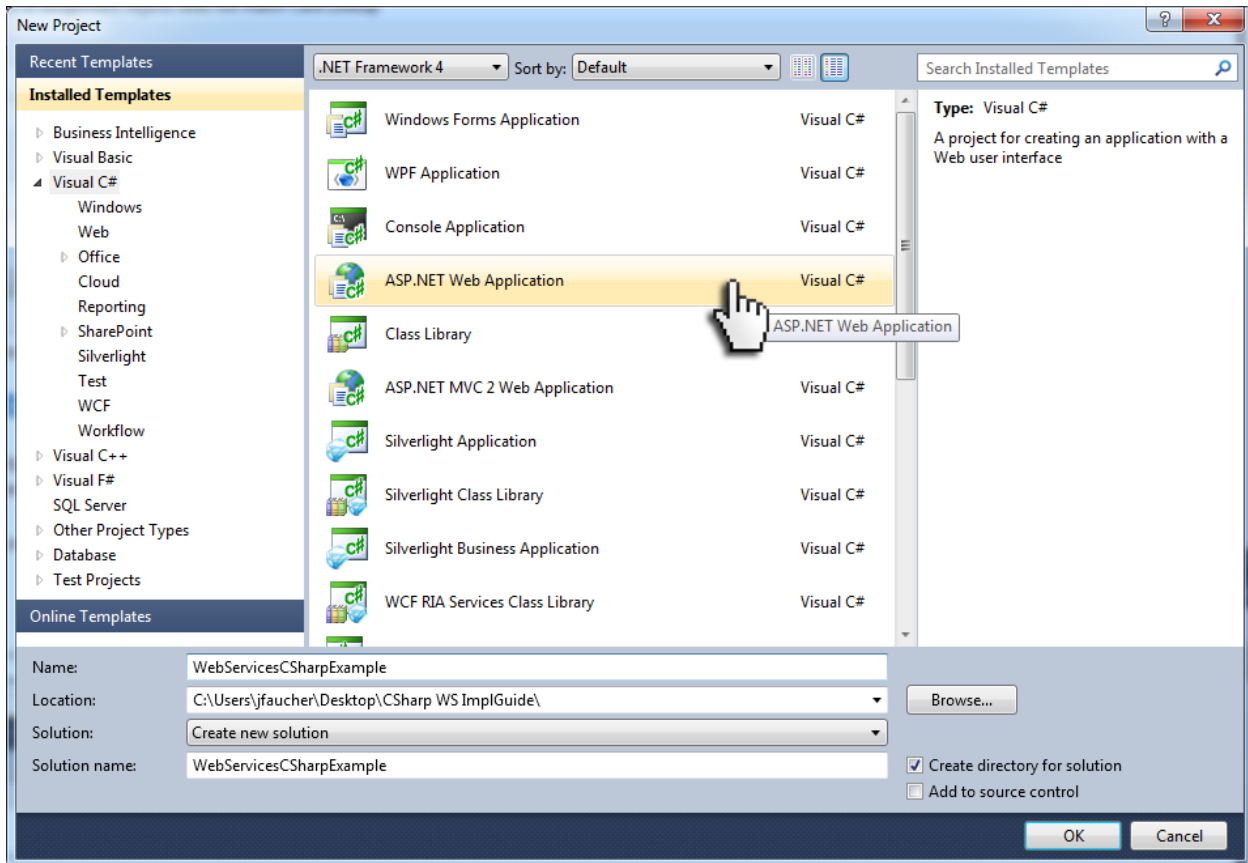
**Aloha Stored Value Web Services
Client Implementation Guide
*With C# .NET Examples***

C# .Net Web Service Client Notes

This document contains some very high level notes on creating a C# .NET Web Service client with Visual Studio 2010.

