

VevoCart

Version 5.1.0

Web Service Guide

VevoCart.com

support@vevocart.com

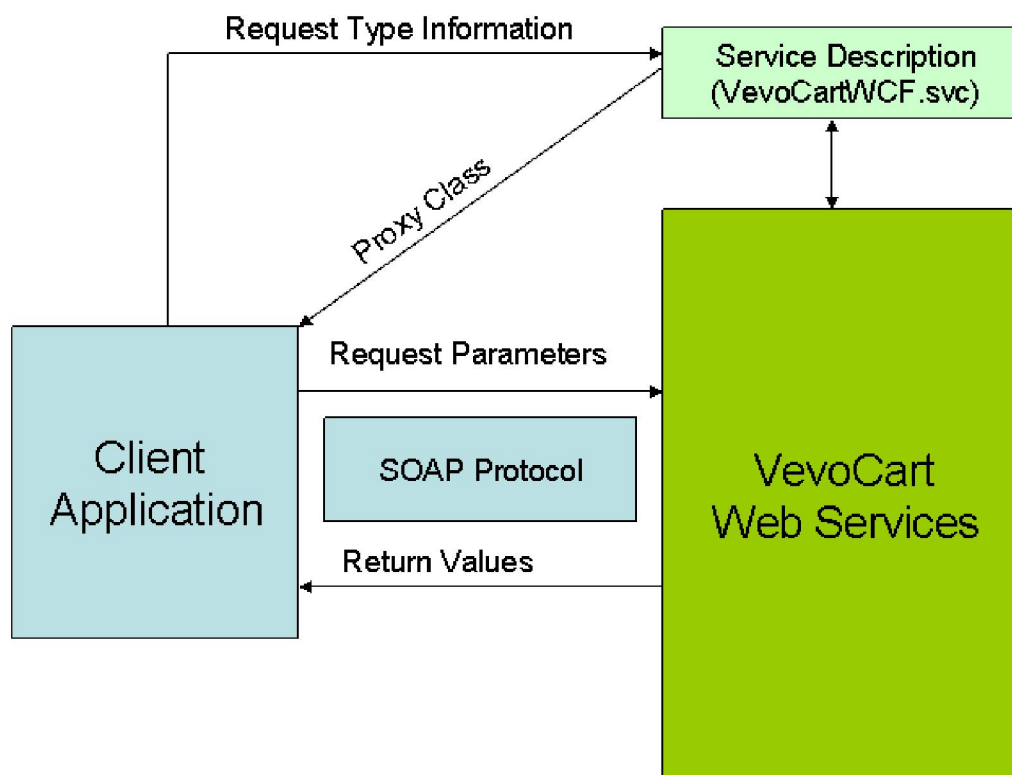
www.vevocart.com

Table of Contents

1. Overview	2
2. Web Service Configuration	2
3. Using VevoCart's Web Service in Microsoft Visual Studio	3
3.1 Add Web Reference	3
3.2 Using VevoCart web service in project.....	4
3.3 View Web Service Details.....	4
4. Web Service API Description.....	6
4.1 Security Service.....	6
4.2 Store Service	6
4.3 Cache Service	7
4.4 Category Service	7
4.5 Department Service	8
4.6 Configuration Service	9
4.7 Culture Service	9
4.8 Customer Service	10
4.9 Info Service	11
4.10 Order Service	12
4.11 Product Service	15

1. Overview

Web Service is service system which is designed to support communication between software-to-software via network protocol and using XML to transfer information. VevoCart's service is developed to support developers who want to communicate with store website via SOAP protocol. It can help developers to develop specific software service to support special requirement and don't want to modify VevoCart original source code.



For more information about basic web service, you can see in:

<http://www.w3schools.com/webservices/default.asp>

2. Web Service Configuration

Before using VevoCart's web service, you need to set up web service's configuration because developer must log in by only specified account while using service in another application.

Go to Configuration menu of setting tab in admin section. After that, please select other tab and enter admin account name what you want to associate with web service.



A dialog box titled "Web Service" with a question mark icon. It contains a label "Web Service Admin Account" and a text input field containing the word "admin".

