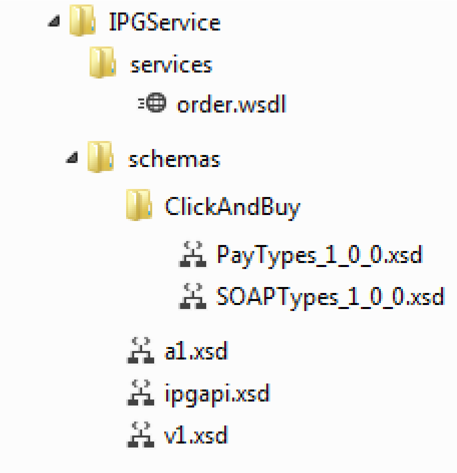
.NET API Integration

1. Install client and server certificates (The process is described in IPG_IntegrationGuide_API section 16.2 ASP)

2. Add Web Reference order.wsdl from https://test.ipg-online.com/ipgapi/services/order.wsdl

   If you are unable to add the Web Reference from the *URL* you will need to have all schema files on your PC:
   - order.wsd
   - ipgapi.xsd
   - a1.xsd
   - v1.xsd
   - PayTypes_1_0_0.xsd
   - SOAPTypes_1_0_0.xsd

   Schemas hierarchy:

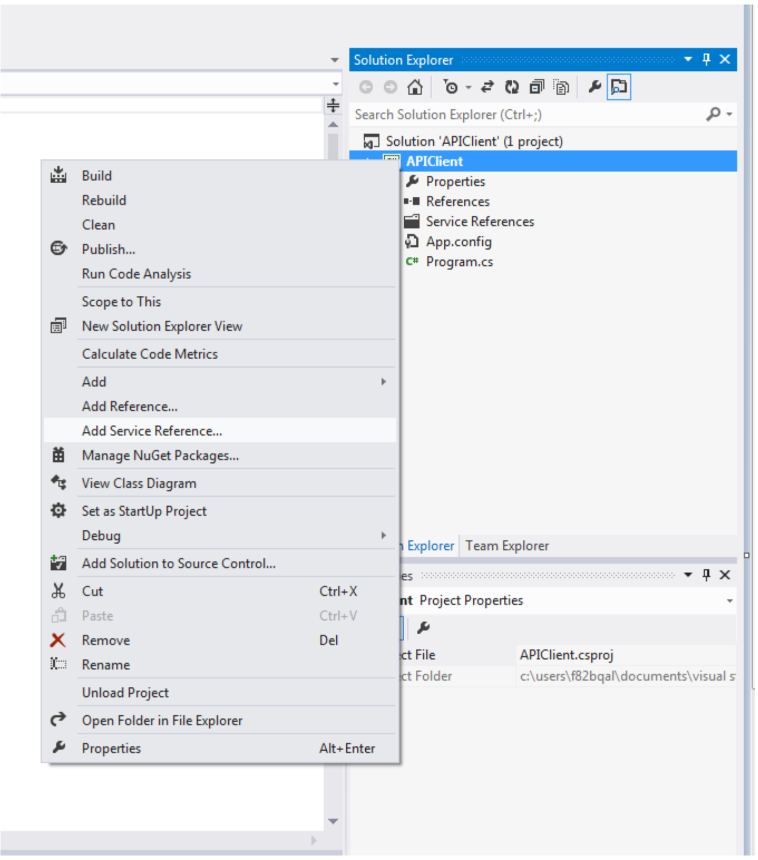


```
IPGService
    services
        order.wsdl
    schemas
        ClickAndBuy
            PayTypes_1_0_0.xsd
            SOAPTypes_1_0_0.xsd
        a1.xsd
        ipgapi.xsd
        v1.xsd
```

