SYSTINT Project VT2025

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

Part I – Creating Salesforce Developer Account and Generating WSDL

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:

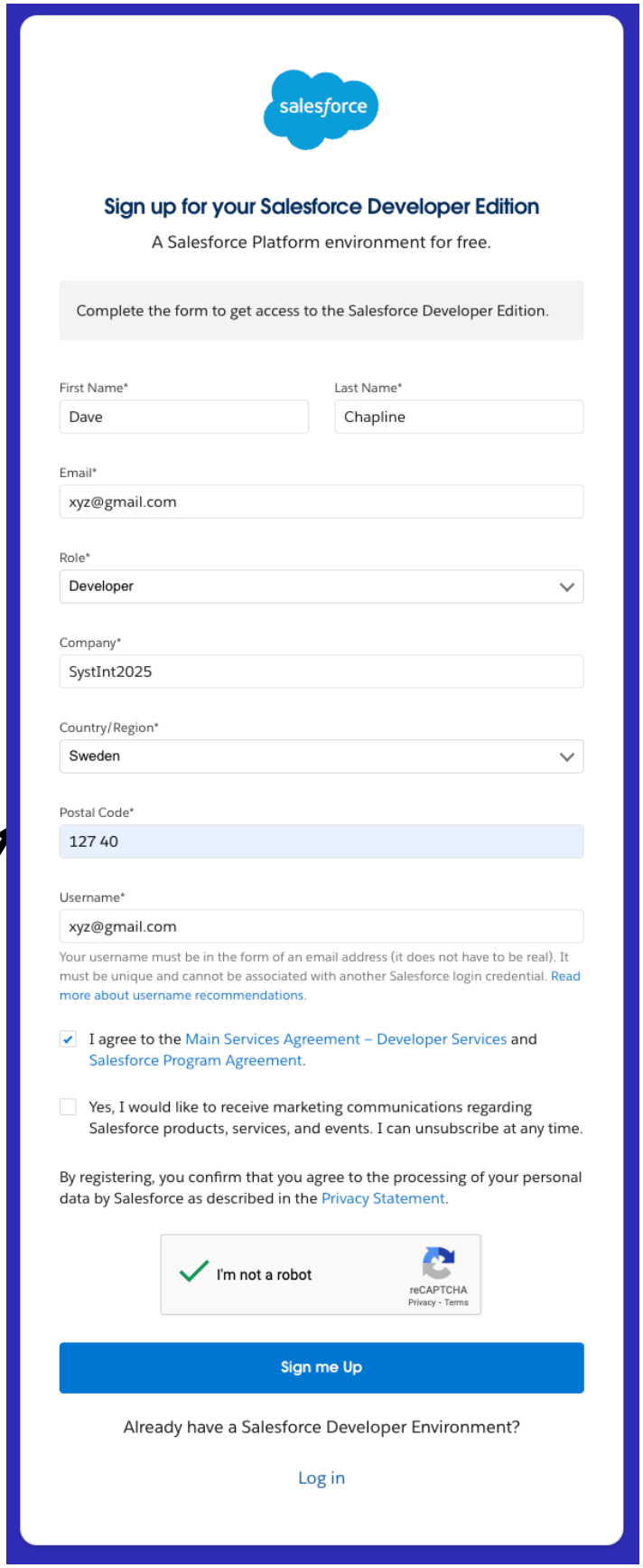
- username and password
- Security Token
- SOAP API (WSDL file)

1.1 Setting UP Salesforce Developer Account

1.2

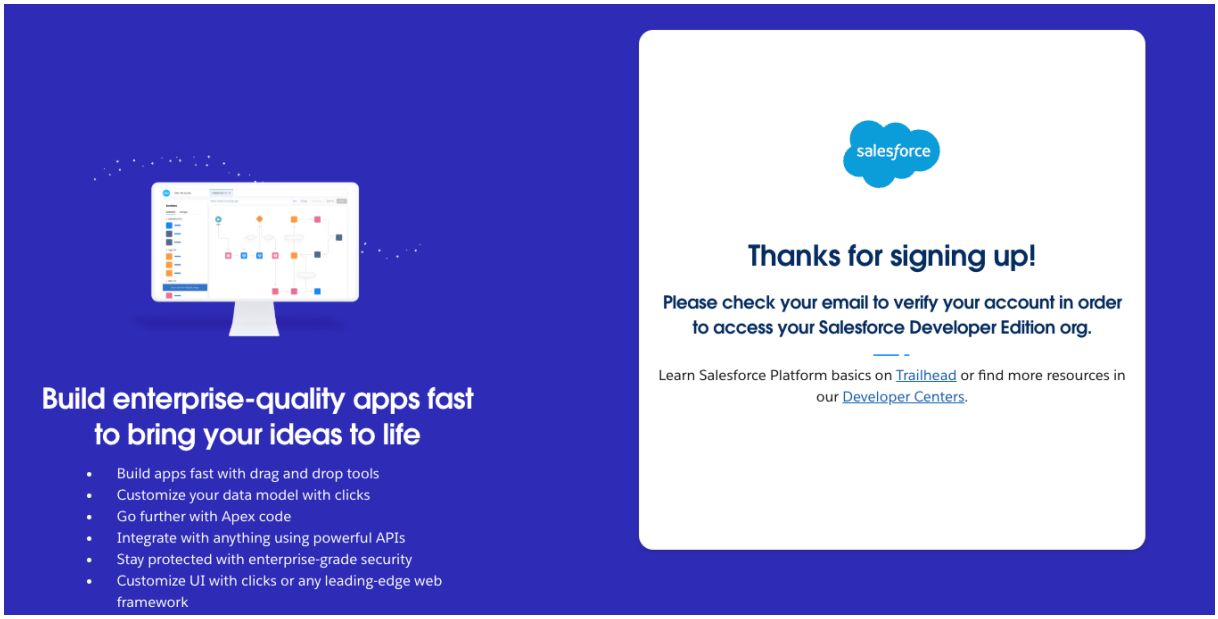
Steps

- Open a browser preferably Firefox (Chrome also works)
- Enter – https://developer.salesforce.com/signup in the address bar > then press Enter
- Enter your information:
    - **Name** – your **First Name** and **Last Name**
    - **Email** – your email – this should be a real email
    - **Role** – Developer
    - **Company –** Enter **SystInt2025**
    - **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