3. Using VevoCart's Web Service in Microsoft Visual Studio

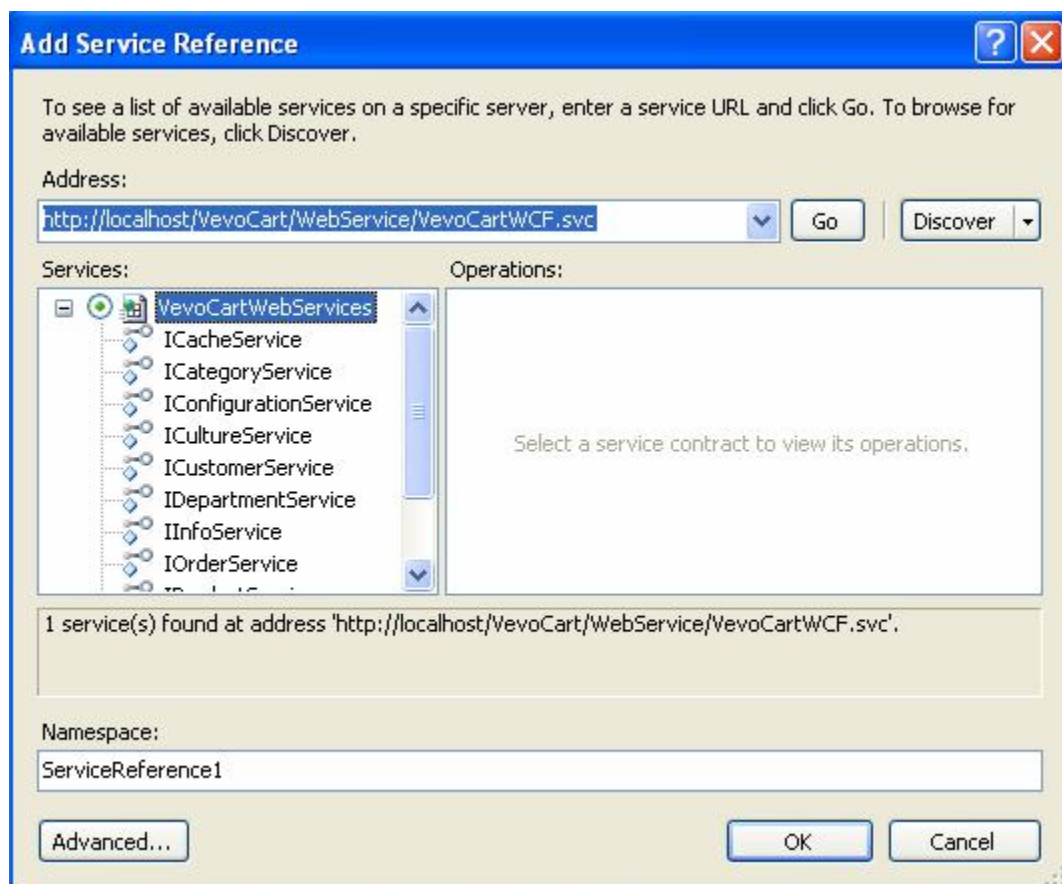
If you use Microsoft Visual Studio in development process, you can use this topic to enhance VevoCart's web service to your application.

3.1 Add Web Reference

You need to add web service reference to the project first. Go to the Solution Explorer tab and select the project what you want and right click on the References and select Add Service Reference. When a dialog pops up, you should enter location of VevoCart's web service to locate it and click Go.

The location of VevoCart web service (VevoCartWCF.svc) is in "WebService" sub folder.

Example: <http://www.yourdomain.com/WebService/VevoCartWCF.svc>.



A screenshot of the "Add Service Reference" dialog box in Visual Studio. The "Address" field contains the URL "http://localhost/VevoCart/WebService/VevoCartWCF.svc". The "Services" list on the left shows "VevoCartWebServices" selected, with a tree view below it listing various interfaces like ICacheService, ICategoryService, etc. The "Operations" pane on the right is empty with the text "Select a service contract to view its operations." A status bar at the bottom of the dialog says "1 service(s) found at address 'http://localhost/VevoCart/WebService/VevoCartWCF.svc'." The "Namespace" field contains "ServiceReference1". Buttons for "Go", "Discover", "Advanced...", "OK", and "Cancel" are visible.