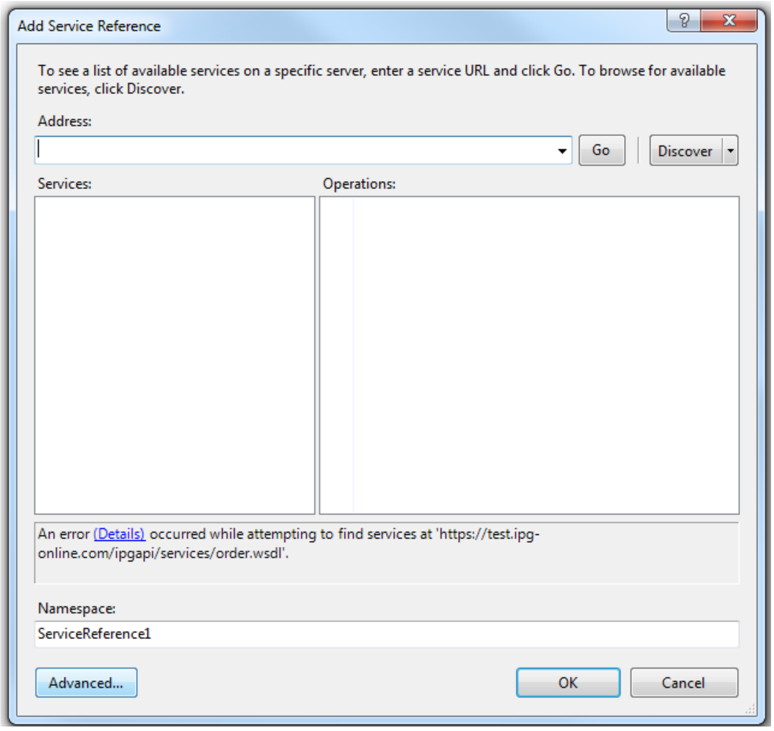
   And use the path to the order.wsdl file you downloaded instead.

Adding the Web Reference

1. Right Click on your project and click on *Add Service Reference*



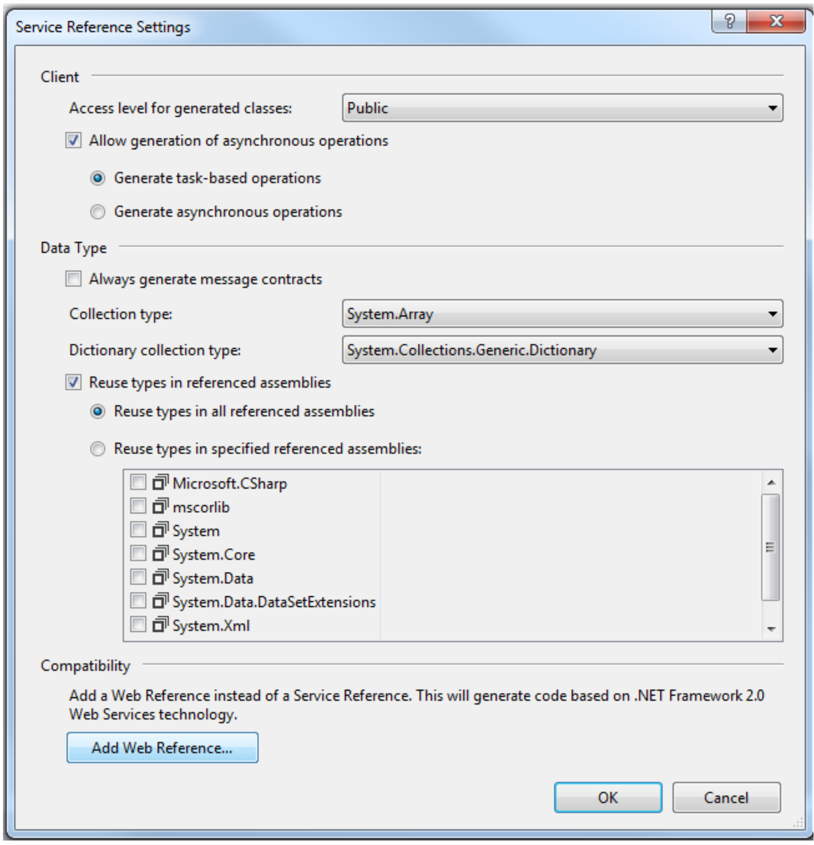
2. Click on *Advanced…* button



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.

An error (Details) occurred while attempting to find services at 'https://test.ipg-online.com/ipgapi/services/order.wsdl'.