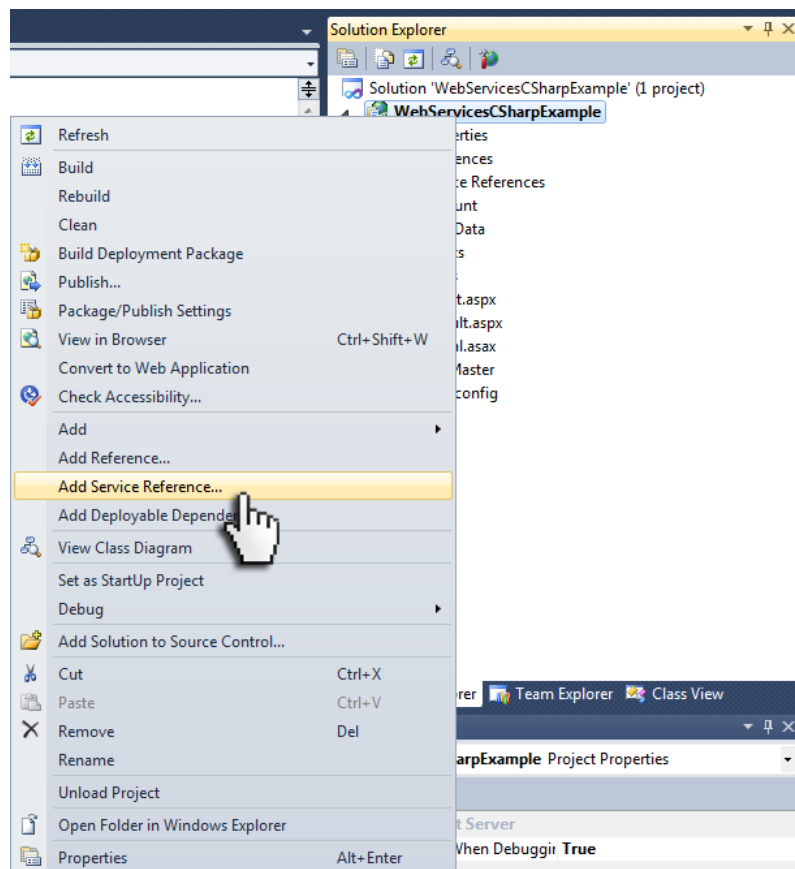
Create a Project

Choose the ASP.NET Web Application. Change the name and location as you see fit.