If location is correct, Services box will show interfaces. Enter namespace name what you want to define in project and click OK to finish add web service reference process.

3.2 Using VevoCart web service in project

Developer can use VevoCart's web service in project easily. Because after add web service reference, Visual studio will generate proxy classes which can be replaced XML communication. So you can call methods of web service via proxy classes and no need to understand XML format for transfer data between client and server.

After adding VevoCart's web service into your project, you need to reference service's namespace to source code where you want.

```
using YourProjectNamespace.WebServiceNamespace;
```

Before calling any methods of web service, you need to authenticate security at first.

```
SecurityServiceClient securityService = new SecurityServiceClient();
string ticket = securityService.Login( "admin", "admin" );
```

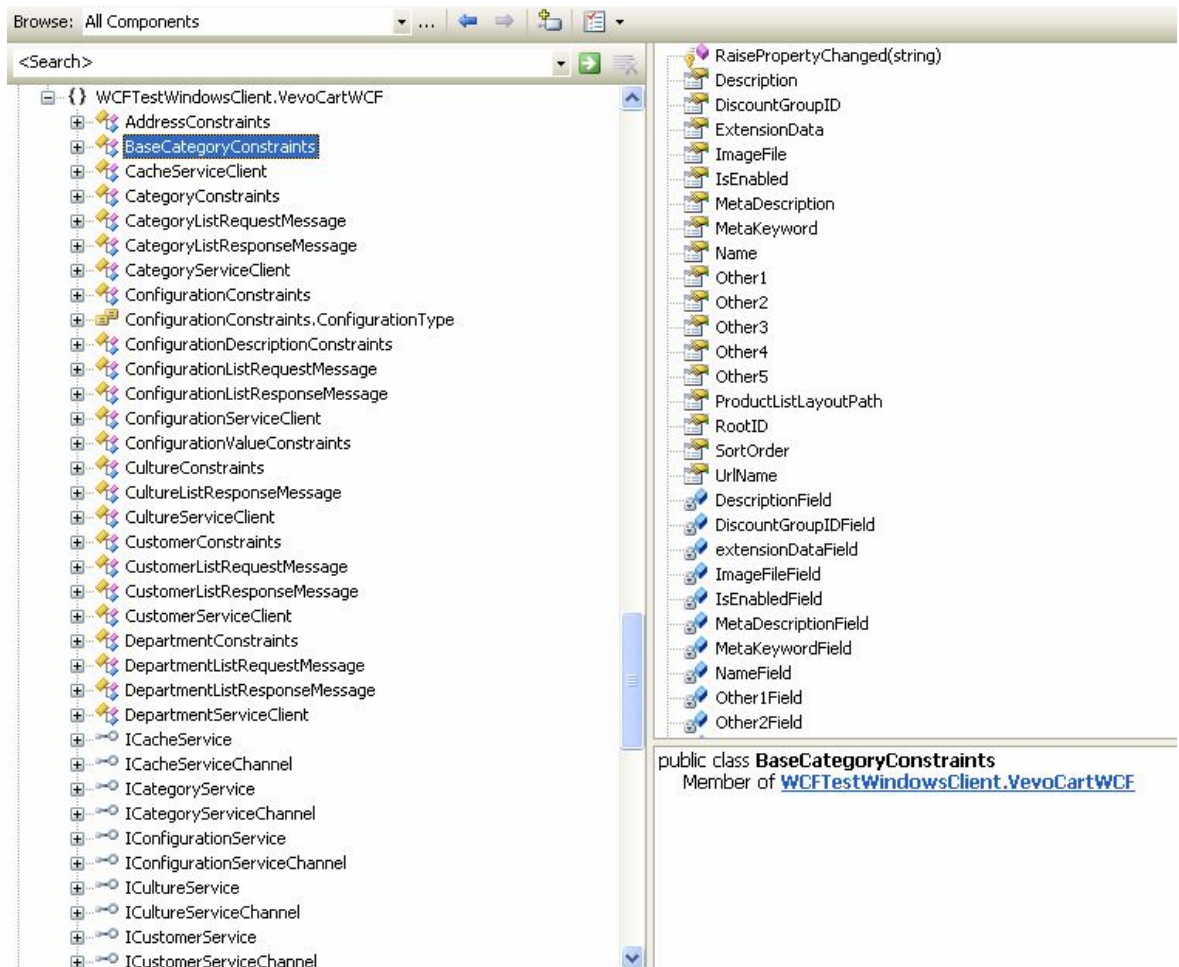
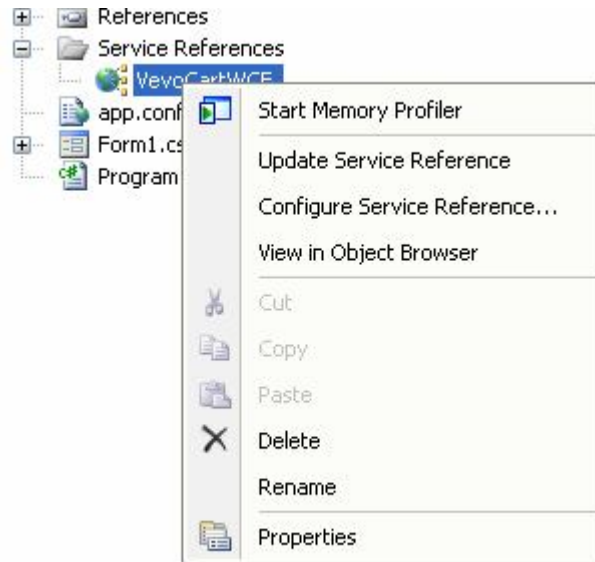
You need to login into web service security by admin account which is specified in VevoCart's configuration. If user name and password are correct, ticket will be returned and you can use it as the key for any methods of service.