3. Click on *Add Web Reference…*



Service Reference Settings

Client

Access level for generated classes: Public

☑ Allow generation of asynchronous operations

● Generate task-based operations

○ Generate asynchronous operations

Data Type

□ Always generate message contracts

Collection type: System.Array

Dictionary collection type: System.Collections.Generic.Dictionary

☑ Reuse types in referenced assemblies

● Reuse types in all referenced assemblies

○ Reuse types in specified referenced assemblies:

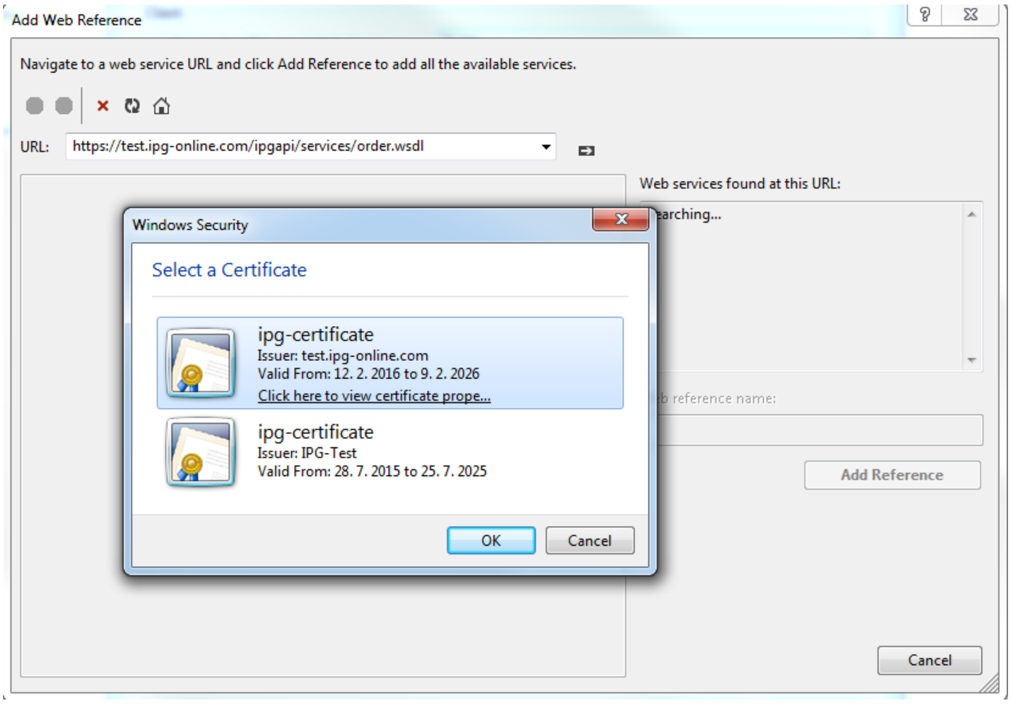
□ Microsoft.CSharp
□ mscorlib
□ System
□ System.Core
□ System.Data
□ System.Data.DataSetExtensions
□ System.Xml

Compatibility

Add a Web Reference instead of a Service Reference. This will generate code based on .NET Framework 2.0 Web Services technology.

\

4. Give https://test.ipg-online.com/ipgapi/services/order.wsdl as the *URL* and press enter. You will be asked to select a certificate (the one you have installed). Select it and press *OK*.



Add Web Reference

Navigate to a web service URL and click Add Reference to add all the available services.

URL: https://test.ipg-online.com/ipgapi/services/order.wsdl

Web services found at this URL:

earching...

b reference name:

