Network›

International Payment Solutions

PAYMENT GATEWAY INTEGRATION REVERSAL API SAMPLE CODE C#

USER MANUAL

VERSION: 1.0.0

20-JUNE-2017

Contents

Introduction 3
Prerequisites 3
Installation 3
Settings & Executing the file 4
Adding .ASMX files as reference 7
Reversal API Web-Service Details 8

Introduction

Reversal API Web Service is used perform transactions related queries, much easily and quickly as compared to doing so through the 'Merchant Portal'. You can send the transaction details to the payment gateway using the Web Service API containing certain parameters as defined in the individual transaction message structures and you can perfom different types of reversal processes which are given below.

Prerequisites

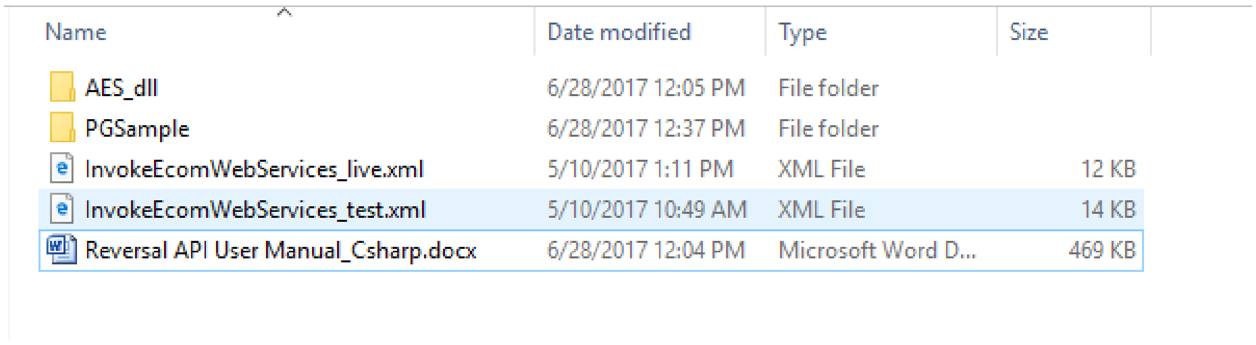
Requires .Net framework: 4.0 or above.

Visual Studio IDE

AES.DLL version 1.0.0.0

Installation

Extract the zip folder and copy the folder to your local folder. Once the copying is done,if needed you can remove the documentation file`s like (pdf`s, doc, .txt files).

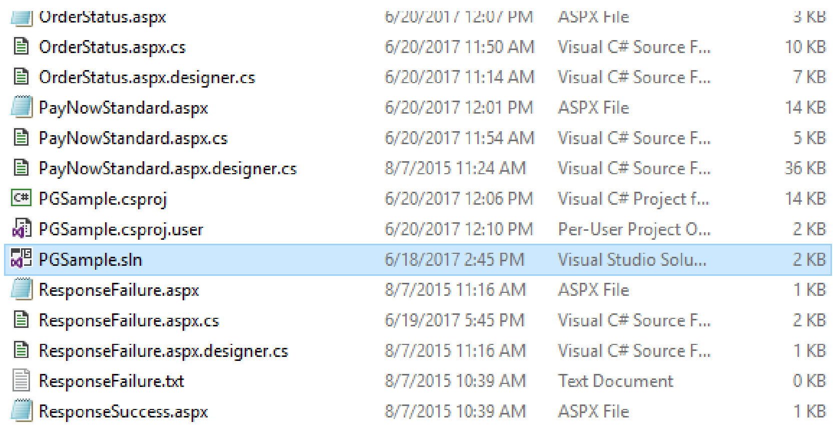


Name	Date modified	Type	Size
AES_dll	6/28/2017 12:05 PM	File folder	
PGSample	6/28/2017 12:37 PM	File folder	
InvokeEcomWebServices_live.xml	5/10/2017 1:11 PM	XML File	12 KB
InvokeEcomWebServices_test.xml	5/10/2017 10:49 AM	XML File	14 KB
Reversal API User Manual_Csharp.docx	6/28/2017 12:04 PM	Microsoft Word D...	469 KB