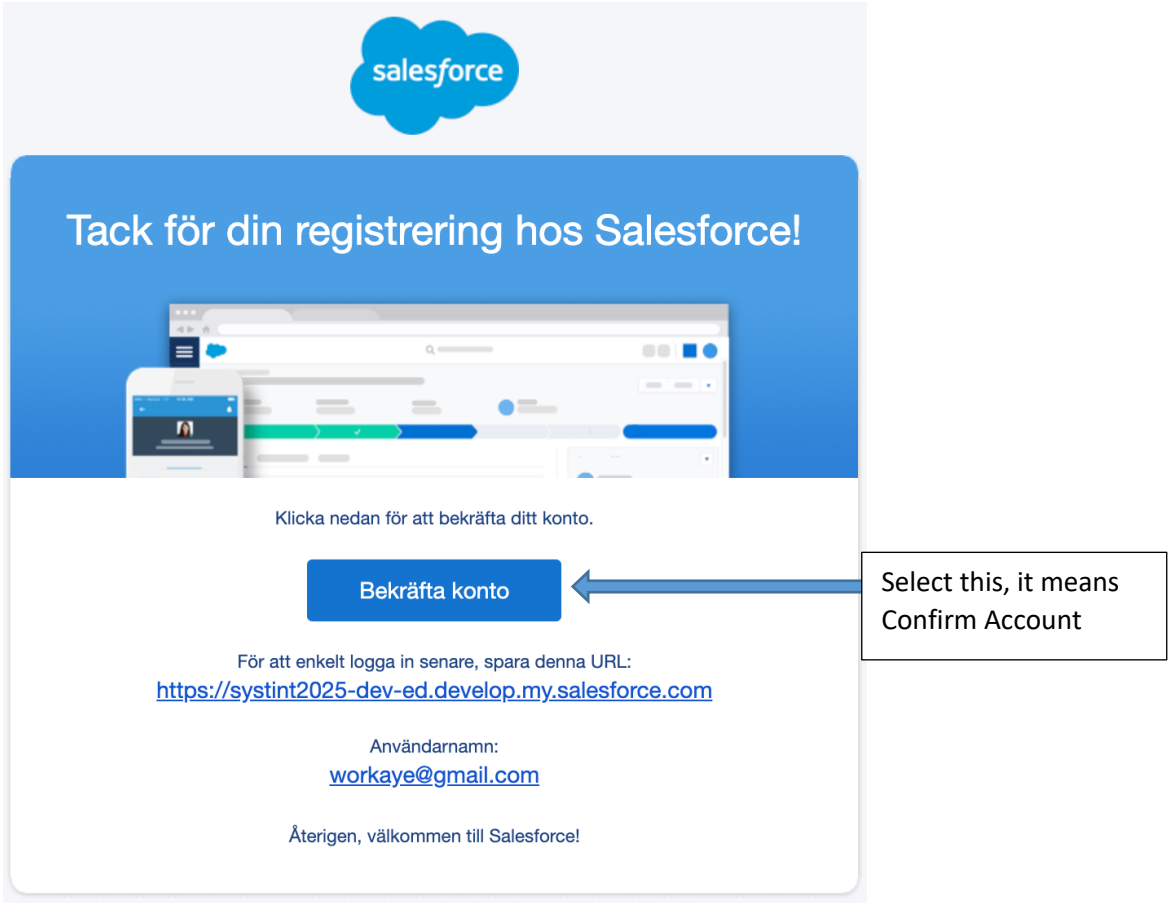


elsewhere (ex. xyz1@dsv.su.se ← this is just an example, so please do not use it as it is already taken – but use your SU email)

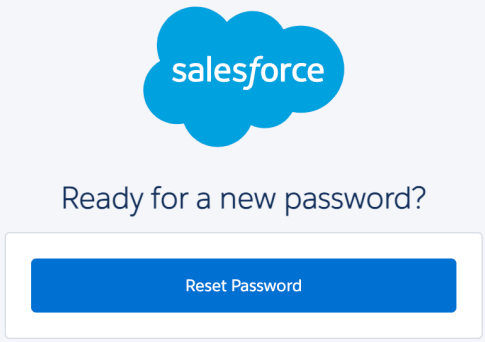
- Check that you agree with the **Terms of Use**, the **Master Subscription Agreement**, as described in the **Privacy Statement** then click **Sign me Up >**