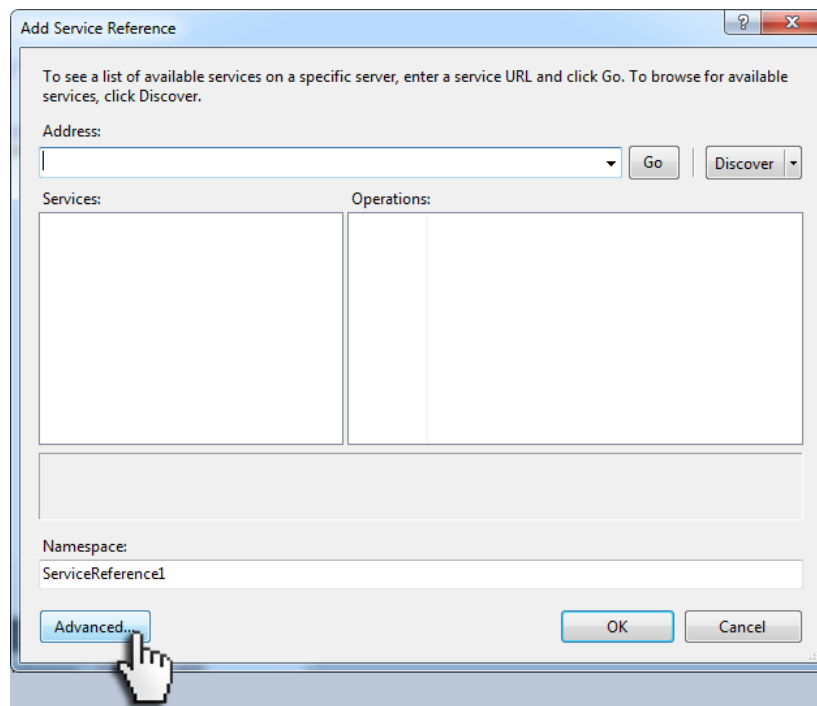


Add a Web Reference

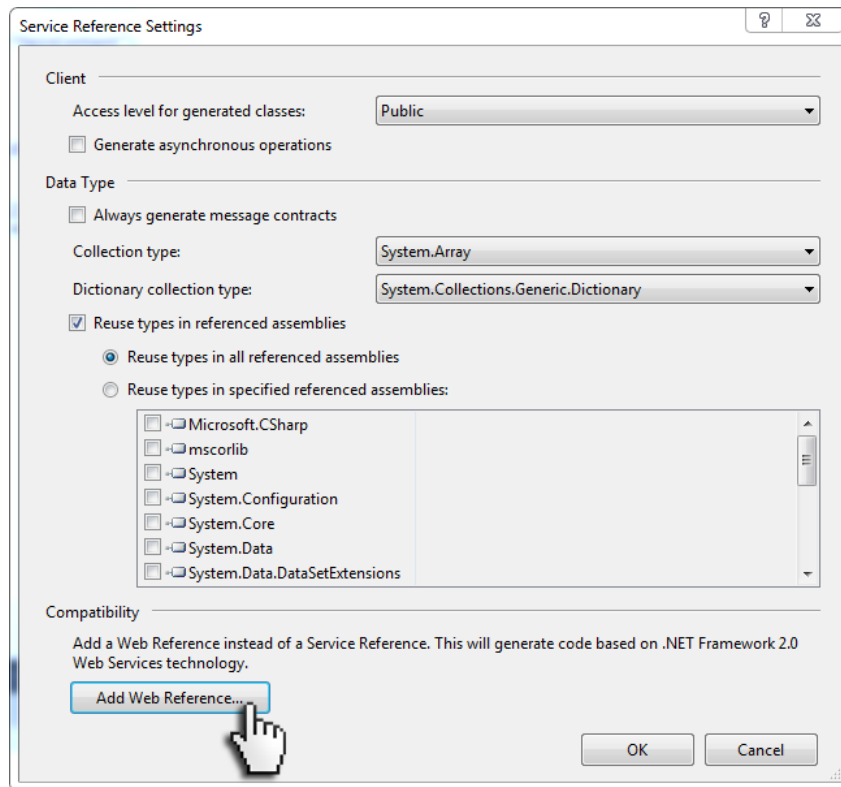
Right click on the project and select *Add Service Reference...*



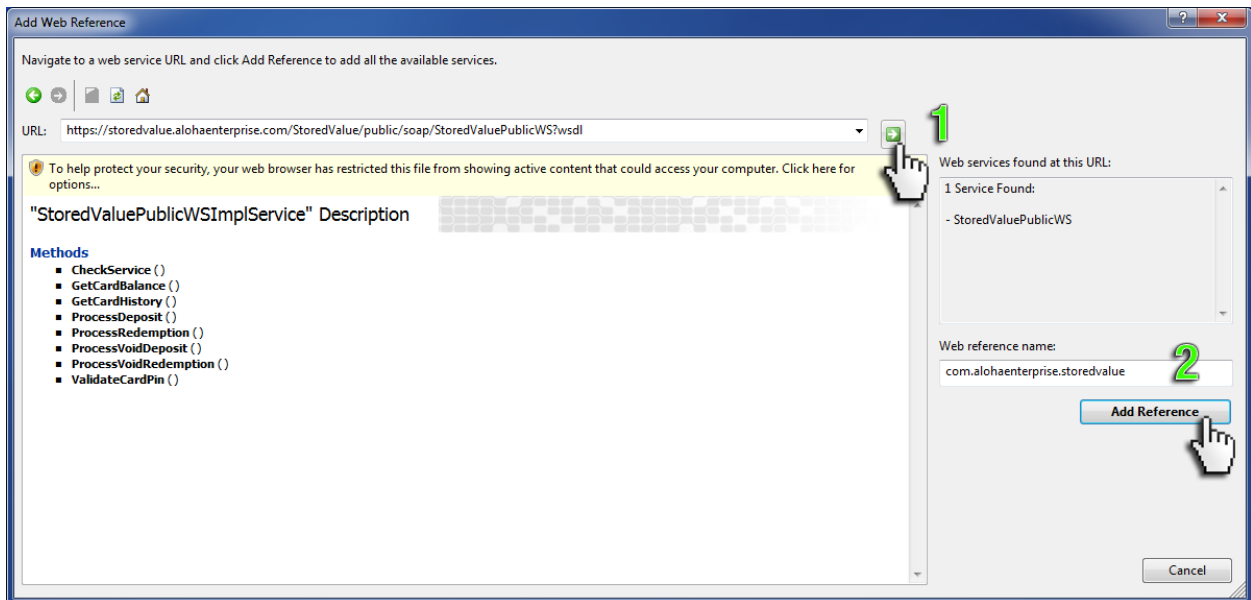
Click on *Advanced* at the bottom of the dialog box.