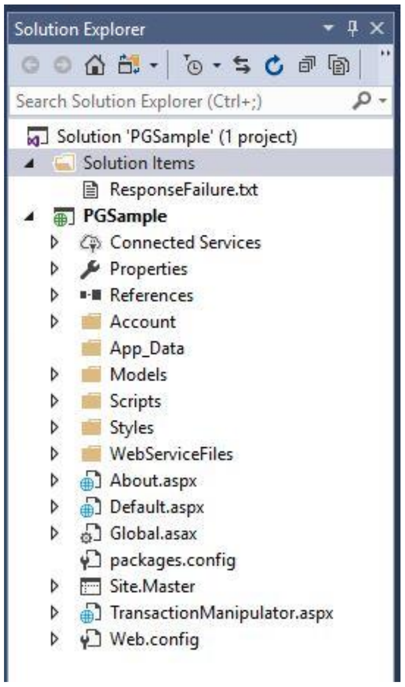
Settings & Executing the file

Once the installation is done please open the solution using PGSample.sln file in Visual studio IDE.

Values like **merchantKey merchantId** with KEY & MID which you have and be inserted through the UI provided with TransactionManipulator.aspx . Along with this values please assign the **NI Reference Number** and amount to refund.



OrderStatus.aspx	6/20/2017 12:07 PM	ASPX File	3 KB
OrderStatus.aspx.cs	6/20/2017 11:50 AM	Visual C# Source F...	10 KB
OrderStatus.aspx.designer.cs	6/20/2017 11:14 AM	Visual C# Source F...	7 KB
PayNowStandard.aspx	6/20/2017 12:01 PM	ASPX File	14 KB
PayNowStandard.aspx.cs	6/20/2017 11:54 AM	Visual C# Source F...	5 KB
PayNowStandard.aspx.designer.cs	8/7/2015 11:24 AM	Visual C# Source F...	36 KB
PGSample.csproj	6/20/2017 12:06 PM	Visual C# Project f...	14 KB
PGSample.csproj.user	6/20/2017 12:10 PM	Per-User Project O...	2 KB
PGSample.sln	6/18/2017 2:45 PM	Visual Studio Solu...	2 KB
ResponseFailure.aspx	8/7/2015 11:16 AM	ASPX File	1 KB
ResponseFailure.aspx.cs	6/19/2017 5:45 PM	Visual C# Source F...	2 KB
ResponseFailure.aspx.designer.cs	8/7/2015 11:16 AM	Visual C# Source F...	1 KB
ResponseFailure.txt	8/7/2015 10:39 AM	Text Document	0 KB
ResponseSuccess.aspx	8/7/2015 10:39 AM	ASPX File	1 KB