When finish calling methods, you need to log out by:

```
securityService.Logout( ticket );
```

3.3 View Web Service Details

When you add VevoCart's web service into project already, you can see details of web service by viewing in Object Browser as normal reference. Right click on reference which is in Service Reference menu and select "View in Object Browser" to open dialog.



4. Web Service API Description

4.1 Security Service

This is the service which is called at first time because other services need security ticket parameter which is returned from login process.

- Login

Parameters

- Username (string): Admin account user name which is defined in configuration.
- Password (string): Admin account password.

Results

- Ticket (string): Security ticket for authentication.

Example

```
SecurityServiceClient securityService = new SecurityServiceClient();
string ticket = securityService.Login( "admin", "admin" );
```

- Logout

Parameters

- Ticket (string): Security ticket which you want to logout.

Example

```
securityService.Logout( ticket );
```

4.2 Store Service

- ListStores

Parameters

- ListMessage (StoreListRequestMessage): Request message for retrieving stores information which is inside website.

Result

- ReturnMessage (StoreListResponseMessage): Response message which contain stores information

Example

```
StoreListRequestMessage srm = new StoreListRequestMessage();
srm.Ticket = ticket;
srm.SearchFilter = new SearchFilterConstraints();
srm.SearchFilter.StartIndex = 0;
srm.SearchFilter.EndIndex = 10;
srm.SearchFilter.FieldName = "Name";
srm.SearchFilter.SearchType =
SearchFilterConstraints.FilterType.Text;
srm.SearchFilter.Value1 = "";
```

```
StoreServiceClient storeService = new StoreServiceClient();
StoreListResponseMessage listMessage =
storeService.ListStores( srm );
```

4.3 Cache Service

- ClearCache

Parameters

- ClearMessage (RequestMessage): Request to clear cache.

Result

- ReturnMessage (ResponseMessage): Response message which contain the result.

Example

```
CacheServiceClient cacheService = new CacheServiceClient();
RequestMessage rmc = new RequestMessage();
rmc.Ticket = ticket;
ResponseMessage rs = cacheService.ClearCache( rmc );
```

4.4 Category Service

- ListCategories

Parameters

- ListMessage (CategoryListRequestMessage): Request message for retrieving categories information which is inside website.

Result

- ReturnMessage (CategoryListResponseMessage): Response message which contain category information.