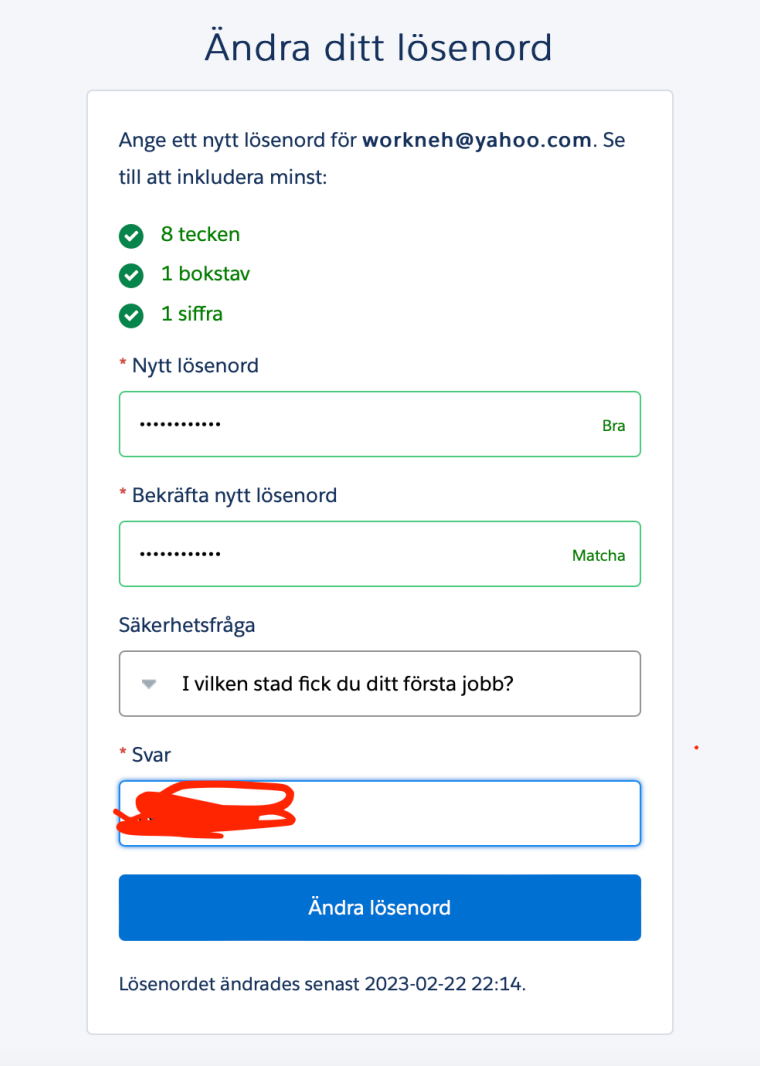
- Check your email you will find information similar to the following (THIS CAN TAKE A WHILE)



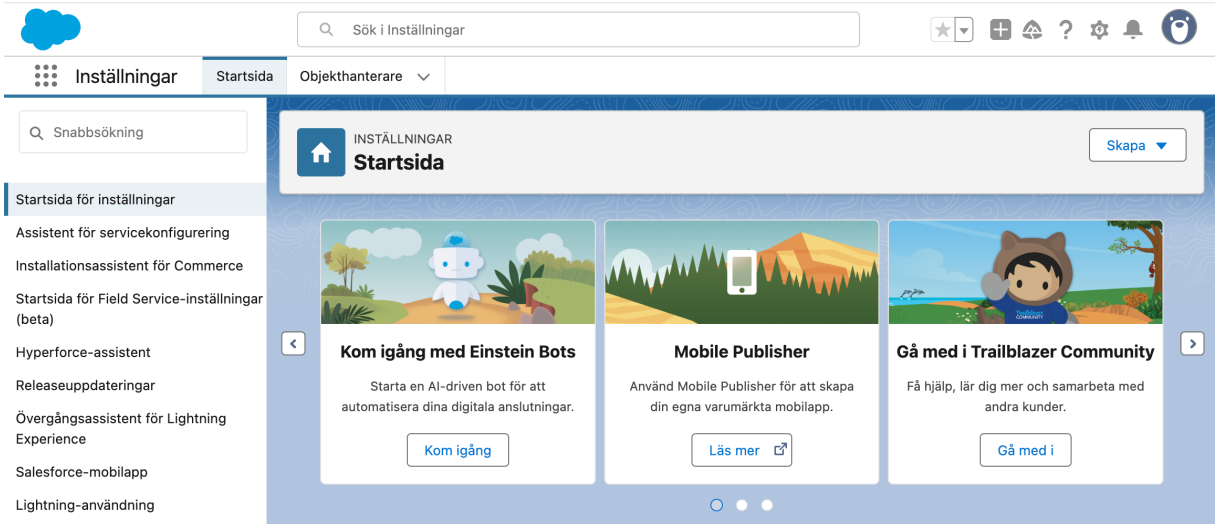
- After selecting **Confirm Account**, you may be prompted to click on **Reset Password**, as illustrated below.



- 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 **SystInt2025** as your password.
- Select your security question, and specify your answer > then specify password two times **Change password**



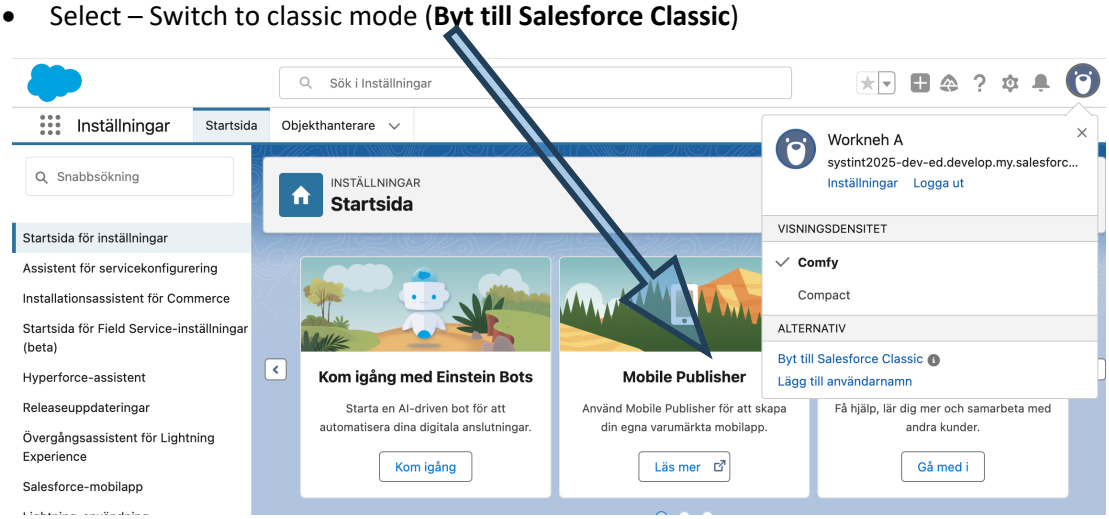
(This could be in Swedish):
TRANSLATION:
Nytt lösenord = **New password**
Bekräfta nytt lösenord = **Confirm the new password**