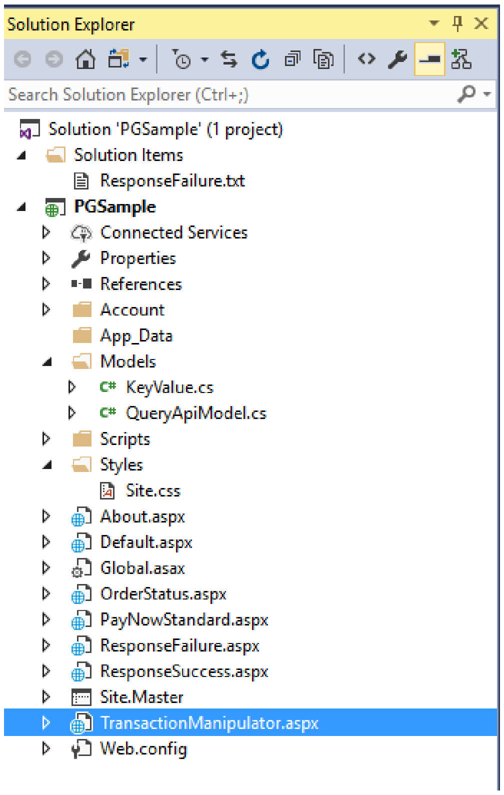


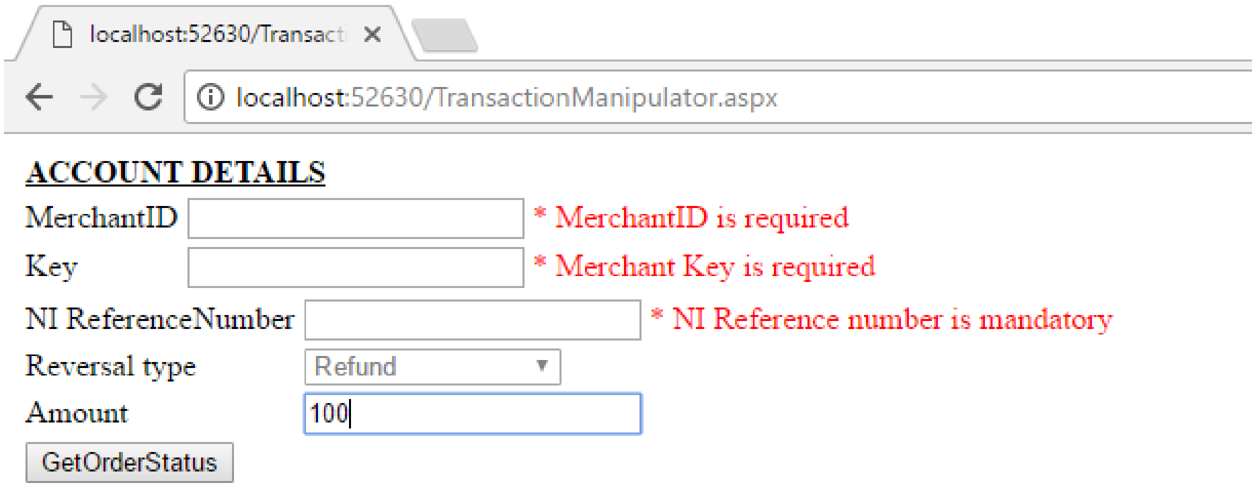
```
Solution 'PGSample' (1 project)
◢ Solution Items
    ResponseFailure.txt
◢ PGSample
  ▷ Connected Services
  ▷ Properties
  ▷ References
  ▷ Account
    App_Data
  ▷ Models
  ▷ Scripts
  ▷ Styles
  ▷ WebServiceFiles
  ▷ About.aspx
  ▷ Default.aspx
  ▷ Global.asax
    packages.config
  ▷ Site.Master
  ▷ TransactionManipulator.aspx
  ▷ Web.config
```

Data insertion:

You can use the below screen to enter values where Order number is mandatory.

To get the below form, please set **TransactionManipulator.aspx** page as start page and run the solution