Example

```
CategoryListRequestMessage crm = new CategoryListRequestMessage();
crm.CultureID = "1";
crm.RootID = "1";
crm.Ticket = ticket;
crm.SearchFilter = new SearchFilterConstraints();
crm.SearchFilter.StartIndex = 0;
crm.SearchFilter.EndIndex = 9;
crm.SearchFilter.FieldName = "Name";
crm.SearchFilter.SearchType =
SearchFilterConstraints.FilterType.Text;
crm.SearchFilter.Value1 = "";

CategoryServiceClient categoryService = new CategoryServiceClient();
CategoryListResponseMessage listMessage =
categoryService.ListCategories( crm );
if (listMessage.ErrorMessage != String.Empty)
{
    AddStringToDisplay( "Error while listing: " +
listMessage.ErrorMessage );
}
```



```

    }
    else
    {
        AddStringToDisplay( "List Category Success" );
        AddStringToDisplay( "Category count:" +
            listMessage.Categories.Length.ToString() );
    }
}

```

4.5 Department Service

- ListDepartments

Parameters

- ListMessage (DepartmentListRequestMessage): Request message for retrieving department information which is inside website.

Result

- ReturnMessage (DepartmentListResponseMessage): Response message which contain department information.

Example

```

DepartmentListRequestMessage drm = new
DepartmentListRequestMessage();
drm.CultureID = "1";
drm.RootID = "1";
drm.Ticket = ticket;
drm.SearchFilter = new SearchFilterConstraints();
drm.SearchFilter.StartIndex = 0;
drm.SearchFilter.EndIndex = 9;
drm.SearchFilter.FieldName = "Name";
drm.SearchFilter.SearchType =
SearchFilterConstraints.FilterType.Text;
drm.SearchFilter.Value1 = "";

DepartmentServiceClient departmentService = new
DepartmentServiceClient();
DepartmentListResponseMessage listMessage =
departmentService.ListDepartments( drm );
if (listMessage.ErrorMessage != String.Empty)
{
    AddStringToDisplay( "Error while listing: " +
        listMessage.ErrorMessage );
}
else
{
    AddStringToDisplay( "List Department Success" );
    AddStringToDisplay( "Department count:" +
        listMessage.Departments.Length.ToString() );
}

```

4.6 Configuration Service

- ListConfigurations

Parameters

- ListMessage (ConfigurationListRequestMessage): Request message for retrieving configuration which is inside website.

Result

- ReturnMessage (ConfigurationListResponseMessage): Response message which contain configuration information.

Example

```
ConfigurationListRequestMessage clrm = new
ConfigurationListRequestMessage();
clrm.Ticket = ticket;
clrm.CultureID = "1";
clrm.SearchFilter = new SearchFilterConstraints();
clrm.SearchFilter.FieldName = "DisplayName";
clrm.SearchFilter.SearchType =
SearchFilterConstraints.FilterType.Text;
clrm.SearchFilter.Value1 = "Store";

ConfigurationServiceClient configurationClient = new
ConfigurationServiceClient();
ConfigurationListResponseMessage listMessage =
configurationClient.ListConfigurations( clrm );

if (listMessage.ErrorMessage != String.Empty)
{
AddStringToDisplay( "Error while listing: " +
listMessage.ErrorMessage );
}
else
{
AddStringToDisplay( "List Configurations Success" );
AddStringToDisplay( "Configurations count:" +
listMessage.Configurations.Length.ToString() );
}
```

4.7 Culture Service

- ListAllCultures

Parameters

- ListMessage (RequestMessage): Request message for retrieving all culture information.

Result

- ReturnMessage (ConfigurationListResponseMessage): Response message which contain cultures information.

Example

```
RequestMessage rm33 = new RequestMessage();
rm33.Ticket = ticket;
CultureServiceClient cultureService = new CultureServiceClient();
CultureListResponseMessage listMessage =
cultureService.ListAllCultures( rm33 );
if (listMessage.ErrorMessage != String.Empty)
{
    AddStringToDisplay( "Error while listing: " +
listMessage.ErrorMessage );
}
else
{
    AddStringToDisplay( "List Cultures Success" );
    AddStringToDisplay( "Cultures count:" +
listMessage.Cultures.Length.ToString() );
}
```

4.8 Customer Service

- ListCustomers

Parameters

- ListMessage (CustomerListRequestMessage): Request message for retrieving customer data.

Result

- ReturnMessage (CustomerListResponseMessage): Response message which contain customer data.