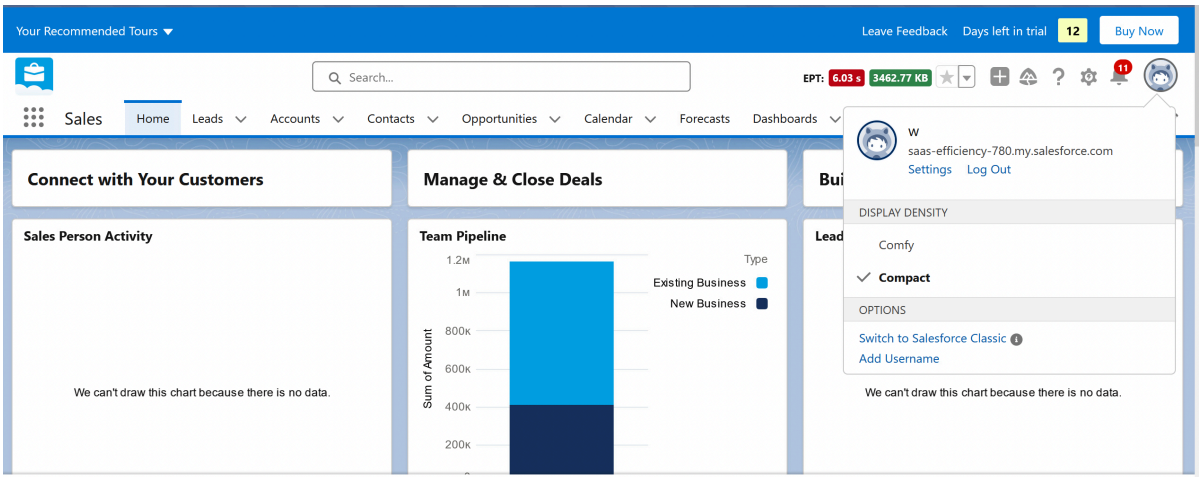


1.3 Changing to classic mode

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**)

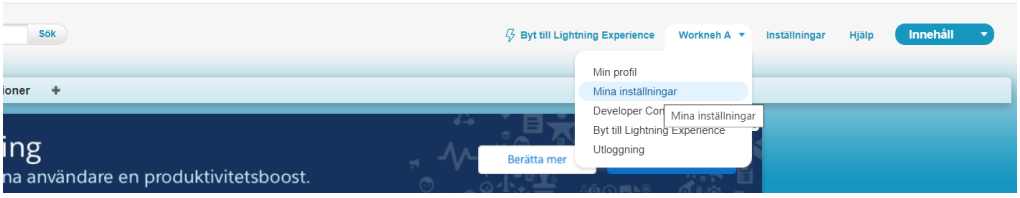




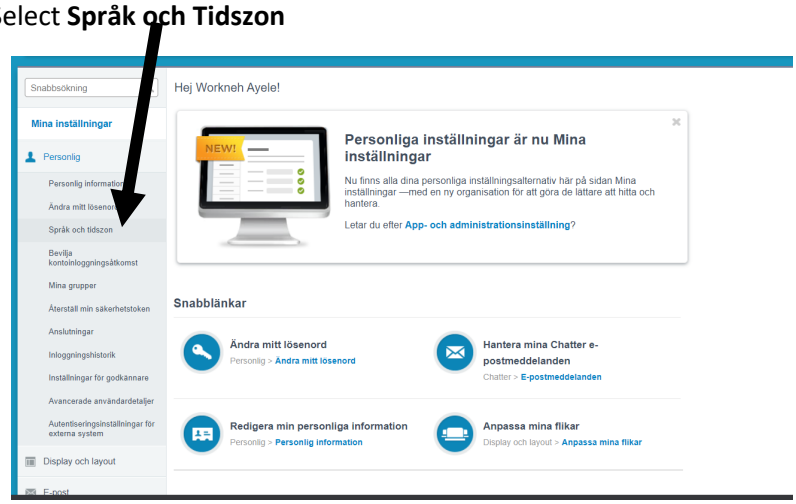
1.4 Changing the default language to English (If the language is already English you can skip this step)

