

SYSTINT Project VT2022

Task 4 – Develop a service integration in the cloud

Goal

- To introduce the integration of Salesforce SOAP API with .NET based application
- To demonstrate how cloud-based web services can be consumed in .NET based Windows application
- To demonstrate how Windows applications are authenticated using usernames passwords and security token
- To demonstrate how a reference to web services are added to Windows .NET applications

Learning outcomes

After doing this exercise, students will learn how to integrate Salesforce SOAP APIs with .NET based Windows Applications.

Good Luck!

Workneh

Contents

1.	Integrating salesforce SOAP API using C# .NET	3
1.1	Setting UP Salesforce Developer Account.....	3
1.2	Changing to classic mode	5
1.3	Changing the default language to English	6
1.4	Salesforce and browser options	6
1.5	Enabling developer mode.....	7
1.6	To get support from Salesforce Forum	8
1.7	Generating security token	8
1.8	Get the WSDL SOAP API	9
2.	Creating Integrated Windows Application with Salesforce.....	10
2.1.	Adding Products in Salesforce	10
2.2.	Creating Integrated Windows .NET Application.....	11
2.3.	Adding controls to Form1 designer	14
2.4.	Adding the code that makes the program work	15
2.5.	Testing if the code works	17

1. Integrating salesforce SOAP API using C# .NET

In this exercise, we will learn how to configure Salesforce API to be accessed by clients from outside. We will see how we can consume or use web service APIs built-in salesforce with a windows application. ASP.NET, Mobile, and java based applications can also be used to consume salesforce services.

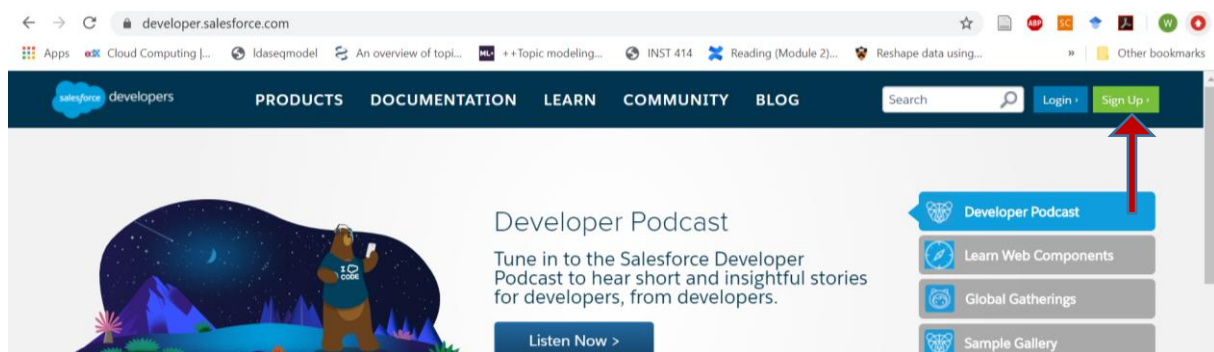
Before interacting with salesforce, one needs to have a Salesforce developer account:

- username and password
- Security Token
- SOAP API (WSDL file)

1.1 Setting UP Salesforce Developer Account

Steps

- Open a browser preferably Firefox (Chrome also works)
- Enter – developer.salesforce.com in the address bar > then press Enter



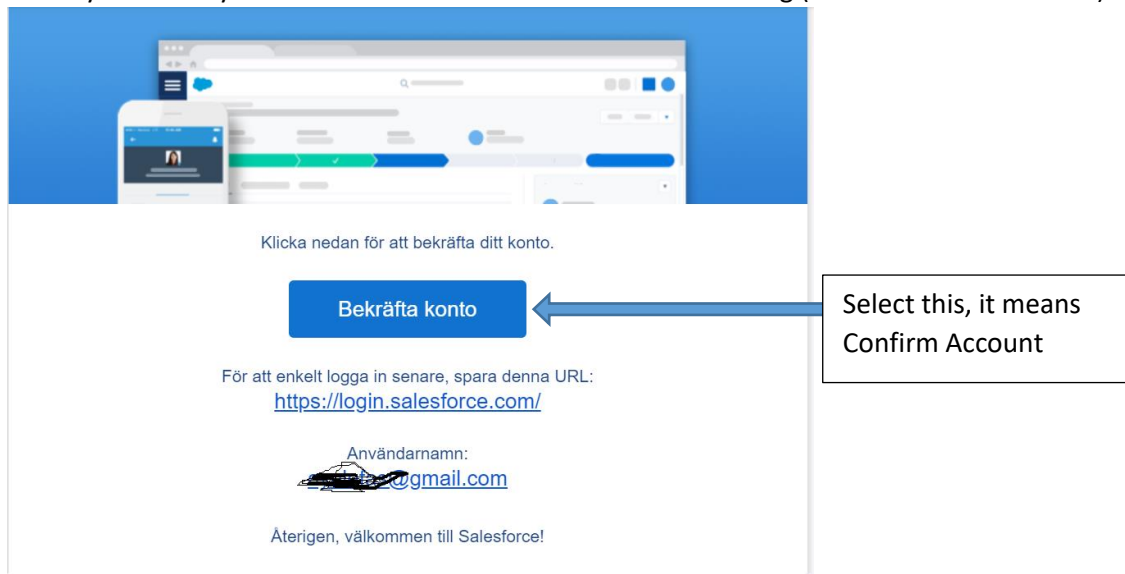
- Select **Signup**
- Enter your information:
 - **Name** – your **First Name** and **Last Name**
 - **Email** – your email – this should be a real email
 - **Role** – Developer
 - **Company** – Enter **SystInt2022**
 - **Country** – Sweden
 - **Postal Code** – you can enter your postal code here
 - **Username** - can be the same as your email address but it should not be used by any other individual elsewhere (ex. workneh1@dsv.su.se ← this is just an example, so please do not use it as it is already taken – but use your SU email)
 - Check that you agree with the **Terms of Use**, the **Master Subscription Agreement**, as described in the **Privacy Statement** then click **Sign me up >**




Almost there...
Please check your email to confirm your account.



- Check your email you will find information similar to the following (THIS CAN TAKE A WHILE)



- After selecting **Confirm Account**, enter your password twice. Your password must have at least eight characters containing at least one digit and at least a letter (Aa to Zz). For now, you can use **SystInt2022** as your password.
- Select your security question, and answer > then click on **Change password (Ändra lösenord)**




Welcome to salesforce.com!

For security reasons, your password must now be changed. Your password was last changed or reset on 18/05/2015 13:54

User Name

New Password

Acceptable A strong password contains at least 8 alphanumeric characters and includes uppercase and lowercase letters.