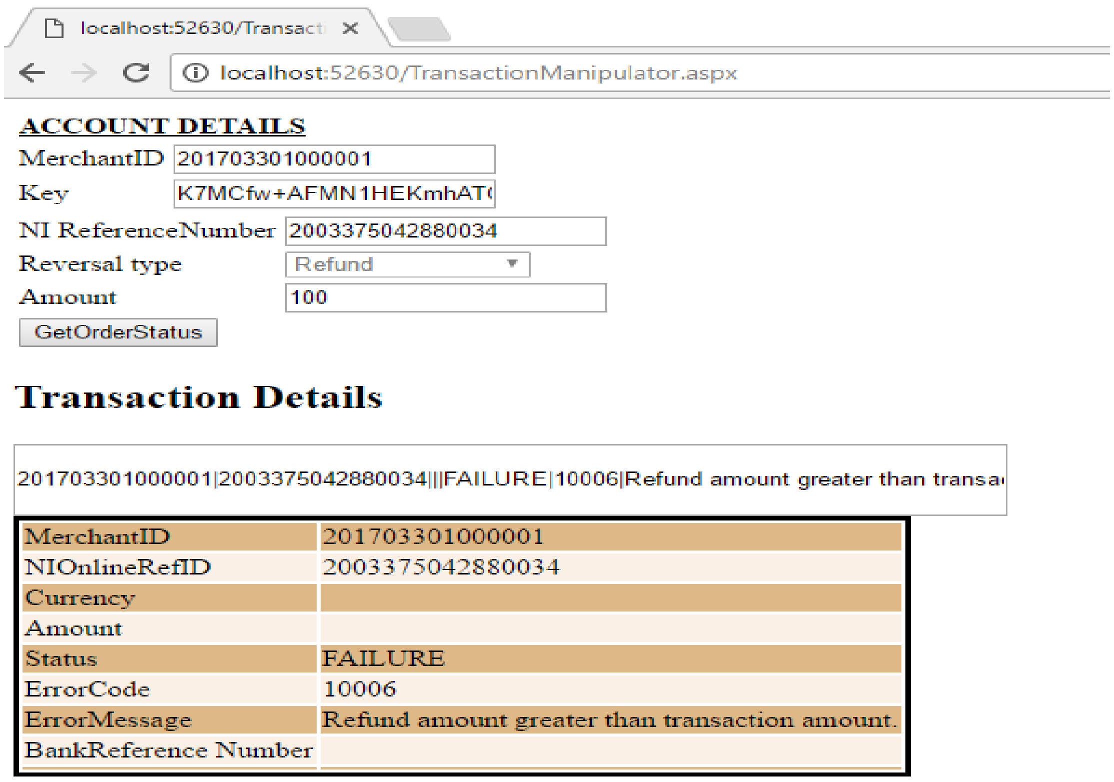


ACCOUNT DETAILS

MerchantID		* MerchantID is required
Key		* Merchant Key is required
NI ReferenceNumber		* NI Reference number is mandatory
Reversal type	Refund	
Amount	100	

URL ex: http://localhost:port/ TransactionManipulator.aspx

You will be getting the following results: This UI is provided for the ease of testing and displaying the decoded response in a better understandable format.



localhost:52630/Transact

localhost:52630/TransactionManipulator.aspx

ACCOUNT DETAILS

MerchantID	201703301000001
Key	K7MCfw+AFMN1HEKmhAT(
NI ReferenceNumber	2003375042880034
Reversal type	Refund
Amount	100

GetOrderStatus

Transaction Details

```
201703301000001|2003375042880034|||FAILURE|10006|Refund amount greater than transa
```

MerchantID	201703301000001
NIOnlineRefID	2003375042880034
Currency	
Amount	
Status	FAILURE
ErrorCode	10006
ErrorMessage	Refund amount greater than transaction amount.
BankReference Number	