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

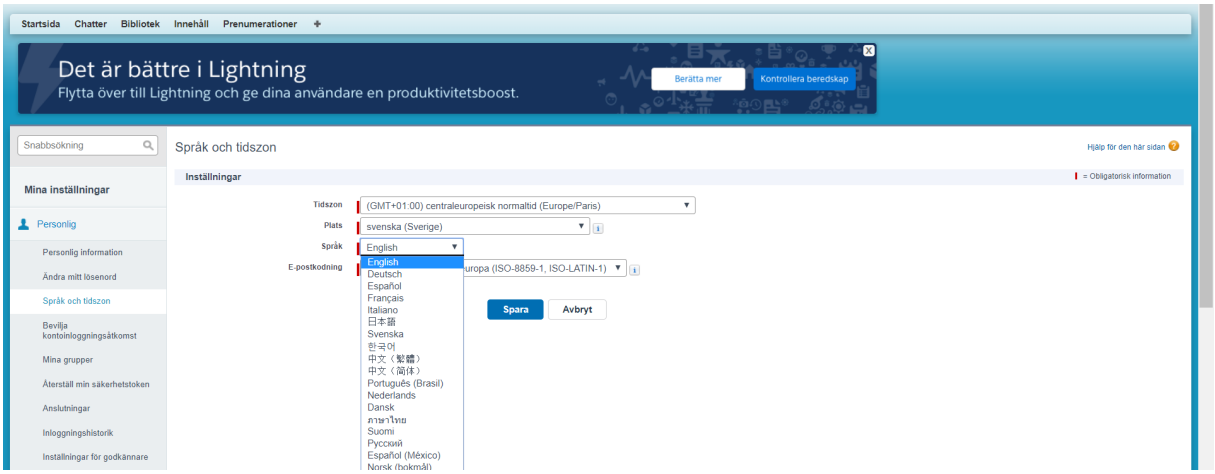
- If the language is Swedish click on **Mina inställningar** under your user name as illustrated below



- Under **Mina Inställningar >** under **Personlig Informatin**
- Select **Språk och Tidszon**



- Select **English** then click on **Spara (Save)**



1.5 Salesforce and browser options

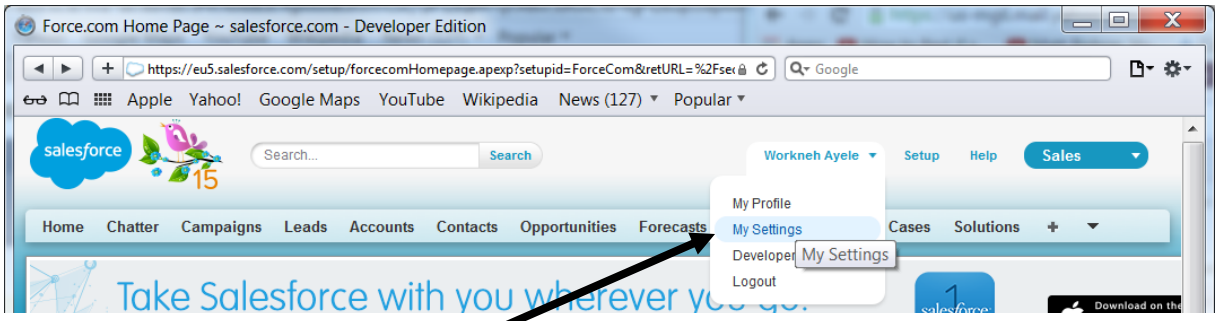
Salesforce is an application provisioned as a cloud service. Cloud computing has three service models in general, such as Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS). If you are using cloud-based applications where you don't have control over the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

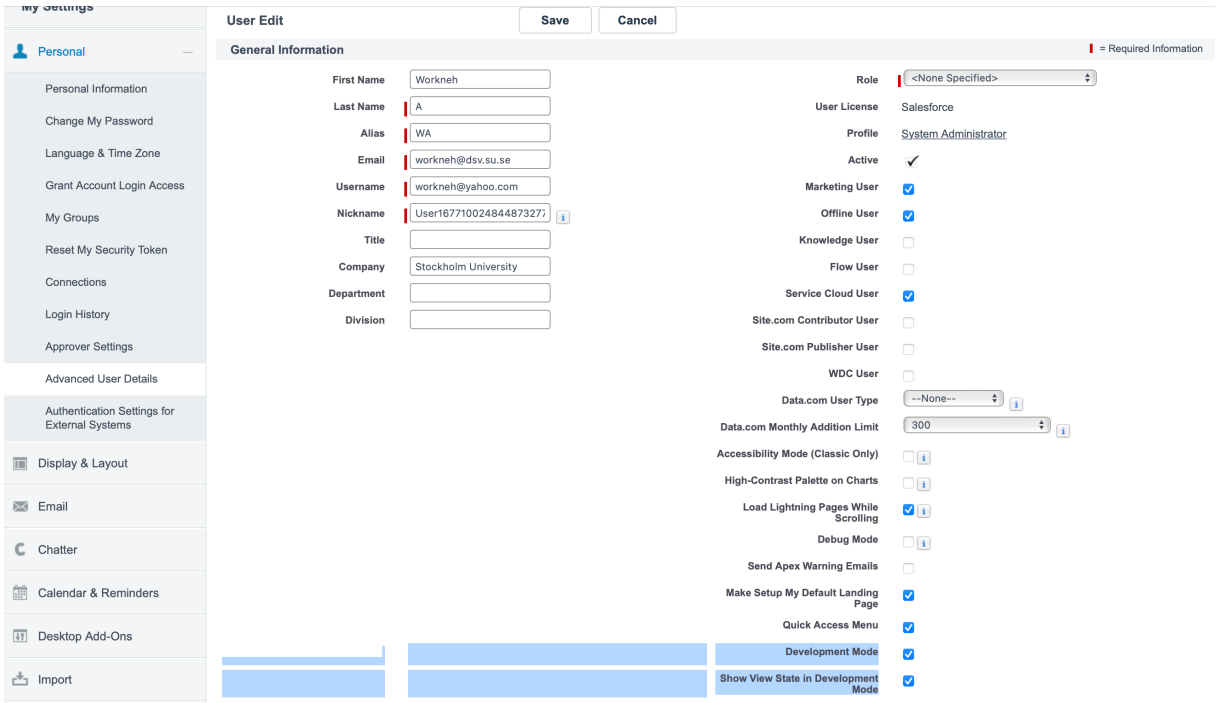
Salesforce provides SaaS (salesforce.com) and PaaS (Foce.com). If you choose the first option, then you have the Customer-Relationship-Management-as a service referred (CRM as a Service) or SaaS. Salesforce.com is the SaaS interface where you do actual customer case handling and support. To access the 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**.