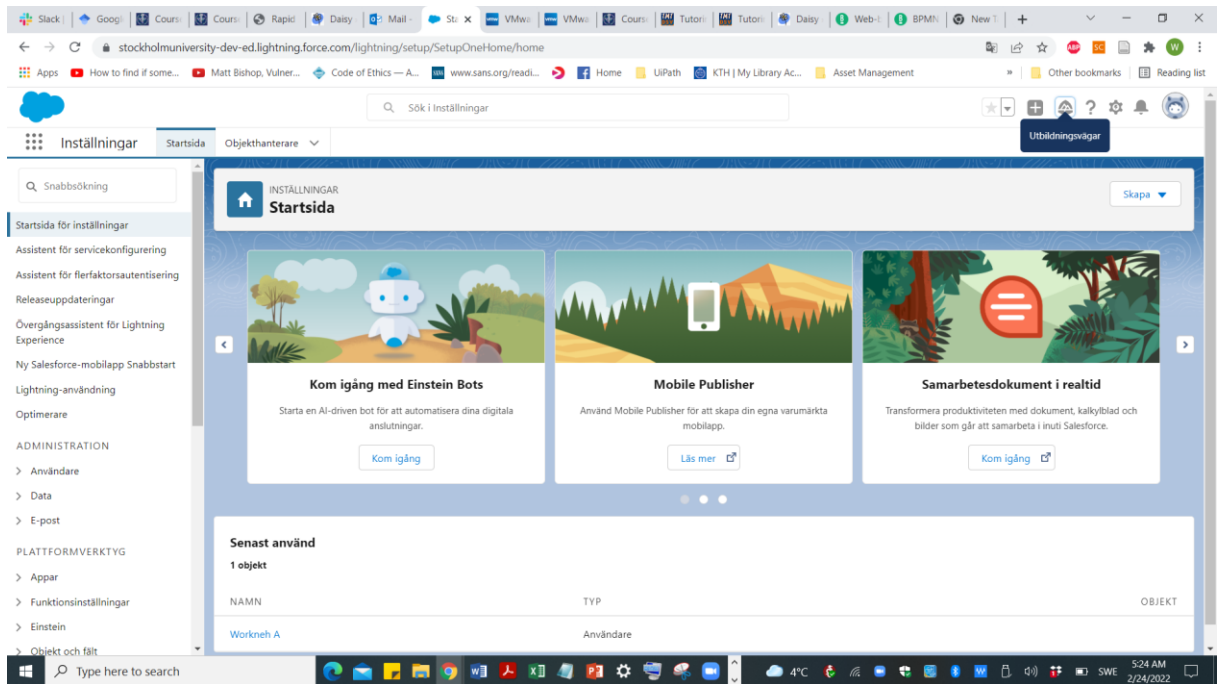
Verify New Password 

Question

Answer

We will ask you to answer this question if you forget your password.

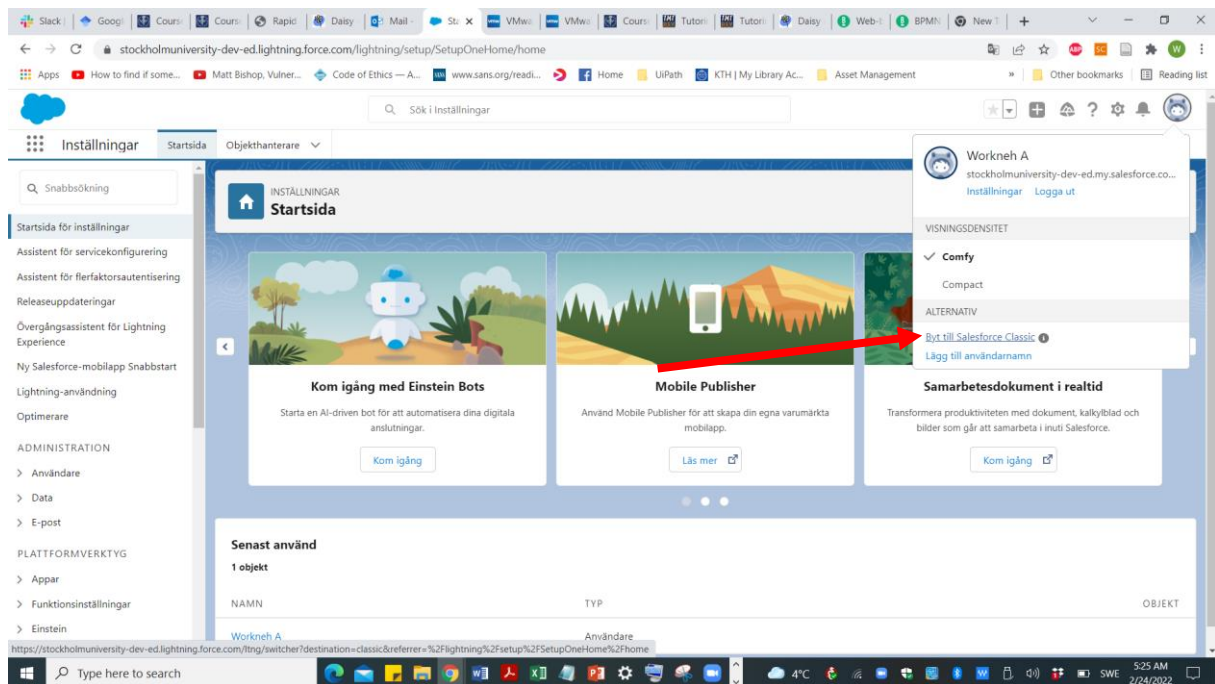
(This could be in Swedish)



1.2 Changing to classic mode

Salesforce has classic mode or light mode; the instructions in this manual are written to assist you in classic mode. So change it to classic model; to change your Salesforce to the classic:

- Click on the User Login Icon as illustrated below
- Select – Switch to classic mode (**Byt till Salesforce Classic**)



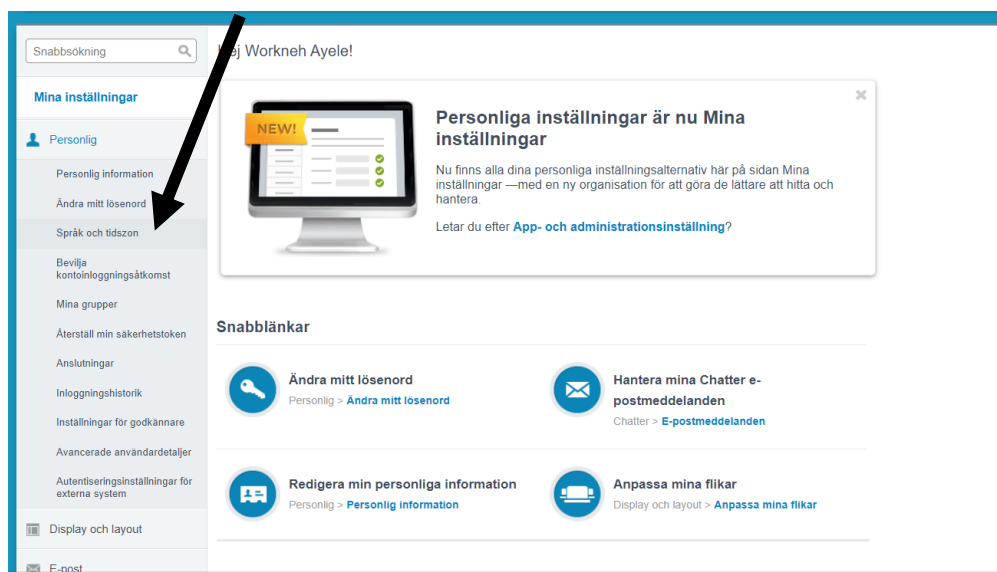
1.3 Changing the default language to English

If the default language is English, you can skip this Section. Otherwise, follow the following steps.