Example

```
CustomerListRequestMessage crm = new CustomerListRequestMessage();
crm.Ticket = ticket;
crm.SearchFilter = new SearchFilterConstraints();
crm.SearchFilter.StartIndex = 0;
crm.SearchFilter.EndIndex = 10;
crm.SearchFilter.FieldName = "Name";
crm.SearchFilter.SearchType =
SearchFilterConstraints.FilterType.Text;
crm.SearchFilter.Value1 = "";

CustomerServiceClient customerService = new CustomerServiceClient();
CustomerListResponseMessage listMessage =
customerService.ListCustomers( crm );
if (listMessage.ErrorMessage != String.Empty)
{
    AddStringToDisplay( "Error while listing: " +
listMessage.ErrorMessage );
}
else
{
    AddStringToDisplay( "List Customers Success" );
    AddStringToDisplay( "Customer count:" +
listMessage.Customers.Length.ToString() );
}
```

- ListWholesales

Parameters

- ListMessage (CustomerListRequestMessage): Request message for retrieving customer data (Wholesales only).

Result

- ReturnMessage (CustomerListResponseMessage): Response message which contain customer data.

Example

```
CustomerListRequestMessage crm = new CustomerListRequestMessage();
crm.Ticket = ticket;
crm.SearchFilter = new SearchFilterConstraints();
crm.SearchFilter.StartIndex = 0;
crm.SearchFilter.EndIndex = 10;
crm.SearchFilter.FieldName = "Name";
crm.SearchFilter.SearchType =
SearchFilterConstraints.FilterType.Text;
crm.SearchFilter.Value1 = "";

CustomerServiceClient customerService = new CustomerServiceClient();
CustomerListResponseMessage listMessage2 =
customerService.ListWholesales( crm );
if (listMessage2.ErrorMessage != String.Empty)
{
    AddStringToDisplay( "Error while listing: " +
listMessage2.ErrorMessage );
}
else
{
    AddStringToDisplay( "List Wholesales Success" );
    AddStringToDisplay( "Wholesale count:" +
listMessage2.Customers.Length.ToString() );
}
```

4.9 Info Service

- GetInfo

Parameters

- ListMessage (RequestMessage): Request message for retrieving general information of website store (Version, Default culture, etc.).

Result

- ReturnMessage (InfoMessage): Response message which contain general information.

Example

```
RequestMessage rm = new RequestMessage();
rm.Ticket = ticket;
InfoServiceClient infoService = new InfoServiceClient();
InfoMessage infoMessage = infoService.GetInfo( rm );
```

```
AddStringToDisplay( infoMessage.Information.CurrentVersion );
AddStringToDisplay( infoMessage.Information.CurrentVersionNumber );
AddStringToDisplay( infoMessage.Information.FullDetailVersion );
AddStringToDisplay( infoMessage.Information.DefaultUrlCultureName );
AddStringToDisplay( infoMessage.Information.AdminDefaultCultureName
);
```

4.10 Order Service

- ListOrders

Parameters

- ListMessage (OrderListRequestMessage): Request message for retrieving order information.

Result

- ReturnMessage (OrderListResponseMessage): Response message which contain order information.

Example

```
OrderListRequestMessage rm2 = new OrderListRequestMessage();
rm2.Ticket = ticket;
rm2.StoreID = "1";
```

```
rm2.SearchFilter = new SearchFilterConstraints();
rm2.SearchFilter.StartIndex = 0;
rm2.SearchFilter.EndIndex = 9;
rm2.SearchFilter.FieldName = "StoreID";
rm2.SearchFilter.SearchType =
SearchFilterConstraints.FilterType.Text;
rm2.SearchFilter.Value1 = order.StoreID;
OrderServiceClient orderService = new OrderServiceClient();
if ((listMessage.ErrorMessage != null) &&
(listMessage.ErrorMessage != String.Empty))
{
    AddStringToDisplay( "Error while listing: " +
listMessage.ErrorMessage );
    return;
}
else
{
    AddStringToDisplay( "List Order Success" );
}
```

- AddOrders

Parameters

- AddMessage (OrderSaveRequestMessage): Request message for adding order information into website store.

Result

- ReturnMessage (ResponseMessage): Response message for adding orders process.