Adding .ASMX files as reference

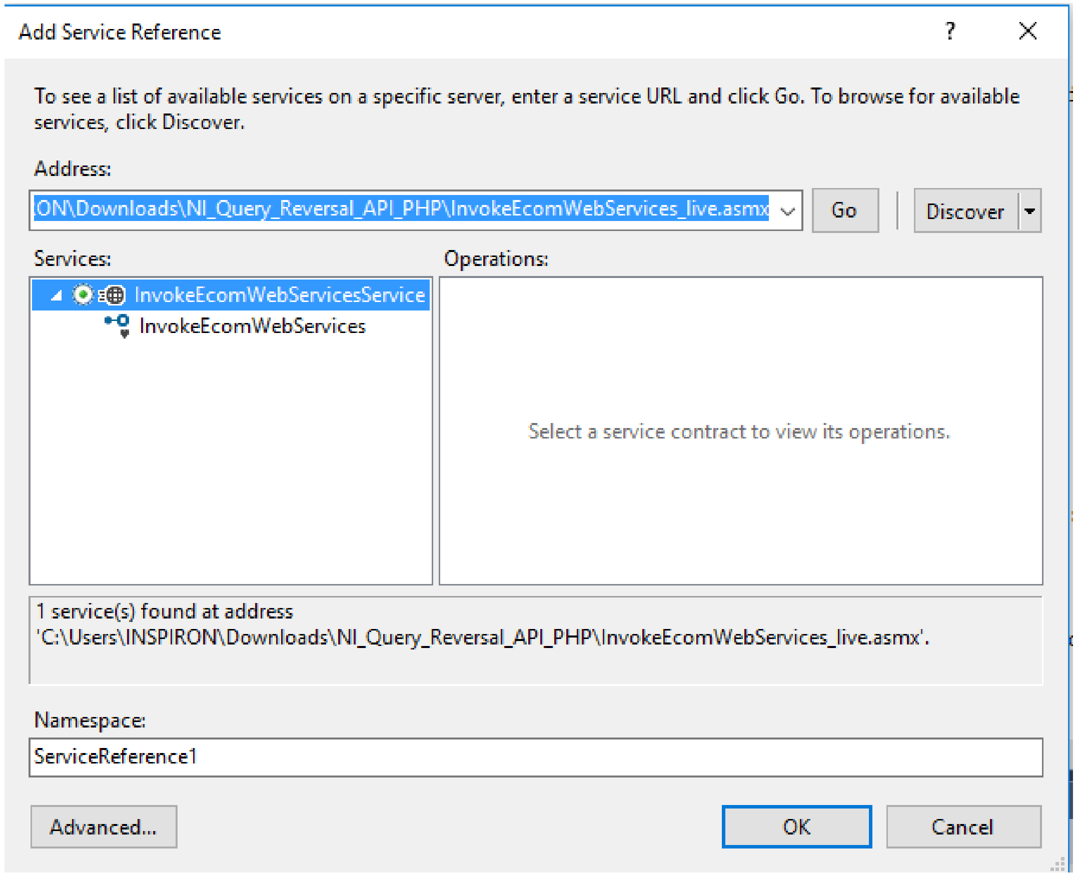
First change the extension of the required XML file to .asmx and add service reference.Give meaningful name space name and click ok which will create a proxy class and you can use it as below.

WSDL files
InvokeEcomWebServices_live.xml
InvokeEcomWebServices_test.xml

One file is for live environment and the other is for test environment which will handle the SOAP calls.

```
ServiceReferenceInvokeEcomWebServices.InvokeEcomWebServicesClient clientObj= new
ServiceReferenceInvokeEcomWebServices.InvokeEcomWebServicesClient();

ServiceReferenceInvokeEcomWebServices_live.InvokeEcomWebServicesClient clientObj= new
ServiceReferenceInvokeEcomWebServices_live.InvokeEcomWebServicesClient();
```



To see a list of available services on a specific server, enter a service URL and click Go. To browse for available services, click Discover.

Address: ON\Downloads\NI_Query_Reversal_API_PHP\InvokeEcomWebServices_live.asmx

Services:
- InvokeEcomWebServicesService
  - InvokeEcomWebServices

Operations:
Select a service contract to view its operations.