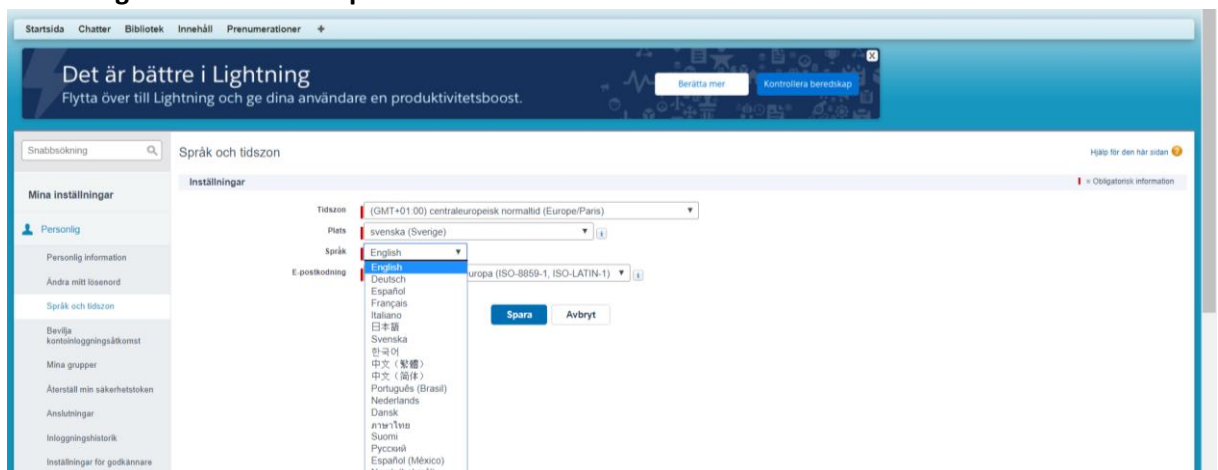
- If the language is Swedish click on **Mina inställningar** under your user name as illustrated below



- Under **Mina Inställningar**
- Select **Språk och Tidszon**



- Select **English** then click on **Spara**



1.4 Salesforce and browser options

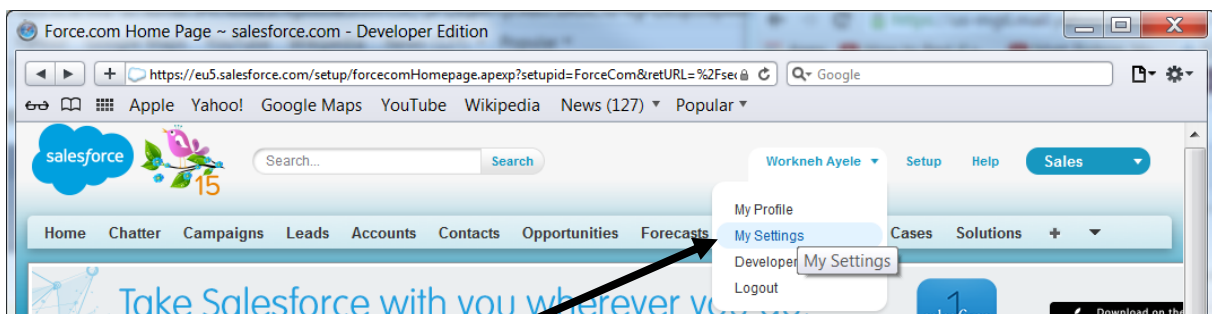
Salesforce is an application provisioned as a cloud service. Cloud computing has three service models in general, such as Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS). If you are using cloud-based applications where you don't have control over the internal infrastructure and operating environment, then the service model is referred to as SaaS. If

you are getting platforms such as operating systems to configure and install run your applications, then you are using the PaaS service model. On the other hand, if you have full control over the infrastructure, such as the hardware – CPU, GPU, etc., where you could install your operating environment and applications, then you are on the IaaS service model.

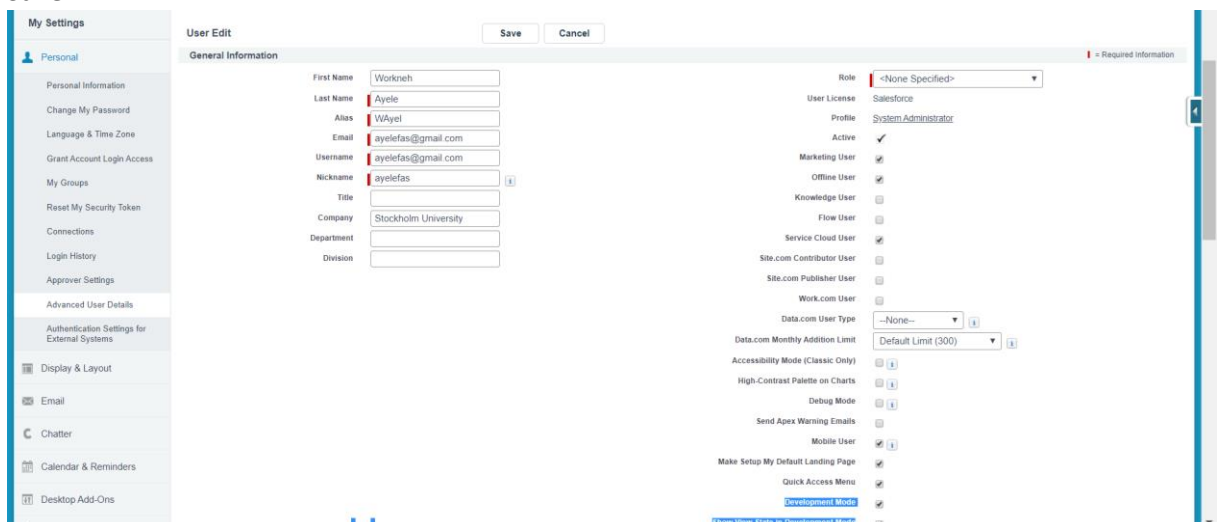
Salesforce provides SaaS (salesforce.com) and PaaS (Force.com). If you choose the first option, then you have the Customer-Relationship-Management-as a service referred (CRM as a Service) or SaaS. Salesforce.com is the SaaS interface where you do actual customer case handling and support. To access the PaaS of Salesforce, you need to browse to Force.com, where you can customize Salesforce itself, integrate Salesforce with other systems, develop applications and pages, and do other advanced customization tasks. To switch to Salesforce.com, click on **Home**, and if you are in Salesforce.com to switch to Force.com, click on **Setup**.

1.5 Enabling developer mode

To use Salesforce API or to use the Force.com then you need to enable developer mode



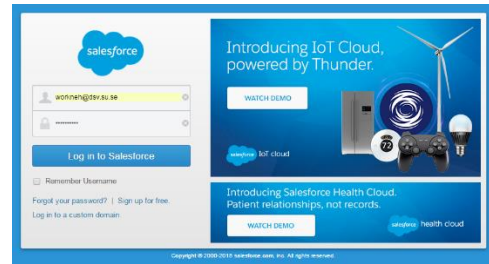
- Select the arrow next to your name, as illustrated above
- Select **My Settings**
- Under **Personal information**, select **Personal**, select **Advanced User Details**
- Select **Edit**
- Now you can check **Developer Mode** and **Show View State in Developer Mode**, then click **Save**



1.6 To get support from Salesforce Forum (You can use the forum if you have issues, skip it for now)

To use the Salesforce forum:

- Enter **developer.salesforce.com**
- Enter user name and password
- Then **Log in to Salesforce**



Your developer account will enable you to access resources, forums, and so on.

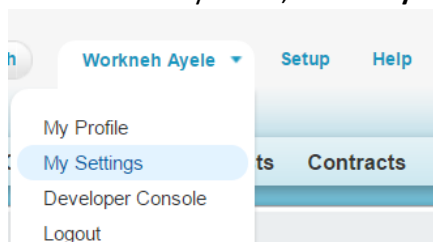
1.7 Generating security token

When you access Salesforce from an IP address that isn't trusted for your company and use a desktop client or the API, you need a security token to log in.