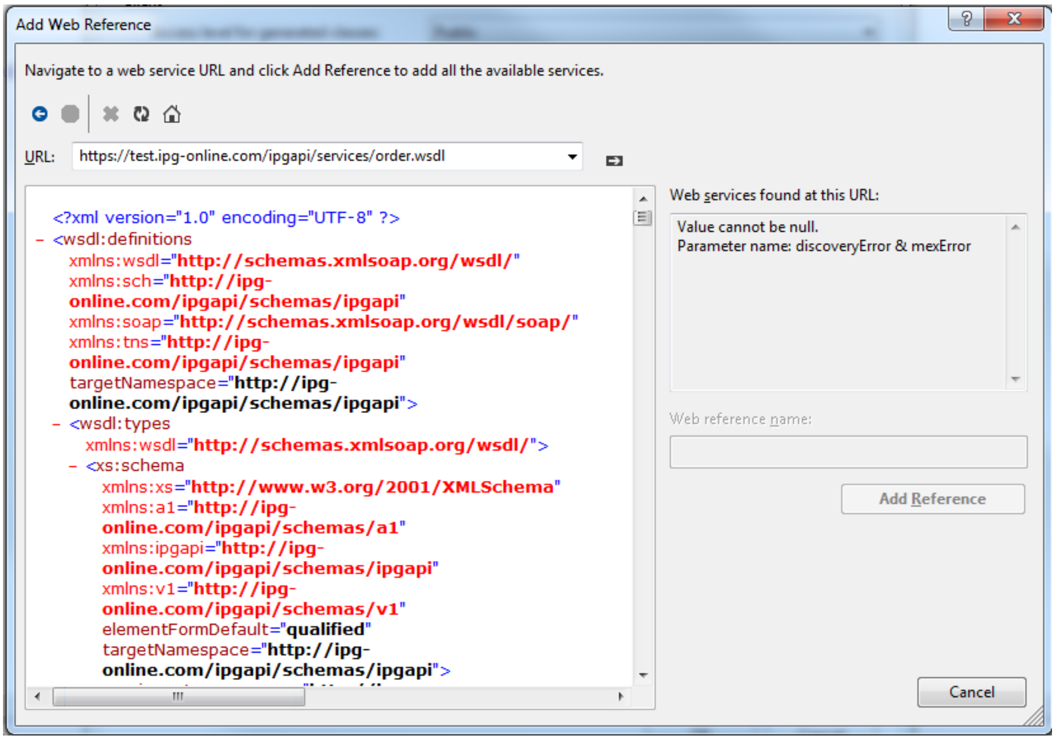
Windows Security

Select a Certificate

ipg-certificate
Issuer: test.ipg-online.com
Valid From: 12. 2. 2016 to 9. 2. 2026
Click here to view certificate prope...

ipg-certificate
Issuer: IPG-Test
Valid From: 28. 7. 2015 to 25. 7. 2025

5. If you are unable to add it as Web Reference with the *URL* https://test.ipg-online.com/ipgapi/services/order.wsdl, you will have to add it from your PC.



Add Web Reference

Navigate to a web service URL and click Add Reference to add all the available services.