1.6 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,** you can select **Personal information (**to see personal information**),** click on **Advanced User Details**
- Select **Edit**
- Now you can check both **Developer Mode** and **Show View State in Developer Mode,** then click **Save**



1.7 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**
- Select Salesforce under login
- Enter user name and password
- Then **Log in to Salesforce**

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

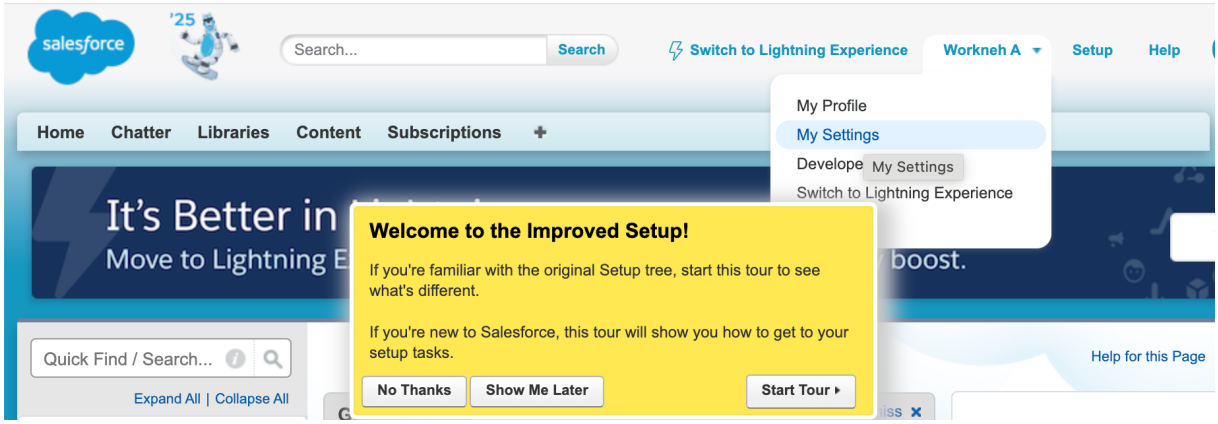
1.8 Generating security token

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.

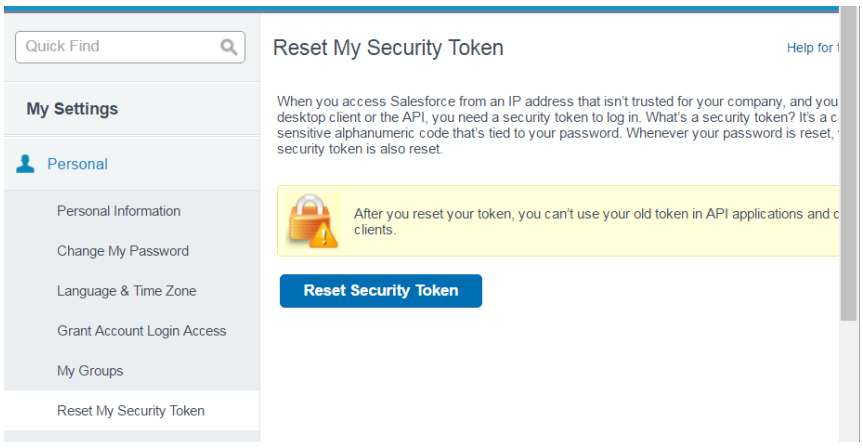
What's a security token? It's a case-sensitive alphanumeric code that's tied to your password. Whenever your password is reset, your security token is also reset. If the security token is not reset, then there is no security token, and you can use your password.

Steps

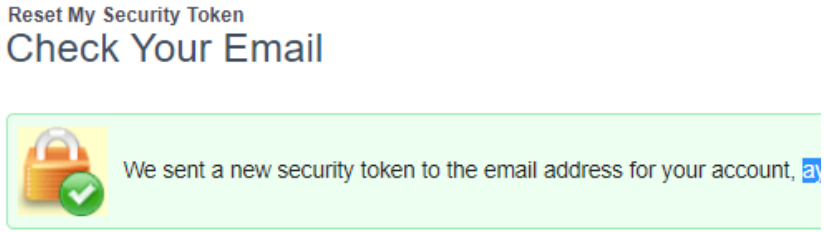
- To reset security token, Select **My Settings** from your account name as illustrated below



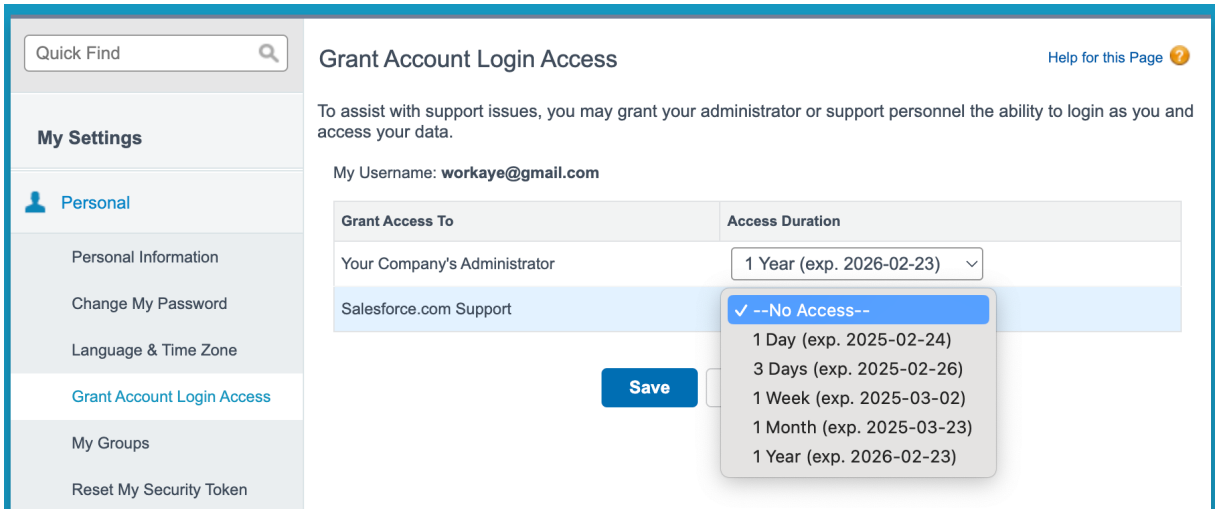
- Under **Personal** you can select **Reset My Security Token**
- Click on **Reset My Security Token**