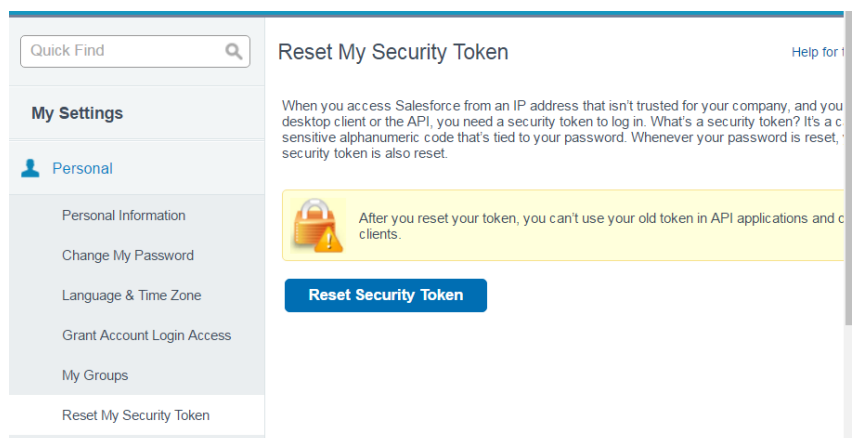
What's a security token? It's a case-sensitive alphanumeric code that's tied to your password. Whenever your password is reset, your security token is also reset. If the security token is not reset, then there is no security token, and you can use your password.

Steps

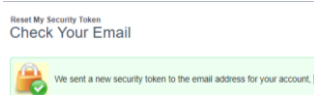
- To reset security token, Select **My Settings** from your account name as illustrated below



- Under **Personal** you can select **Reset My Security Token**
- Click on **Reset My Security Token**



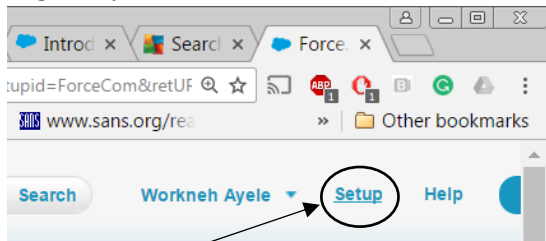
- Your security token will be sent to your email. Note that you need to use this security token in your application to integrate your cloud-based CRM application with applications.



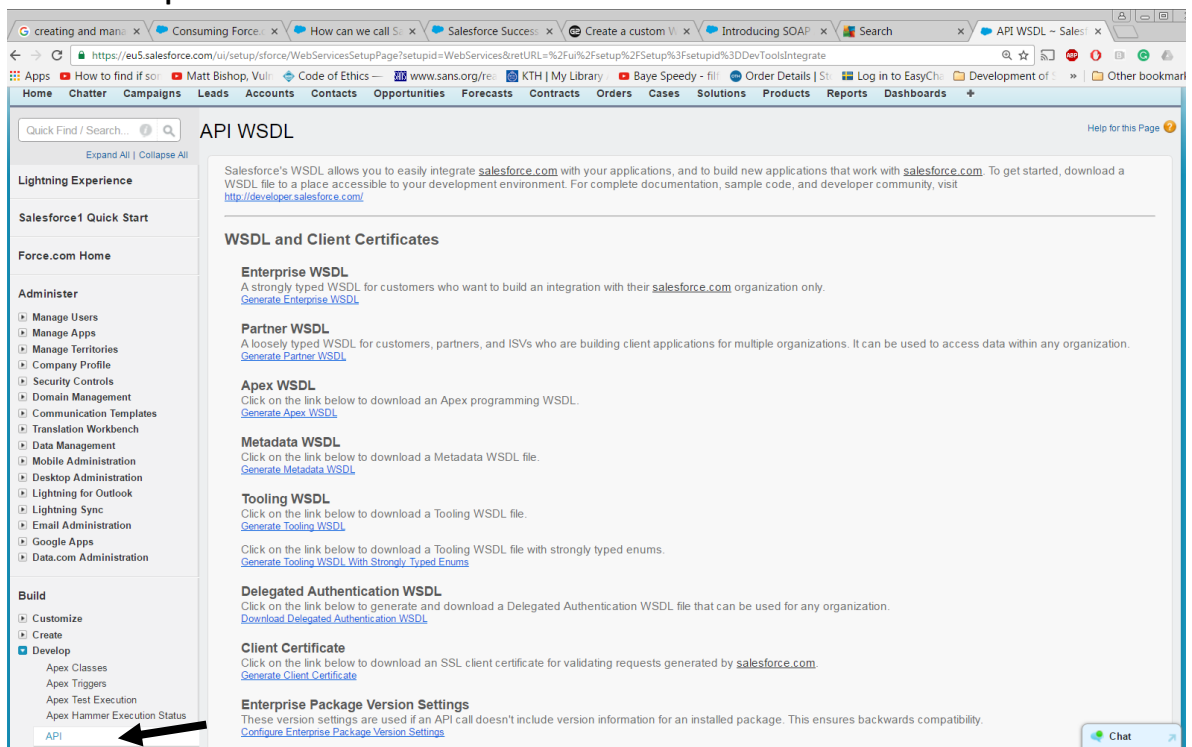
1.8 Get the WSDL SOAP API

Steps

- Login to your salesforce.com account



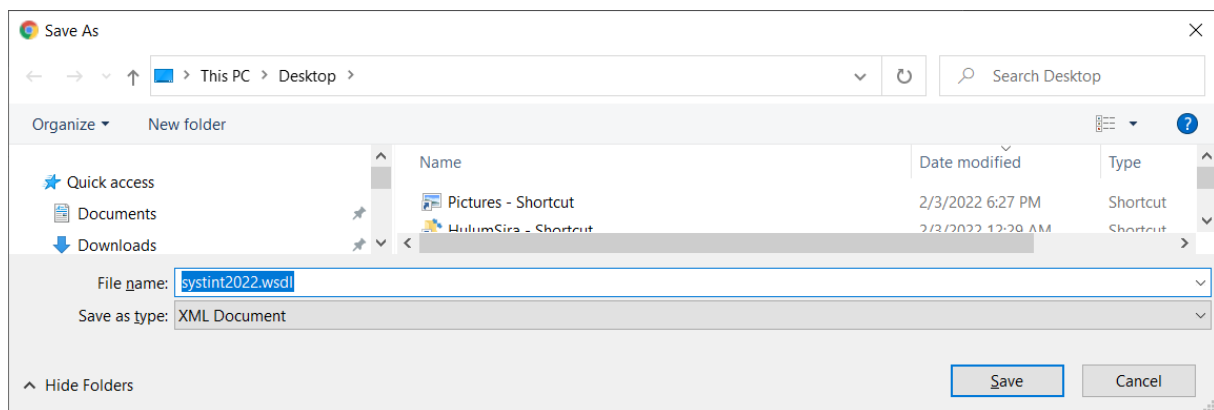
- Select **Setup**
- Go to **Develop** on the left sidebar under **Build** then select **API**



- Select **Generate Enterprise WSDL**



- Right click on the code generated then click **Save as**, save it in your computer in your project folder - **C:\AllFiles\Task4\systint2022.wsdl**.