URL: https://test.ipg-online.com/ipgapi/services/order.wsdl

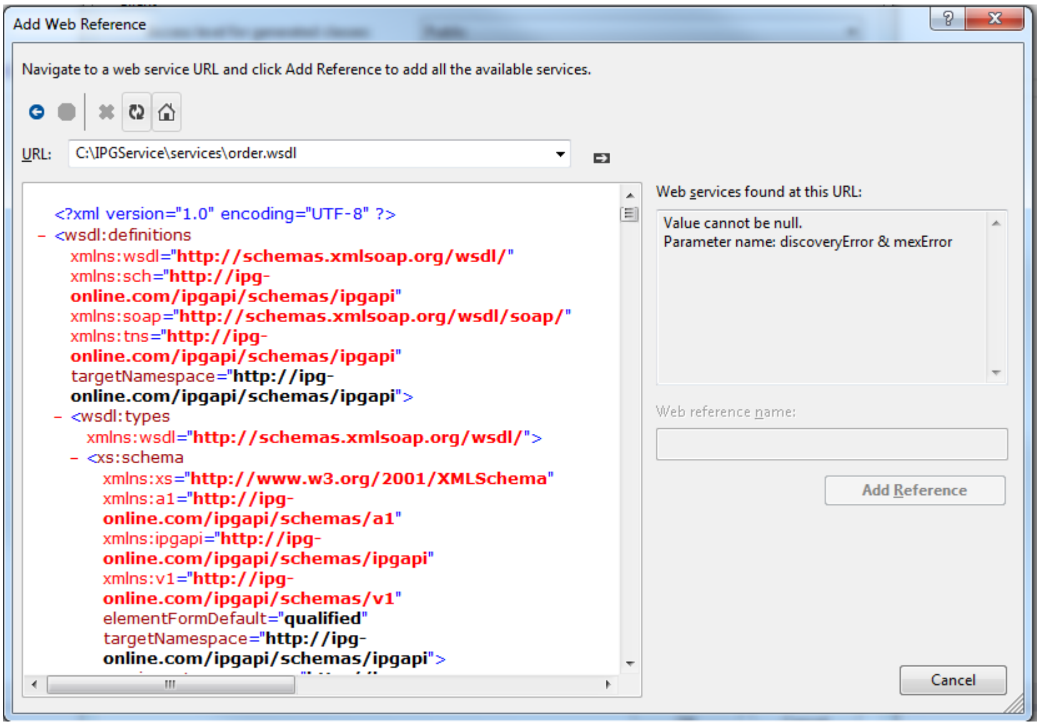
```
  <?xml version="1.0" encoding="UTF-8" ?>
- <wsdl:definitions
    xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
    xmlns:sch="http://ipg-
    online.com/ipgapi/schemas/ipgapi"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns:tns="http://ipg-
    online.com/ipgapi/schemas/ipgapi"
    targetNamespace="http://ipg-
    online.com/ipgapi/schemas/ipgapi">
  - <wsdl:types
      xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
    - <xs:schema
        xmlns:xs="http://www.w3.org/2001/XMLSchema"
        xmlns:a1="http://ipg-
        online.com/ipgapi/schemas/a1"
        xmlns:ipgapi="http://ipg-
        online.com/ipgapi/schemas/ipgapi"
        xmlns:v1="http://ipg-
        online.com/ipgapi/schemas/v1"
        elementFormDefault="qualified"
        targetNamespace="http://ipg-
        online.com/ipgapi/schemas/ipgapi">
```

Web services found at this URL:

Value cannot be null.
Parameter name: discoveryError & mexError

Web reference name:

6. (In case you weren't able to add the Web Reference with the *URL*)
Give as *URL* the path to the order.wsdl that you have on your PC. If you are adding the Web Reference this way you need to have all the schemas present on your computer.



Add Web Reference

Navigate to a web service URL and click Add Reference to add all the available services.

URL: C:\IPGService\services\order.wsdl

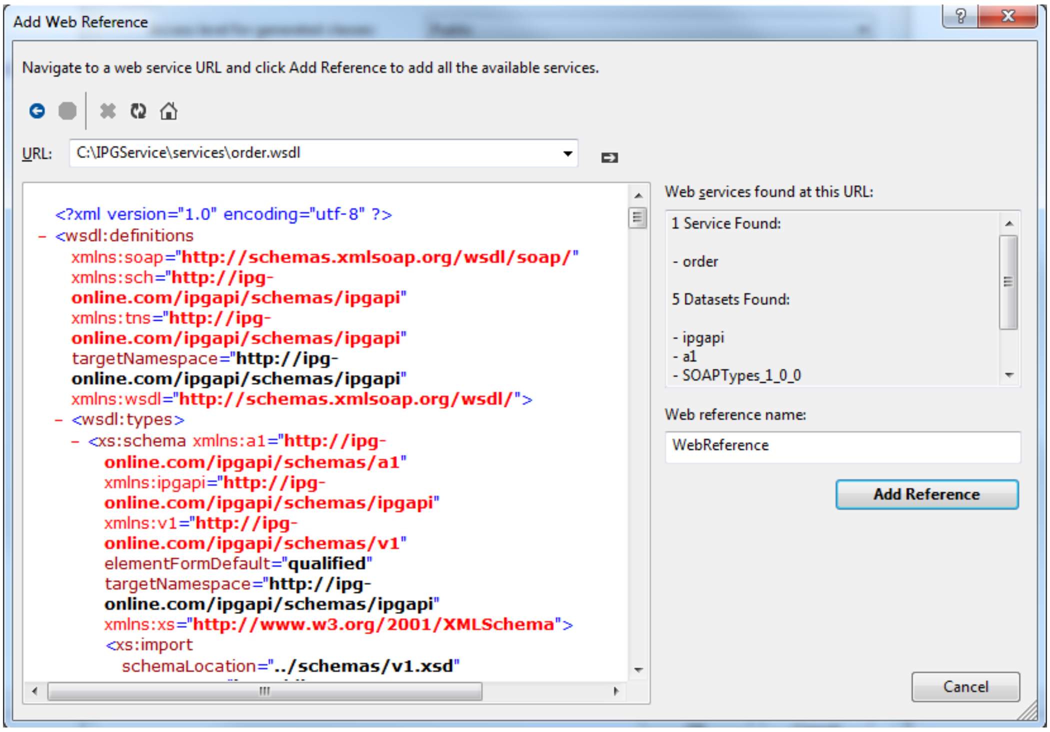
```
  <?xml version="1.0" encoding="UTF-8" ?>
- <wsdl:definitions
    xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
    xmlns:sch="http://ipg-
    online.com/ipgapi/schemas/ipgapi"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns:tns="http://ipg-
    online.com/ipgapi/schemas/ipgapi"
    targetNamespace="http://ipg-
    online.com/ipgapi/schemas/ipgapi">
  - <wsdl:types
      xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
    - <xs:schema
        xmlns:xs="http://www.w3.org/2001/XMLSchema"
        xmlns:a1="http://ipg-
        online.com/ipgapi/schemas/a1"
        xmlns:ipgapi="http://ipg-
        online.com/ipgapi/schemas/ipgapi"
        xmlns:v1="http://ipg-
        online.com/ipgapi/schemas/v1"
        elementFormDefault="qualified"
        targetNamespace="http://ipg-
        online.com/ipgapi/schemas/ipgapi">
```