Click on *Add Web Reference...*

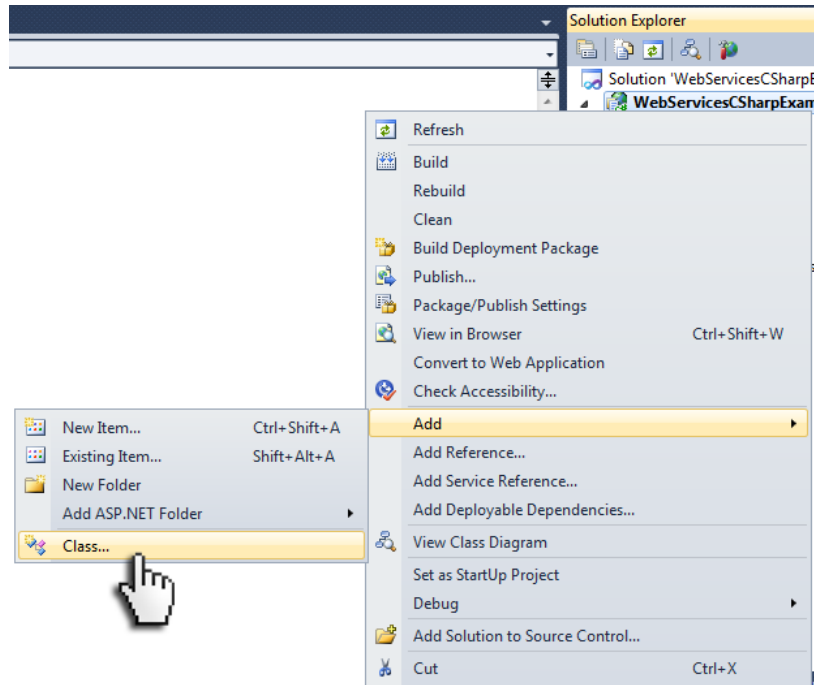


Enter the following in the *URL:* field,
'https://storedvalue.alohaenterprise.com/StoredValue/public/soap/StoredValuePublicWS?wsdl'.
Then click on the green arrow, which is labeled as 1. Once VS has populated the box with the
StoredValuePublicWSImplService Description, click on *Add Reference* labeled as 2.