2. Creating Integrated Windows Application with Salesforce

2.1. Adding Products in Salesforce

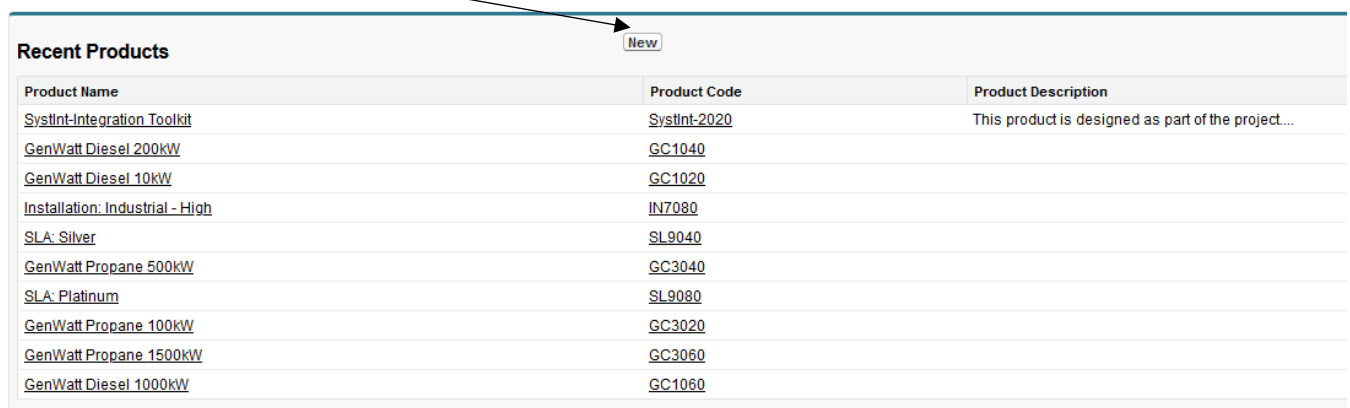
- Login to your Salesforce
- <https://www.salesforce.com/se/?ir=1>
- enter your username and password if you are not logged in
- click on **Home** then click on **+**



- Select **Products**



- Click on **New**



- Under Product Name – enter **System Integration Toolkit**
- Under Product Code – enter **SystInt-2022**

- Click on **Save**
- Click on **Add** under **Standard Price**
- Enter a price value as illustrated below, then click on **Save**

TODO: Create three other more products with the same product code (SystInt-2022) as you just did

Product names

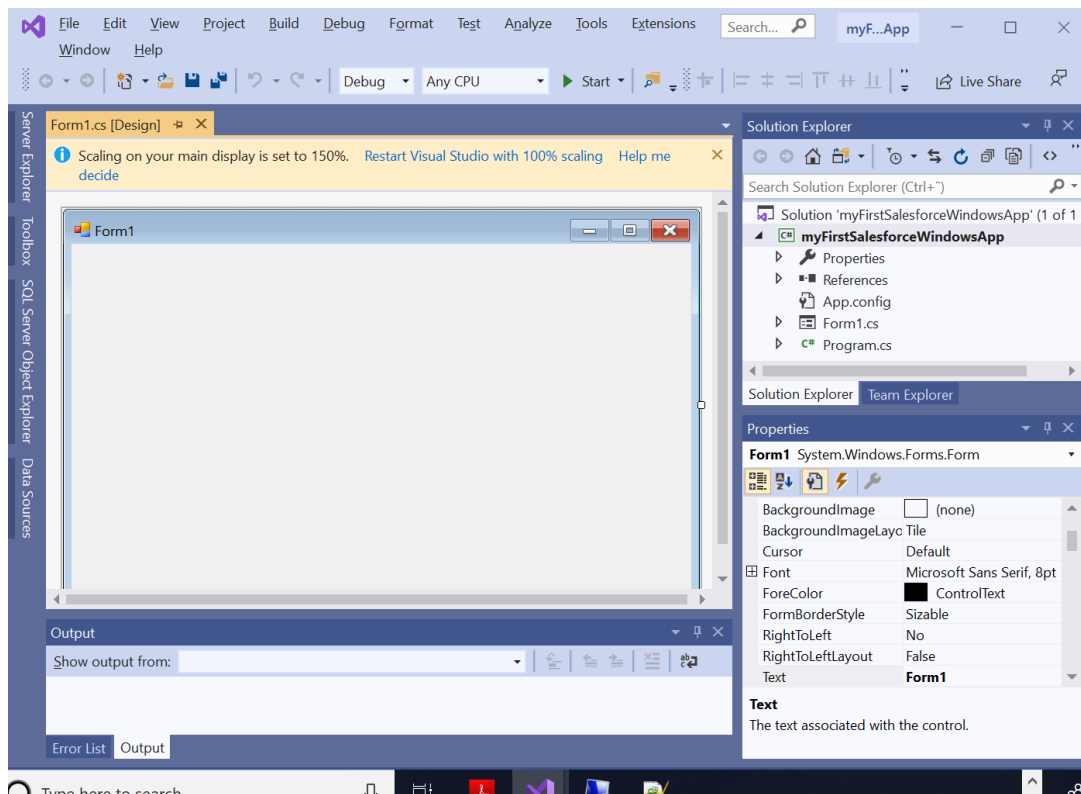
- Visual studio (1500 kr.)
- Intel Server (45000 kr.)
- Router (25000 kr.)

2.2. Creating Integrated Windows .NET Application

Steps

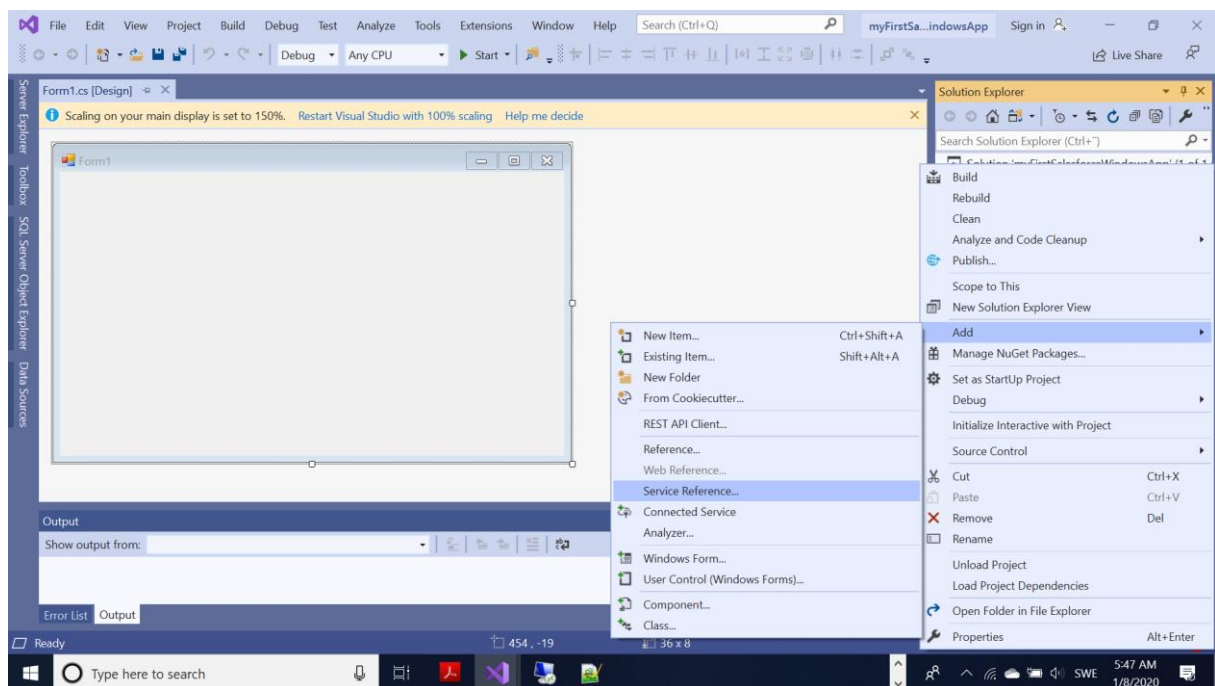
- Start a new **Visual Studio Community** instance by using the **Start** menu
- Select **Windows Forms Application (.NET Framework)** after selecting **Visual C#**
- Enter the applications name – **myFirstSalesforceWindowsApp**
- Specify **Location** to be – C:\AllFiles\Task4