Example

```

OrderSaveRequestMessage rm = new OrderSaveRequestMessage();
rm.Ticket = ticket;

rm.Orders = new OrderConstraints[1];
OrderConstraints order = new OrderConstraints();
order.UserName = "user1";
order.Email = "user1@127.0.0.1";
order.ShippingMethod = "UPS";
order.PaymentMethod = "Bank Transfer";
order.IPAddress = "127.0.0.1";
order.Subtotal = 100;
order.ShippingCost = 4.87M;
order.BaseCurrencyCode = "USD";
order.UserCurrencyCode = "USD";
order.TrackingMethod = "UPS";
order.AvsAddrStatus = "Pass";
order.AvsZipStatus = "Pass";
order.CvvStatus = "Pass";
order.QBTxnID = "";
order.IsQBExported = false;
order.StoreID = "1";

AddressConstraints address1 = new AddressConstraints();
address1.Address1 = "123";
address1.FirstName = "User";
address1.LastName = "One";
address1.State = "LA";
address1.City = "CA";
address1.Country = "US";
order.Billing = address1;

ShippingAddressConstraints address2 = new
ShippingAddressConstraints();
address2.Address1 = "123";
address2.FirstName = "User";
address2.LastName = "One";
address2.State = "LA";
address2.City = "CA";
address2.Country = "US";
address2.IsResidential = true;
order.Shipping = address2;

rm.Orders[0] = order;

OrderServiceClient orderService = new OrderServiceClient();
ResponseMessage addMessage = orderService.AddOrders( rm );

if ((addMessage.ErrorMessage != null) && (addMessage.ErrorMessage !=
String.Empty))
{
    AddStringToDisplay( "Error while adding" );
    return;
}

```

```
else
{
    AddStringToDisplay( "Add Order Success" );
}
```

- **UpdateOrders**

Parameters

- **UpdateMessage (OrderSaveRequestMessage):** Request message for updating order information.

Result

- **ReturnMessage (ResponseMessage):** Response message for updating orders process.

Example

```
OrderConstraints orderUpdate = listMessage.Orders[0];
orderUpdate.Email = "xxx@xxx.com";
OrderSaveRequestMessage rm3 = new OrderSaveRequestMessage();
rm3.Ticket = ticket;
rm3.Orders = new OrderConstraints[1];
rm3.Orders[0] = orderUpdate;
ResponseMessage updateMessage = orderService.UpdateOrders( rm3 );
OrderListResponseMessage listMessage2 =
orderService.ListOrders( rm2 );
if ((listMessage2.ErrorMessage != null) &&
(listMessage2.ErrorMessage != String.Empty))
{
    AddStringToDisplay( "Error while listing (2): " +
listMessage2.ErrorMessage );
    return;
}
else
{
    if (listMessage2.Orders[0].Email != "xxx@xxx.com")
    {
        AddStringToDisplay( "Error updating" );
        return;
    }
    else
    {
        AddStringToDisplay( "Update Order Success" );
    }
}
```

- **DeleteOrders**

Parameters

- **DeleteMessage (OrderDeleteRequestMessage):** Request message for delete orders.

Result

- **ReturnMessage (ResponseMessage):** Response message for delete orders process.

Example

```
OrderDeleteRequestMessage rm4 = new OrderDeleteRequestMessage();
rm4.Ticket = ticket;
rm4.OrderID = new string[1];
rm4.OrderID[0] = listMessage.Orders[0].OrderID;
OrderServiceClient orderService = new OrderServiceClient();

ResponseMessage deleteMessage = orderService.DeleteOrders( rm4 );

if (addMessage.ErrorMessage != String.Empty)
{
    AddStringToDisplay( "Error while deleting: " +
addMessage.ErrorMessage );
}
else
{
    AddStringToDisplay( "Delete Order Success" );
}
```

4.11 Product Service

- ListProducts

Parameters

- ListMessage (ProductListRequestMessage): Request message for retrieving product information.

Result

- ReturnMessage (ProductListResponseMessage): Response message which contain product information.