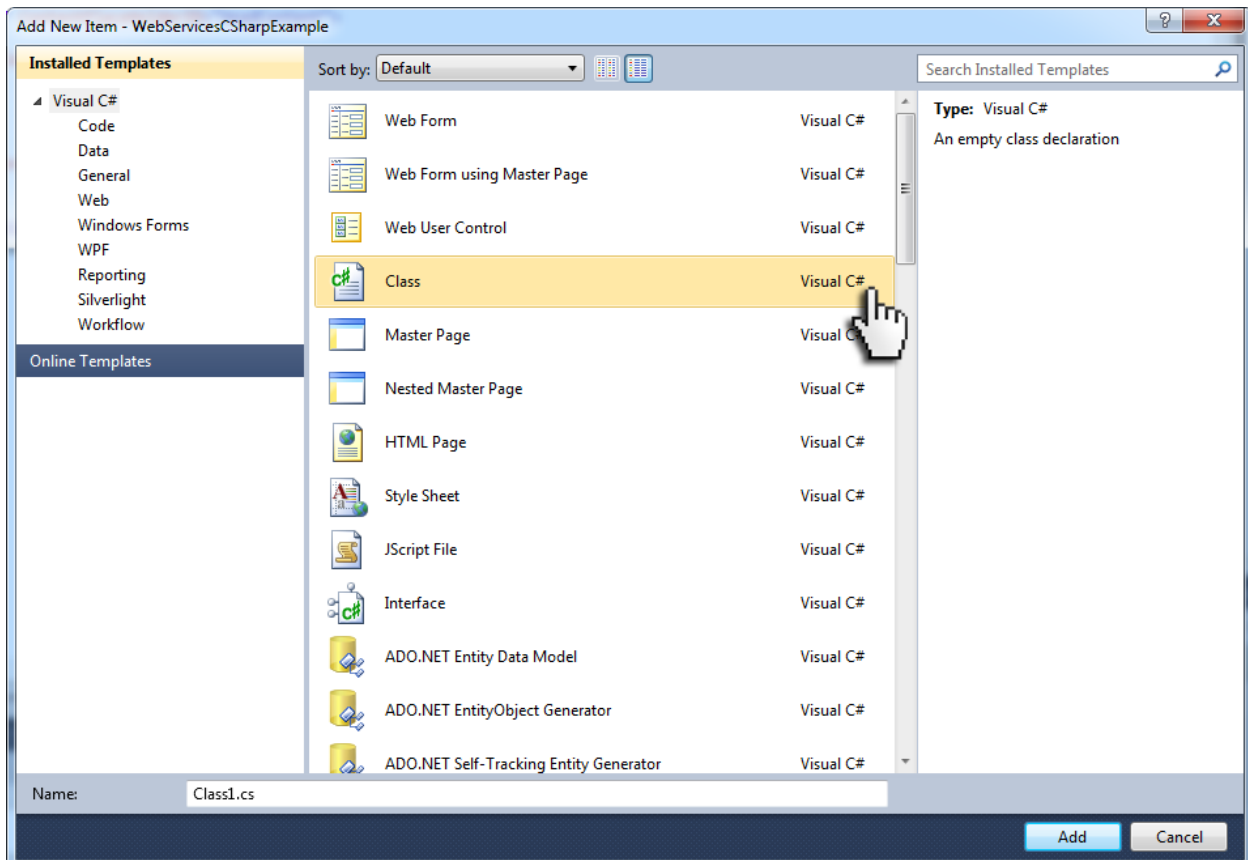


Add a C# Class

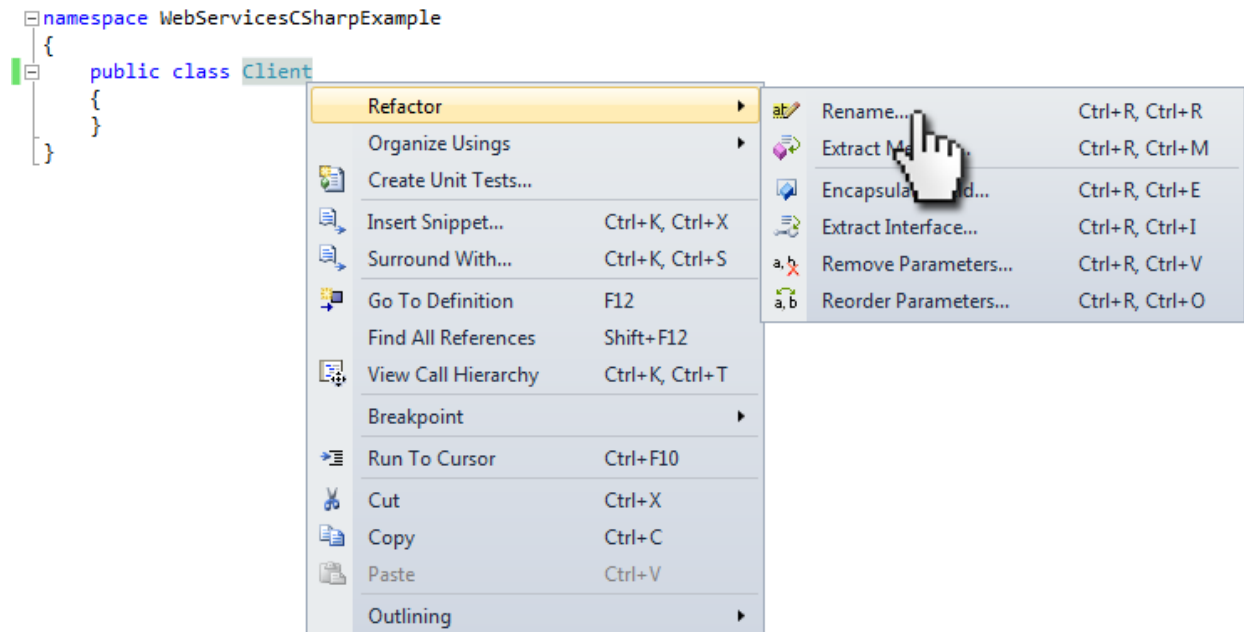
Right click on the project and select *Add* then *Class...*



Then choose the Visual C# class.



You can name the class in the *Name:* field at the bottom of *Add New Item* screen or rename the class by right clicking on the name and then click on *Refactor -> Rename*.



This class is where your C# code will go. The example code is written below. Some of the names may change depending on how you name your classes.

The *Default.aspx* file is the main web page that displays the results in the example. That code is written after the C# example code.

Example Web Access Method (C#)

Note: All instances of “?????” must be replaced with values appropriate to your setup.

Note: The ‘using’ path for *com.alohaenterprise.storedvalue* may vary slightly based on user input during the Web Reference creation step and the name of your project.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

using WebServicesCSharpExample.com.alohaenterprise.storedvalue; //Namespace to shorten the name of the function calls

namespace WebServicesCSharpExample
{
    public class Client
    {
        public static GetCardBalanceResponse getCardStatus(string cardNumber)
        {
            //WebService object and URL
            StoredValuePublicWSImplService wsObject = new StoredValuePublicWSImplService();
            wsObject.Url = "https://storedvalue.alohaenterprise.com/StoredValue/public/soap/StoredValuePublicWS";

            try
            {
                //Setup a request to return a response
                GetCardBalanceRequest cbRequest = new GetCardBalanceRequest();

                //Company variables
                cbRequest.company = "?????"; //String
                cbRequest.storeID = ?????; //Integer
                cbRequest.user = "?????"; //String
                cbRequest.password = "?????"; //String
                cbRequest.requestID = Guid.NewGuid().ToString(); //Random transaction ID

                //User input variables
                cbRequest.cardNumber = cardNumber; //String

                return wsObject.GetCardBalance(cbRequest); //GetCardBalance requires a GetCardBalanceRequest
            }
            catch (System.Web.Services.Protocols.SoapException e)
            {
                Console.WriteLine("exception=" + e);
                Console.WriteLine("exception=" + e.InnerException);
                Console.WriteLine("exception=" + e.Message);
                throw e;
            }
        }
    }
}
```