- Click on **Create**, and your Windows Form will be created, as illustrated below



To create the reference to the service

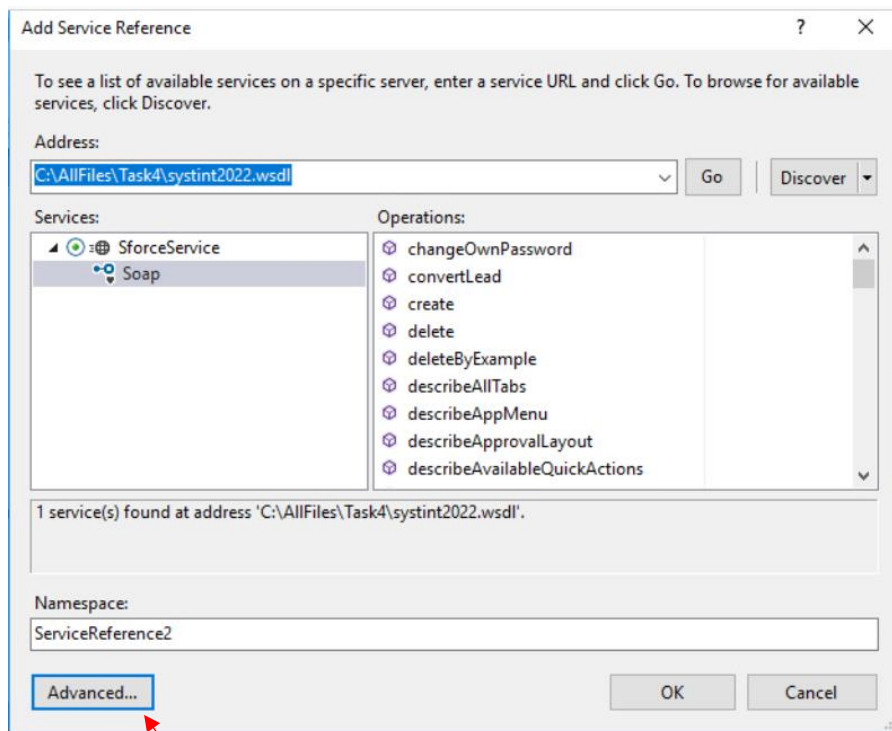
- Right-click on your **Project** > Point to **Add** > Select **Services References**



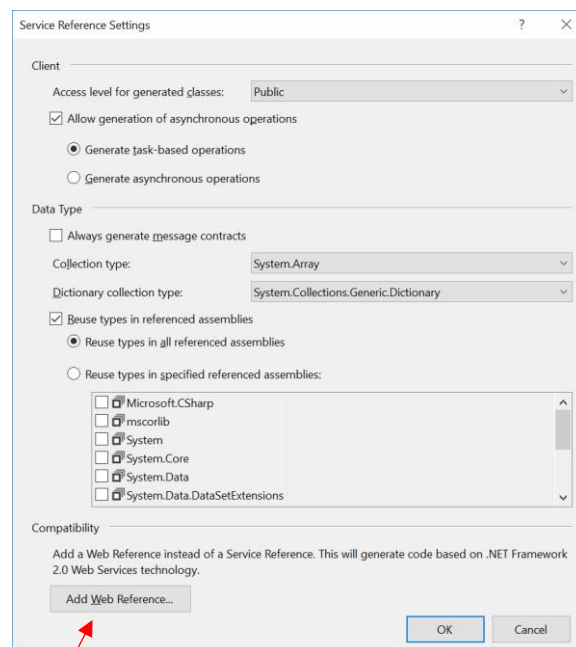
- Add the address of the WSDL file, i.e., systint2022.wsdl
 - The address is: **"C:\AllFiles\Task4\systint2022.wsdl"**

- Then click **Go**
- Enter **ServiceReference2** as the name, and for illustration, see the snapshot below

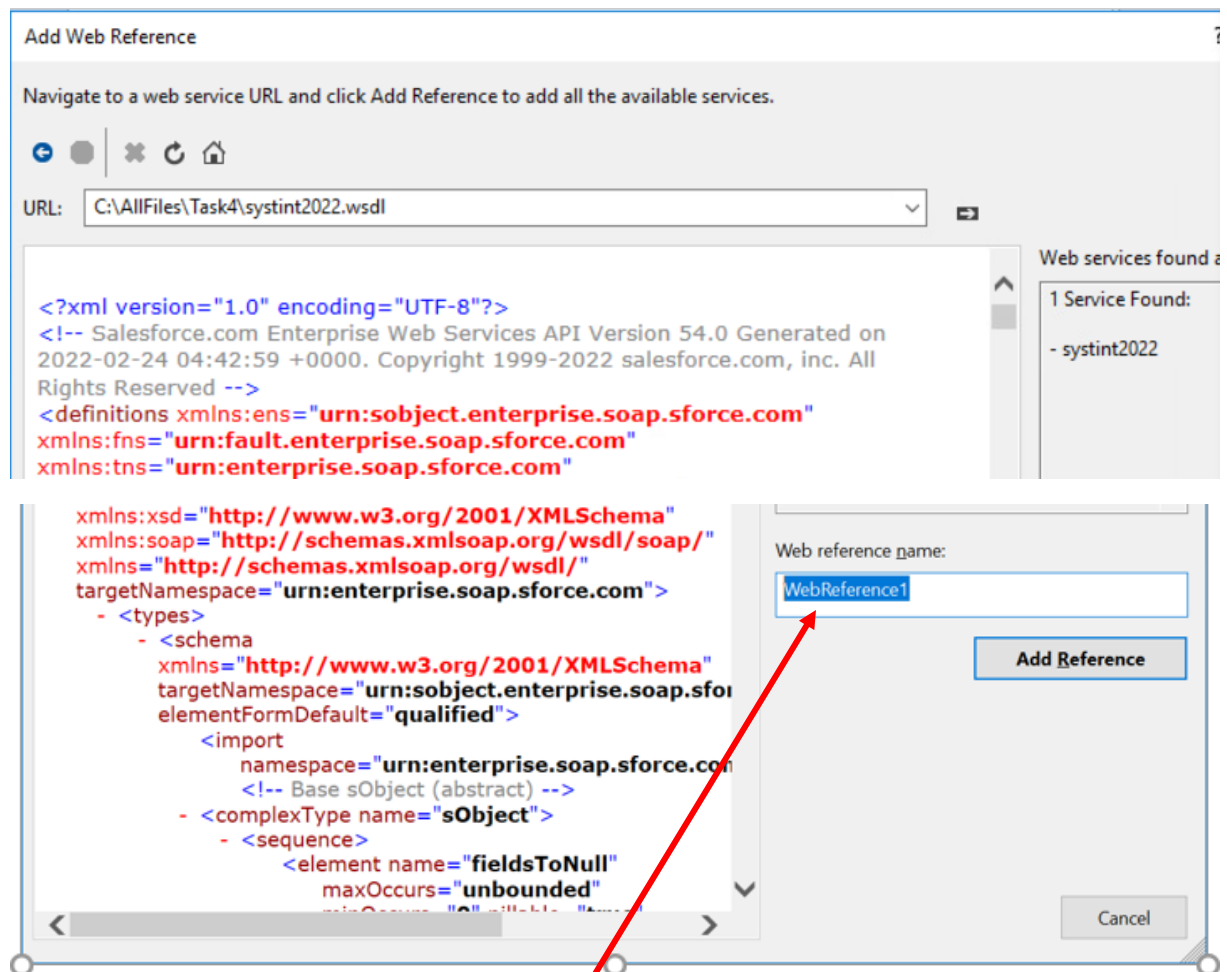
Note: You can see the functions available in the service.