Example

```
ProductListRequestMessage rm2 = new ProductListRequestMessage();
rm2.Ticket = ticket;
rm2.CategoryID = "2";
rm2.CultureID = "1";
rm2.StoreID = "1";

rm2.SearchFilter = new SearchFilterConstraints();
rm2.SearchFilter.StartIndex = 0;
rm2.SearchFilter.EndIndex = 9;
rm2.SearchFilter.FieldName = "SKU";
rm2.SearchFilter.SearchType =
SearchFilterConstraints.FilterType.Text;
rm2.SearchFilter.Value1 = product.SKU;
ProductServiceClient productService = new ProductServiceClient();

ProductListResponseMessage listMessage =
productService.ListProducts( rm2 );
if ((listMessage.ErrorMessage != null) &&
(listMessage.ErrorMessage != String.Empty))
{
    AddStringToDisplay( "Error while listing: " +
listMessage.ErrorMessage );
    return;
}
```



```

    }
    else
    {
        AddStringToDisplay( "List Product Success" );
    }
}

```

- **AddProducts**

Parameters

- **AddMessage (ProductSaveRequestMessage):** Request message for adding product data into website store.

Result

- **ReturnMessage (ResponseMessage):** Response message for adding product process.

Example

```

ProductSaveRequestMessage rm = new ProductSaveRequestMessage();
rm.Ticket = ticket;
rm.CultureID = "1";
rm.StoreID = "1";

rm.Products = new ProductConstraints[1];
ProductConstraints product = new ProductConstraints();
product.CategoryIDs = new string[] { "2" };
product.DepartmentIDs = new string[] { "" };
product.Name = RandomString( 10, false );
product.SKU = RandomString( 5, false );
product.ProductPrices = new ProductPriceConstraints();
product.ProductPrices.Price = 10;
product.UseInventory = true;
product.ProductStocks = new ProductStockConstraints();
product.ProductStocks.Stock = 10;
rm.Products[0] = product;

ProductServiceClient productService = new ProductServiceClient();
ResponseMessage addMessage = productService.AddProducts( rm );
if ((addMessage.ErrorMessage != null) && (addMessage.ErrorMessage !=
String.Empty))
{
    AddStringToDisplay( "Error while adding" );
    return;
}
else
{
    AddStringToDisplay( "Add Product Success" );
}
}

```

- **UpdateProducts**

Parameters

- **UpdateMessage (ProductSaveRequestMessage):** Request message for updating product data.

Result

- **ReturnMessage (ResponseMessage):** Response message for updating product process.

Example

```
ProductConstraints productUpdate = listMessage.Products[0];
productUpdate.ShortDescription = "Update Save";
ProductSaveRequestMessage rm3 = new ProductSaveRequestMessage();
rm3.Ticket = ticket;
rm3.CultureID = "1";
rm3.StoreID = "1";
rm3.Products = new ProductConstraints[1];
rm3.Products[0] = productUpdate;

ProductServiceClient productService = new ProductServiceClient();
ResponseMessage updateMessage =
productService.UpdateProducts( rm3 );
ProductListResponseMessage listMessage2 =
productService.ListProducts( rm2 );
if ((listMessage2.ErrorMessage != null) &&
(listMessage2.ErrorMessage != String.Empty))
{
    AddStringToDisplay( "Error while listing (2): " +
listMessage2.ErrorMessage );
    return;
}
else
{
    if (listMessage2.Products[0].ShortDescription != "Update Save")
    {
        AddStringToDisplay( "Error updating" );
        return;
    }
    else
    {
        AddStringToDisplay( "Update Product Success" );
    }
}
```

- **DeleteProducts**

Parameters

- **DeleteMessage (ProductDeleteRequestMessage):** Request message for delete products.

Result

- **ReturnMessage (ResponseMessage):** Response message for delete product process.

Example

```
ProductDeleteRequestMessage rm4 = new ProductDeleteRequestMessage();
rm4.Ticket = ticket;
rm4.ProductID = new string[1];
rm4.ProductID[0] = listMessage.Products[0].ProductID;
ProductServiceClient productService = new ProductServiceClient();
```

```

ResponseMessage deleteMessage =
productService.DeleteProducts( rm4 );

if (addMessage.ErrorMessage != String.Empty)
{
    AddStringToDisplay( "Error while deleting: " +
addMessage.ErrorMessage );
}
else
{
    AddStringToDisplay( "Delete Product Success" );
}

```