```
} //End of GetCardBalanceResponse
```

```
public static ProcessDepositResponse processDeposit(string cardNumber, decimal amount, int quantity)
{
    //WebService object and URL
    StoredValuePublicWSImplService wsObject = new StoredValuePublicWSImplService();
    wsObject.Url = "https://storedvalue.alohaenterprise.com/StoredValue/public/soap/StoredValuePublicWS";

    try
    {
        //Setup a request to return a response
        ProcessDepositRequest pdRequest = new ProcessDepositRequest();

        //Company variables
        pdRequest.company = "?????";
        pdRequest.storeID = "?????";
        pdRequest.user = "?????";
        pdRequest.password = "?????";
        pdRequest.requestID = Guid.NewGuid().ToString();

        //User input variables
        pdRequest.cardNumber = cardNumber; //String
        pdRequest.amount = amount; //Decimal that must have an m with the number (i.e. 10.25m)
        pdRequest.quantity = quantity; //Integer

        return wsObject.ProcessDeposit(pdRequest); //ProcessDeposit requires a ProcessDepositRequest
    }
    catch (System.Web.Services.Protocols.SoapException e)
    {
        Console.WriteLine("exception=" + e);
        Console.WriteLine("exception=" + e.InnerException);
        Console.WriteLine("exception=" + e.Message);
        throw e;
    }
} //End of ProcessDepositResponse
```

```
public static ProcessRedemptionResponse processRedemption(String cardNumber,Decimal amount)
{
    //WebService object and URL
    StoredValuePublicWSImplService wsObject = new StoredValuePublicWSImplService();
    wsObject.Url = "https://storedvalue.alohaenterprise.com/StoredValue/public/soap/StoredValuePublicWS";

    try
    {
        //Setup a request to return a response
        ProcessRedemptionRequest prRequest = new ProcessRedemptionRequest();
```



```
//Company variables
prRequest.company = "?????";
prRequest.storeID = "?????";
prRequest.user = "?????";
prRequest.password = "?????";
prRequest.requestID = Guid.NewGuid().ToString();

//User input variables
prRequest.cardNumber = cardNumber; //String
prRequest.amount = amount; //Decimal that must have an m with the number (i.e. 10.25m)