- Click on **Advanced...**

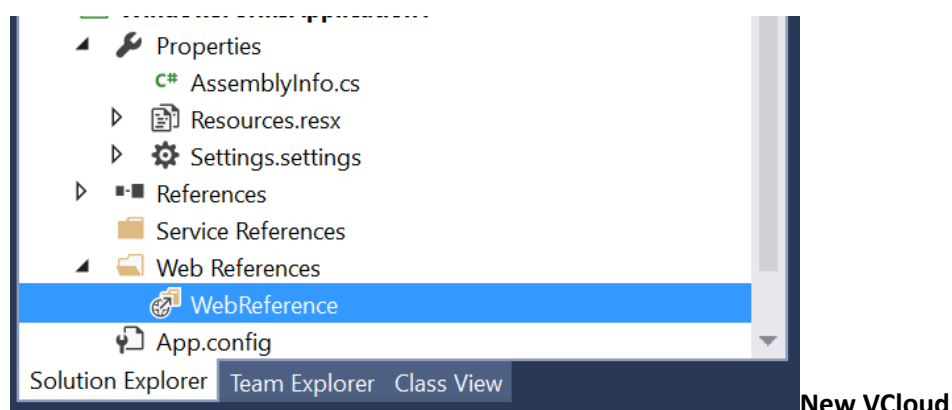


- Select **Add Web Reference**
- Enter URL: **"C:\AllFiles\Task4\systint2022.wsdl"** and click **Go**



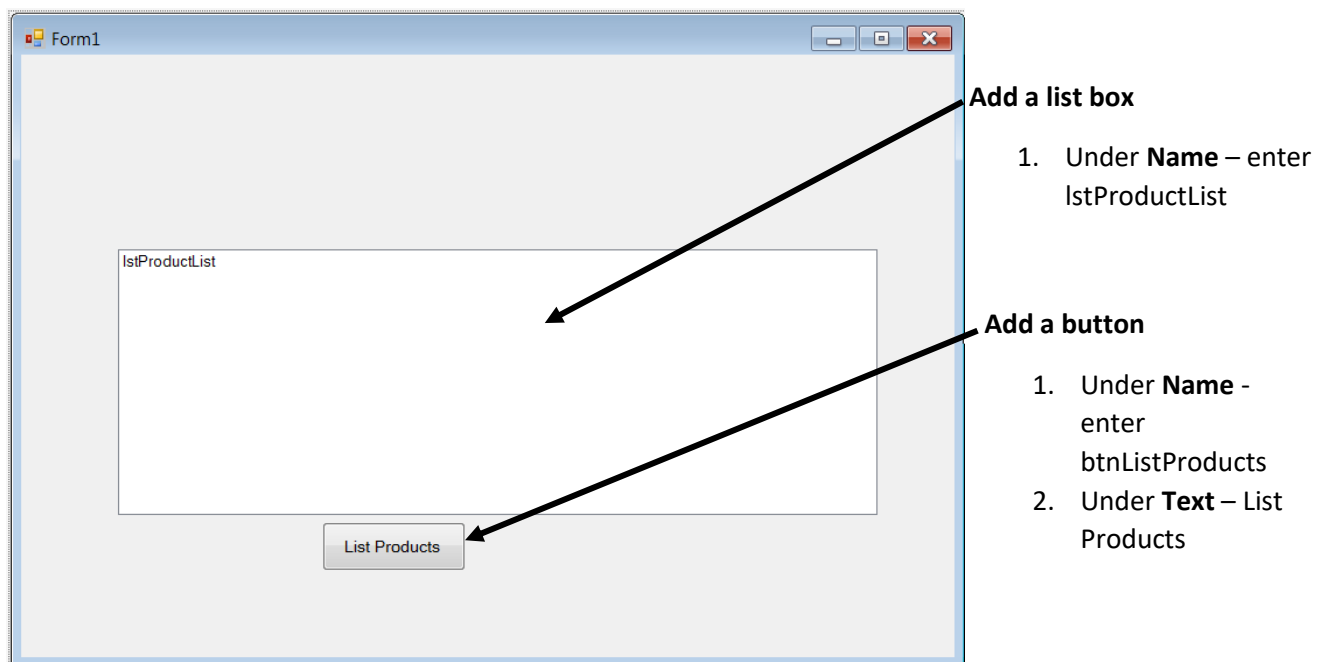
- Enter web service name – **WebReference1** under **Web reference name:**
- Click on **Add Reference**

Your solution explorer will look similar to the snapshot illustrated below



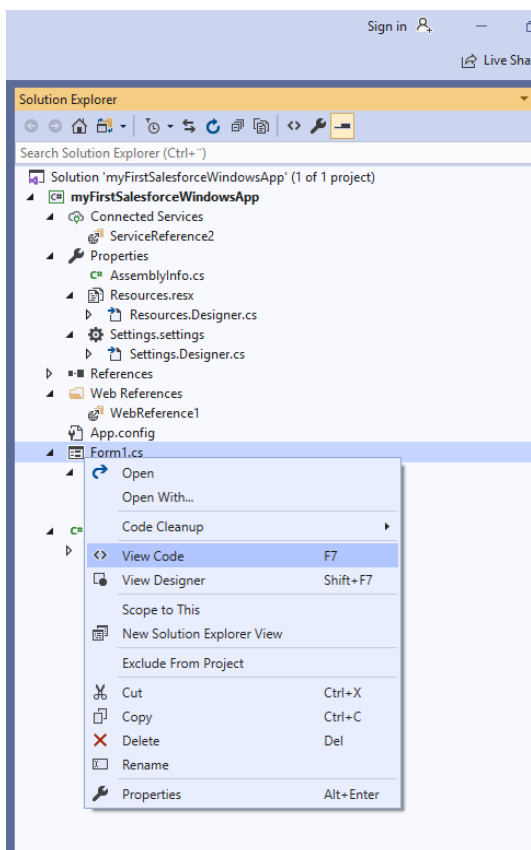
2.3. Adding controls to Form1 designer

Here you will add a list box, and the button, the list box will populate the list of products from your Salesforce. Now the windows application consuming the service with the reference setup to the **Enterprise WSDL** is created, you can now transition to writing code that calls the API.