Web services found at this URL:

Value cannot be null.
Parameter name: discoveryError & mexError

Web reference name:

7. The Service was found successfully. Name your Web Reference and click on *Add Reference*



Navigate to a web service URL and click Add Reference to add all the available services.

URL: C:\IPGService\services\order.wsdl

```
  <?xml version="1.0" encoding="utf-8" ?>
- <wsdl:definitions
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
    xmlns:sch="http://ipg-
    online.com/ipgapi/schemas/ipgapi"
    xmlns:tns="http://ipg-
    online.com/ipgapi/schemas/ipgapi"
    targetNamespace="http://ipg-
    online.com/ipgapi/schemas/ipgapi"
    xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  - <wsdl:types>
    - <xs:schema xmlns:a1="http://ipg-
        online.com/ipgapi/schemas/a1"
        xmlns:ipgapi="http://ipg-
        online.com/ipgapi/schemas/ipgapi"
        xmlns:v1="http://ipg-
        online.com/ipgapi/schemas/v1"
        elementFormDefault="qualified"
        targetNamespace="http://ipg-
        online.com/ipgapi/schemas/ipgapi"
        xmlns:xs="http://www.w3.org/2001/XMLSchema">
        <xs:import
          schemaLocation="../schemas/v1.xsd"
```

Web services found at this URL:

1 Service Found:

- order

5 Datasets Found:

- ipgapi
- a1
- SOAPTypes_1_0_0

Web reference name: WebReference

The application will add the schema files into the project.

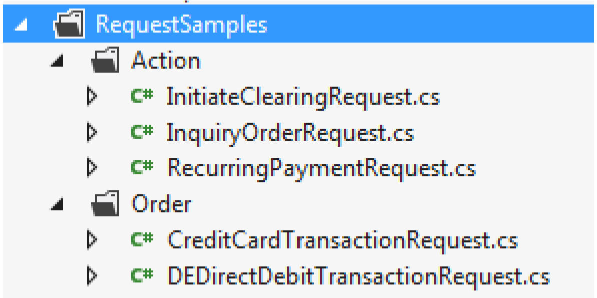
Using the WSIPGAPIClient

WSAPIAPIClient is a console Application that is a sample integration of the IPG API.