return wsObject.ProcessRedemption(prRequest); //ProcessRedemption requires a ProcessRedemptionRequest
}
catch (System.Web.Services.Protocols.SoapException e)
{
    Console.WriteLine("exception=" + e);
    Console.WriteLine("exception=" + e.InnerException);
    Console.WriteLine("exception=" + e.Message);
    throw e;
}
} //End of ProcessRedemptionResponse
} //End of Class
}
```

Example ASPX

Note: All instances of "xxxxx" must be replaced with a valid card series ID.

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

<%@ Import Namespace="WebServicesCSharpExample" %>
<%@ Import Namespace="WebServicesCSharpExample.com.alohaenterprise.storedvalue" %>

<asp:Content ID="HeaderContent" runat="server" ContentPlaceHolderID="HeadContent">
  <style type="text/css">
    td {padding:4px;}
    table {width:500px;border:solid;border-collapse:collapse;}
  </style>
</asp:Content>

<asp:Content ID="BodyContent" runat="server" ContentPlaceHolderID="MainContent">
  <h2>
    Sample Stored Value Web Service Client
  </h2>
  <br/>
  <!-- Get Card Balance Example -->
  <%
    GetCardBalanceResponse csResult = new GetCardBalanceResponse(); //csResult is the object that holds the web service response
    csResult = Client.getCardStatus("xxxxx00000001"); //function call with the required parameters
  %>

  <table>
    <colgroup><col width="150" /><col width="350" /></colgroup>
    <tr><th colspan="2"><center>GetCardStatusResult</center></th></tr>
    <tr><td>message</td> <td><%=csResult.message%></td></tr> <!-- The first table is the status of the response -->
    <tr><td>responseCode</td> <td><%=csResult.responseCode%></td></tr>
    <tr><td>responseCodeName</td> <td><%=csResult.responseCodeName%></td></tr>
  </table>

  <table>
    <colgroup><col width="150" /><col width="350" /></colgroup> <!-- The second table is the elements of the response -->
    <tr><td>cardNumber</td> <td><%=csResult.cardNumber%></td></tr>
    <tr><td>balance</td> <td><%=csResult.balance%></td></tr>
  </table>
  <br />

  <!-- Process Deposit Example -->
  <%
    ProcessDepositResponse pdResult = new ProcessDepositResponse();
    pdResult = Client.processDeposit("xxxxx00000001",10.25m,2); //This function requires a decimal (i.e. 10.25m) and an int
```

```
%>
```

```
<table>  
  <colgroup><col width="150" /><col width="350" /></colgroup>  
  <tr><th colspan="2"><center>ProcessDepositResult</center></th></tr>  
  <tr><td>message</td>    <td><%=pdResult.message%></td></tr>  
  <tr><td>responseCode</td>    <td><%=pdResult.responseCode%></td></tr>  
  <tr><td>responseCodeName</td>    <td><%=pdResult.responseCodeName%></td></tr>  
</table>
```

```
<% if (pdResult.cardTransaction != null)  
  {  
    int sizeOfPDRResult = pdResult.cardTransactions.Count(); //ProcessDeposit can return multiple cardTransactions.  
    for (int i = 0; i < sizeOfPDRResult; i++) //This code accounts for that possibility.  
    {  
      Response.Write("<table>");  
      Response.Write("<colgroup><col width='150' /><col width='350' /></colgroup>");  
      Response.Write("<tr><td>cardNumber</td>    <td>" + pdResult.cardTransactions[i].cardNumber + "</td></tr>");  
      Response.Write("<tr><td>balance</td>    <td>" + pdResult.cardTransactions[i].balance + "</td></tr>");  
      Response.Write("</table>");  
    }  
  }  
</%>
```

```
%>
```

```
<br />
```

```
<!-- Process Response Example -->
```

```
<%  
  ProcessRedemptionResponse prResult = new ProcessRedemptionResponse();  
  prResult = Client.processRedemption("xxxxx00000001",5.75m); //This function call requires a decimal (i.e. 5.75m)  
</%>
```

```
<table>  
  <colgroup><col width="150" /><col width="350" /></colgroup>  
  <tr><th colspan="2"><center>ProcessRedemptionResult</center></th></tr>  
  <tr><td>message</td>    <td><%=prResult.message%></td></tr>  
  <tr><td>responseCode</td>    <td><%=prResult.responseCode%></td></tr>  
  <tr><td>responseCodeName</td>    <td><%=prResult.responseCodeName%></td></tr>  
</table>
```

```
<table>  
  <colgroup><col width="150" /><col width="350" /></colgroup>  
  <tr><td>cardNumber</td>    <td><%=prResult.cardNumber%></td></tr>  
  <tr><td>balance</td>    <td><%=prResult.balance%></td></tr>  
  <tr><td>amount</td>    <td><%=prResult.amount%></td></tr>  
</table>
```

```
<br />
```

```
</asp:Content>
```

Example Output

This is the expected output of a "Successful Response." You must supply the C# code with the proper credentials and a valid 14 digit card number to get successful execution of the web service methods.

My ASP.NET APPLICATION

Home About

SAMPLE STORED VALUE WEB SERVICE CLIENT

GetCardStatusResult	
message	Get card balance successful
responseCode	0
responseCodeName	SUCCESS
cardNumber	xxxxx000000001
balance	61.75

ProcessDepositResult	
message	Process deposit successful
responseCode	0
responseCodeName	SUCCESS
cardNumber	xxxxx000000001
balance	72.00
cardNumber	xxxxx000000002
balance	80.75

ProcessRedemptionResult	
message	Process redemption successful
responseCode	0
responseCodeName	SUCCESS
cardNumber	xxxxx000000001
balance	66.25
amount	5.75