2.4. Adding the code that makes the program work

- Under **Form1** in the **Solution Explorer** right click on **Form1.cs** > then click on **View Code**



- The code is illustrated below, Form1.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Net;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using myFirstSalesforceWindowsApp.WebReference1;

namespace myFirstSalesforceWindowsApp
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void btnListProducts_Click(object sender, EventArgs e)
        {
            QueryResult queryResult = null;
            String SOQL = "";
            string productid = "";
            SOQL = "select Id, Name, ProductCode, Description, CreatedDate,
StockKeepingUnit from Product2 where ProductCode = 'SystInt-2022'";
            queryResult = Login.sfdcBindPublic.query(SOQL);

            if (queryResult.size > 0)
            {
                Product2 readProduct;// (Product2)queryResult.records[i];
                int j = queryResult.size;
                for (int i = 0; i < j; i++)
                {
                    readProduct = (Product2)queryResult.records[i];
                    string productName = readProduct.Name;
                    productid = readProduct.Id; lstProductList.Items.Add(productid + " "
+ productName + " " + readProduct.ProductCode + " " + readProduct.Description + " " +
readProduct.CreatedDate + " " + readProduct.StockKeepingUnit);
                }
            }
            else
            {
                //if no record is returned the the message below will be displayed
                string message = "No records returned.";
                MessageBox.Show(message);
            }
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            ServicePointManager.Expect100Continue = true;
            ServicePointManager.SecurityProtocol = SecurityProtocolType.Tls12;//.Tls11;
            //Ssl3;

            SforceService sfdcBinding = null;
            string accountName = "xxx.yyy@yaa.com";
            string accountPassword = "your_password_here";
            string accountToken = "yourTOKEN_here";
            sfdcBinding = new SforceService();
            LoginResult loginResult = sfdcBinding.login(accountName, accountPassword +
accountToken);

```



```

        if (Login.MyLogin(accountName, accountPassword, accountToken, sfdcBinding,
loginResult))
        {
            MessageBox.Show("Login Succeeded");
            //Change the binding to the new endpoint
            sfdcBinding.Url = loginResult.serverUrl;
        }
        else
        {
            MessageBox.Show("Login failed, windows will close now");
            Form1.ActiveForm.Close();
        }

        // a new session and a header object need to be created and set
        sfdcBinding.SessionHeaderValue = new SessionHeader();
        sfdcBinding.SessionHeaderValue.SessionId = loginResult.sessionId;
    }
    static class Login
    {
        public static LoginResult loginResultPublic;
        public static SforceService sfdcBindPublic;
        public static bool MyLogin(string accName, string accPassword, string
accToken, SforceService sfdcBind, LoginResult myLog)
        {
            try
            {
                myLog = sfdcBind.login(accName, accPassword + accToken);
                loginResultPublic = myLog;
                sfdcBindPublic = sfdcBind;
                //loginResult = client.login(null, accountName, accountPassword +
accountToken);
            }
            catch (System.Web.Services.Protocols.SoapException e)
            {
                MessageBox.Show("Login failed, login error - " + e.Message + "\n" +
e.StackTrace);
                //Form1.ActiveForm.Close();
                sfdcBind = null;
                return false;
            }
            return true;
        }
    }
}

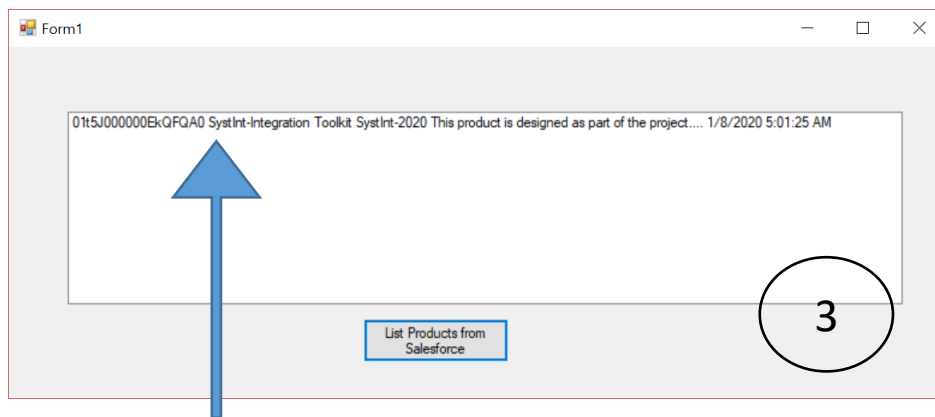
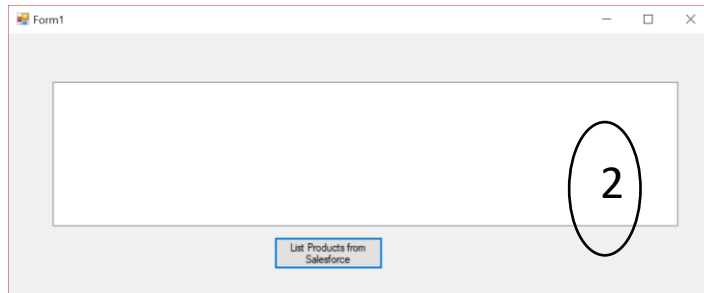
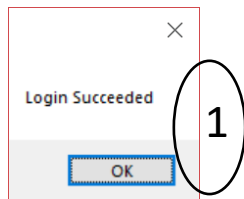
```

2.5. Testing if the code works

- Select **Start** from the toolbox
- You will probably get an error, error CS0030, or something like this. Try to fix this error by yourself
 - If you find it difficult to solve this problem, refer to Appendix 1
- After you fix the error, when you start your program, you will see a message-box showing you the status of your login (1), i.e., succeeded or failed to login, then you will see a blank form (2), and finally, if you click on **List Products** button from Salesforce, then you will see your results (3)

- If you get **"The request was aborted: Could not create SSL/TLS secure"** as an error message-
<https://stackoverflow.com/questions/12888569/soap-client-could-not-create-ssl-tls-secure-channel>

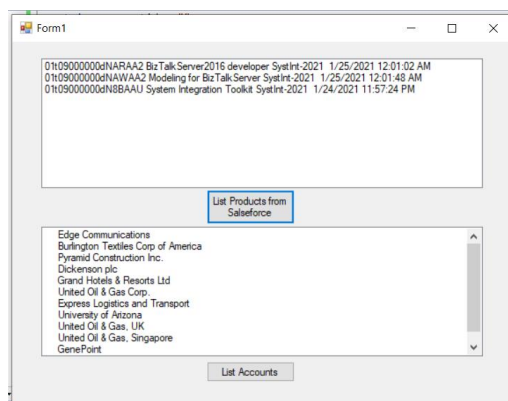
Now you can try the code



It should display data

Exercise question

1. How can you display all products? (Mandatory)
 - a. During the demonstration, you will explain the answer to the question
2. (Optional) Add a list box and a button to pull Account information from Salesforce; write the code that will allow you to display account name, account location, and one or more account information.



Reference

1. https://help.salesforce.com/HTViewHelpDoc?id=user_security_token.htm
2. <http://www.ashishblog.com/integrating-salesforce-soap-api-using-c-net/>
3. https://developer.salesforce.com/page/Consuming_Force.com_SOAP_and_REST_Web_Services_from_.NET_Applications
4. <http://www.jitendrazaa.com/blog/salesforce/create-a-custom-web-service-in-salesforce-and-consume-it-in-c-net-application/>

To fix error related with wsdl file

5. <https://developer.salesforce.com/forums/?id=906F0000000Aj5kIAC>

Appendix 1 – How to solve the error CS0030

Here is how you solve the problem

The address here should be your project address, and it might be different from what you see on the screenshot

- Find **Reference.cs** file in your project folder then find all instances of '[]' and replace them with '['