A sample code for creating Action request (InquiryOrder)

```
static void Main(string[] args)
    {
      // Disable Expect100Continue, when set to true (default value) error occures
      // The request was aborted: Could not create SSL/TLS secure channel.
      ServicePointManager.Expect100Continue = false;

      //load certificate (the installed one)
      X509Certificate2 certificate = null;
      try
      {
        certificate = new X509Certificate2(@"c:\certificates\WS120995000._.4.p12", "tester02");
      }
      catch (Exception e)
      {
        Console.WriteLine("Problem loading certificate");
        Console.WriteLine("Exception: " + e.Message);
      }

      // Create Request: IPGApiOrderService
      IPGApiOrderService Request = new IPGApiOrderService();
      String RequestResponse = "";
      if (certificate != null)
      {
        // Add certificate to the Request
        Request.ClientCertificates.Add(certificate);

        //Specify the service URL
        Request.Url = @"https://test.ipg-online.com:443/ipgapi/services";

        //Set Network Credentials (certificate user and password) to the Request
        Request.Credentials = new NetworkCredential("WS120995000._.4", "Tester02");

        // If needed set Proxy settings to the Request
        IWebProxy webProxy = new WebProxy("fdskbc1vip.sk.transacty.com", 8080);
        webProxy.Credentials = new NetworkCredential("username", "password");
        Request.Proxy = webProxy;

        // create Request: Action: InquiryOrder
        InquiryOrder oInquiryOrder = new InquiryOrder();
        oInquiryOrder.StoreId = "120995000";
        oInquiryOrder.OrderId = "C-0cc41568-aef6-45bb-8b26-5d1d5733191d";

        ClientLocale oClientLocale = new ClientLocale();
        oClientLocale.Country = "UK";
        oClientLocale.Language = "en";

        APIClient.WebReference.Action oAction = new APIClient.WebReference.Action();
        oAction.Item = oInquiryOrder;
        oAction.ClientLocale = oClientLocale;

        IPGApiActionRequest ActionRequest = new IPGApiActionRequest();
        ActionRequest.Item = oAction;
        try
        {
          IPGApiActionResponse oResponse = Request.IPGApiAction(ActionRequest);
          RequestResponse += "Succesfully:" + oResponse.successfully;
        }
        catch (Exception e)
        {
          RequestResponse = e.Message;
        }finally{
          Console.WriteLine(RequestResponse);
          Console.ReadLine();
        }
      }
    }
```

In the WSIPGClient the request samples are located in RequestSamples folder.



```
◢ RequestSamples
    ◢ Action
        ▷ C# InitiateClearingRequest.cs
        ▷ C# InquiryOrderRequest.cs
        ▷ C# RecurringPaymentRequest.cs
    ◢ Order
        ▷ C# CreditCardTransactionRequest.cs
        ▷ C# DEDirectDebitTransactionRequest.cs
```

E.g. For Action Request `InquiryOrder` there is a corresponding class created `InquiryOrdrRequest` that creates the request.

InquiryOrderRequest Class:

```
class InquiryOrderRequest
    {
        public IPGApiActionRequest InquiryOrderActionRequest{ get; set; }

        public InquiryOrderRequest()
        {
            InquiryOrder oInquiryOrder = new InquiryOrder();
            oInquiryOrder.StoreId = "120995000";
            oInquiryOrder.OrderId = "C-0cc41568-aef6-45bb-8b26-5d1d5733191d";

            ClientLocale oClientLocale = new ClientLocale();
            oClientLocale.Country = "UK";
            oClientLocale.Language = "en";

            WSIPGClient.WebReference.Action oAction;
            oAction = new WSIPGClient.WebReference.Action();
            oAction.Item = oInquiryOrder;
            oAction.ClientLocale = oClientLocale;

            InquiryOrderActionRequest = new IPGApiActionRequest();
            InquiryOrderActionRequest.Item = oAction;
        }
    }
```

Request / Response Logging

In this sample the class *LoggerSoapExtension.cs* is used to log the outgoing request / incoming response to a file. All that is needed to do is to add the following configuration to the *app.config* and define the location where the files will be saved to (`LOG_DIRECTORY`)

```
<system.web>
    <webServices>
      <soapExtensionTypes>
        <add type="WSIPGClient.LoggerSoapExtension, WSIPGClient" priority="1" />
      </soapExtensionTypes>
    </webServices>
  </system.web>

  <appSettings>
    <add key="LOG_DIRECTORY" value="C:\log\" />
  </appSettings>
```