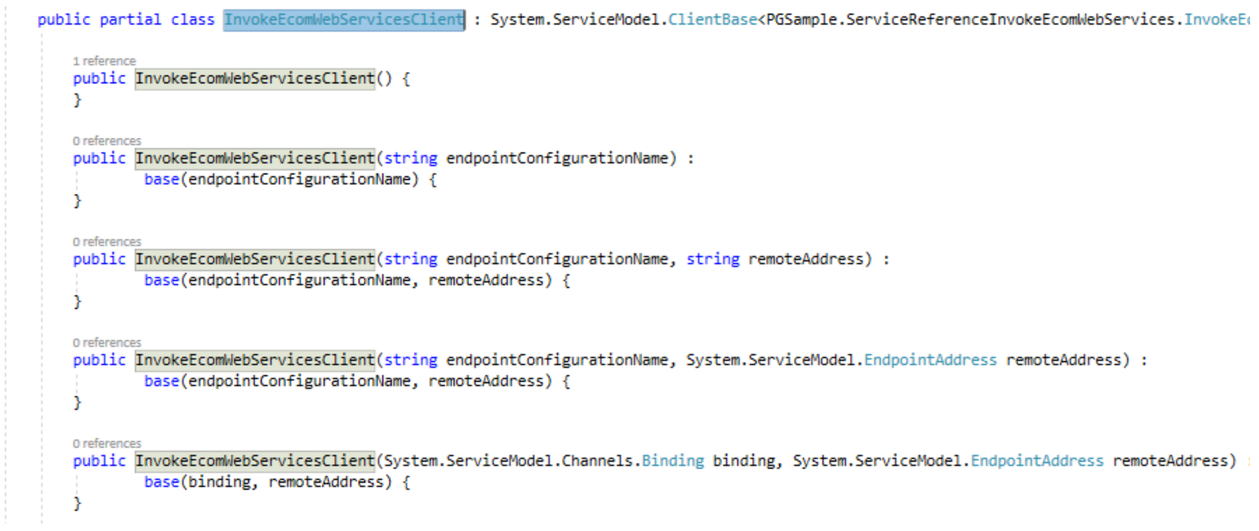
1 service(s) found at address 'C:\Users\INSPIRON\Downloads\NI_Query_Reversal_API_PHP\InvokeEcomWebServices_live.asmx'.

Namespace: ServiceReference1

Note:- This is already added to the solution only required when not added.

Reversal API Web-Service Details

To get web service methods we have generated a proxy class given below which contains methods to pull result from the web service.



```
public partial class InvokeEcomWebServicesClient : System.ServiceModel.ClientBase<PGSample.ServiceReferenceInvokeEcomWebServices.InvokeE

    1 reference
    public InvokeEcomWebServicesClient() {
    }

    0 references
    public InvokeEcomWebServicesClient(string endpointConfigurationName) :
            base(endpointConfigurationName) {
    }

    0 references
    public InvokeEcomWebServicesClient(string endpointConfigurationName, string remoteAddress) :
            base(endpointConfigurationName, remoteAddress) {
    }

    0 references
    public InvokeEcomWebServicesClient(string endpointConfigurationName, System.ServiceModel.EndpointAddress remoteAddress) :
            base(endpointConfigurationName, remoteAddress) {
    }

    0 references
    public InvokeEcomWebServicesClient(System.ServiceModel.Channels.Binding binding, System.ServiceModel.EndpointAddress remoteAddress) :
            base(binding, remoteAddress) {
    }
```

You can pass the encrypted referenceID and separately encrypted amount to web service using below code where client is an object of proxy class.

```
strMessage = aesEncrypt.Encrypt(txtKey.Text, strMessage);
string amount= aesEncrypt.Encrypt(txtKey.Text, txtAmount.Text);
```

web service call for Invoking Refund

string msg = `clientObj.InvokeReversalWS(txtMerchantID.Text, strMessage, amount);`

web service call for Invoking partial capture

string msg = `clientObj. InvokePartialCaptureWS(txtMerchantID.Text, strMessage, amount);`

web service call for Invoking capture

string msg = `clientObj. InvokeCaptureWS (txtMerchantID.Text, strMessage);`

web service call for Invoking void

string msg = `clientObj. InvokeVoidWS (txtMerchantID.Text, strMessage);`

web service call for invoking Full Auth reversal

string msg = `clientObj. InvokeFullAuthReversalWS (txtMerchantID.Text, strMessage);`