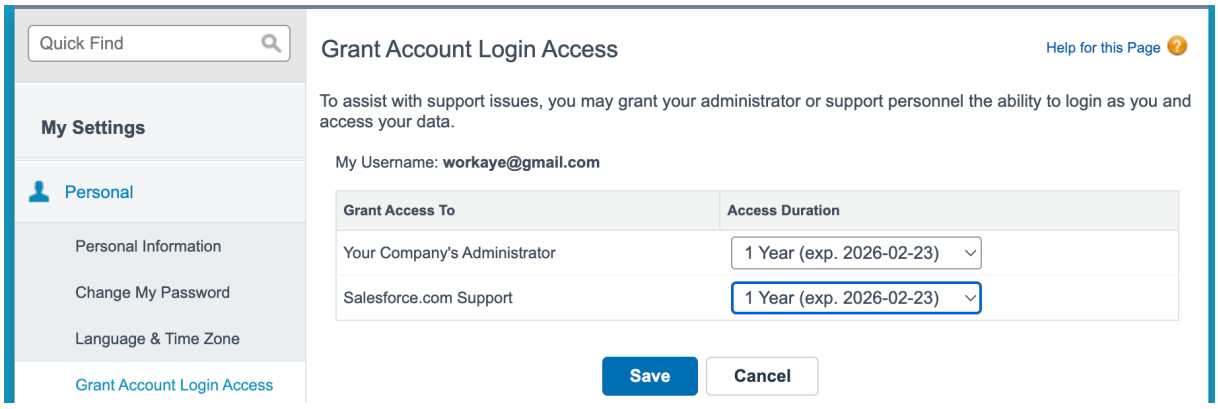
- Your security token will be sent to your email. Note that you need to use this security token in your application to integrate your cloud-based CRM application with applications.



- **Under Grant Account Login Access** <- set Expiry or Access Duration to **one year**



- Click on Save as shown below

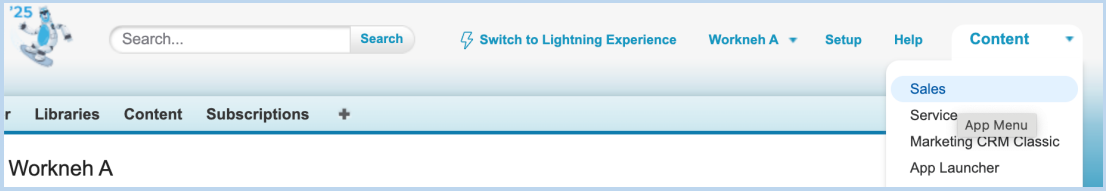


1.9 Adding Products in Salesforce

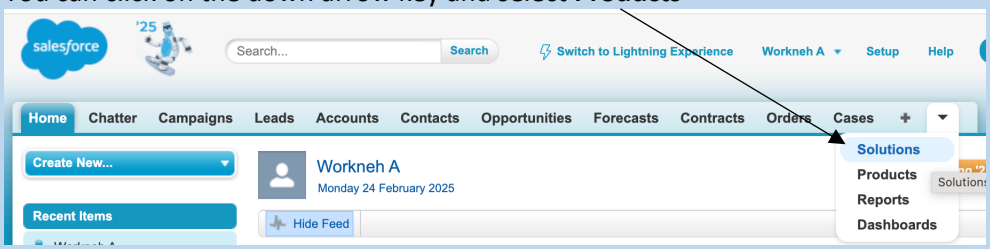
- (If you are not logged in (Login to your Salesforce), enter your username and password if you are not logged in)
- click on **Home** then click on **+** (You may get an error <- you should check later on)

If you encounter an error:

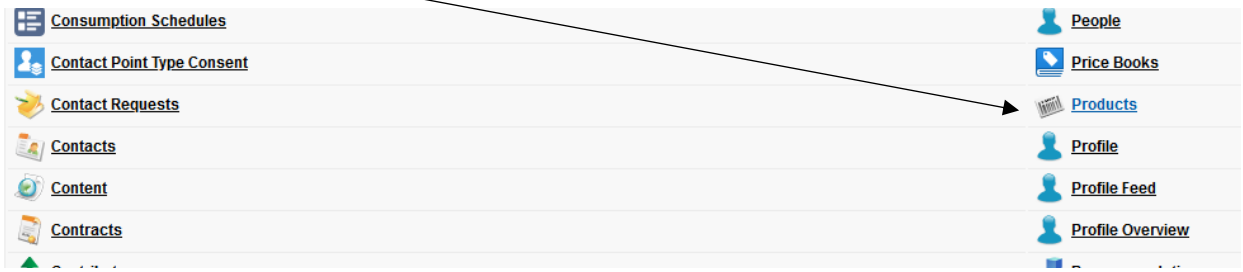
- Select **Sales** instead of **Contents** as illustrated below:



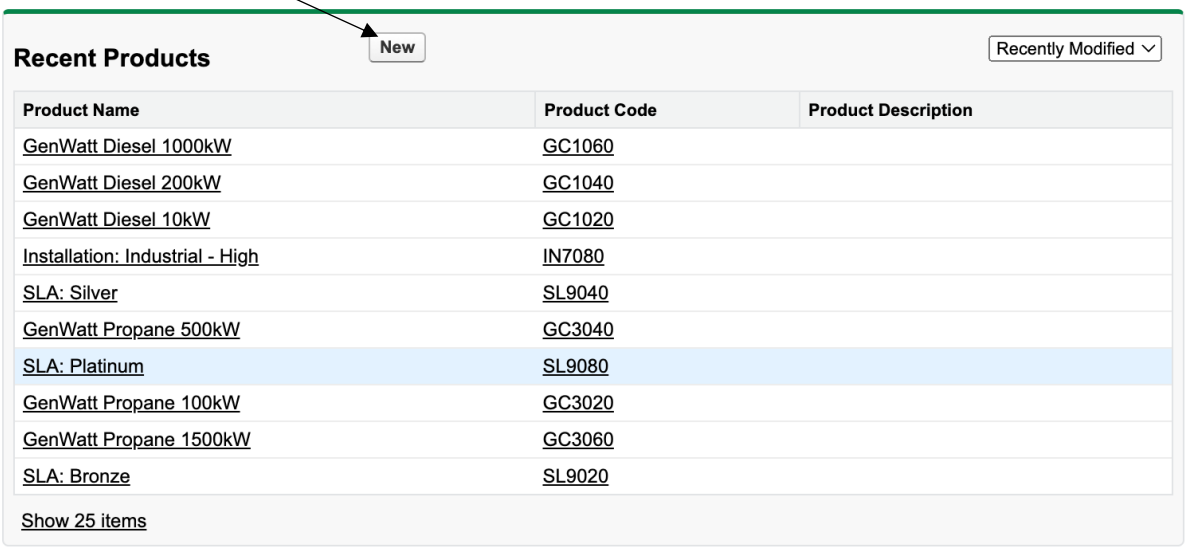
- You can click on the down arrow key and select **Products**



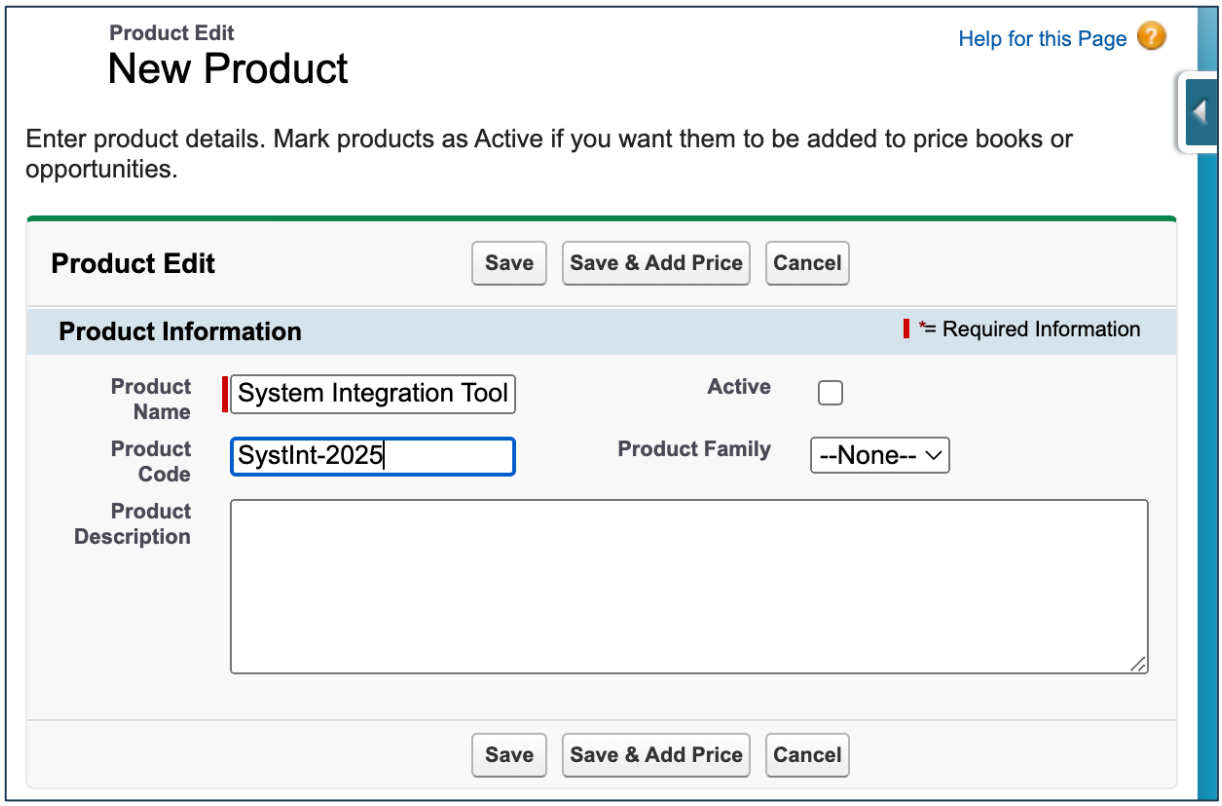
- Select **Products**



- Click on **New** under **Recent Products** (you can scroll down if you do not see this)

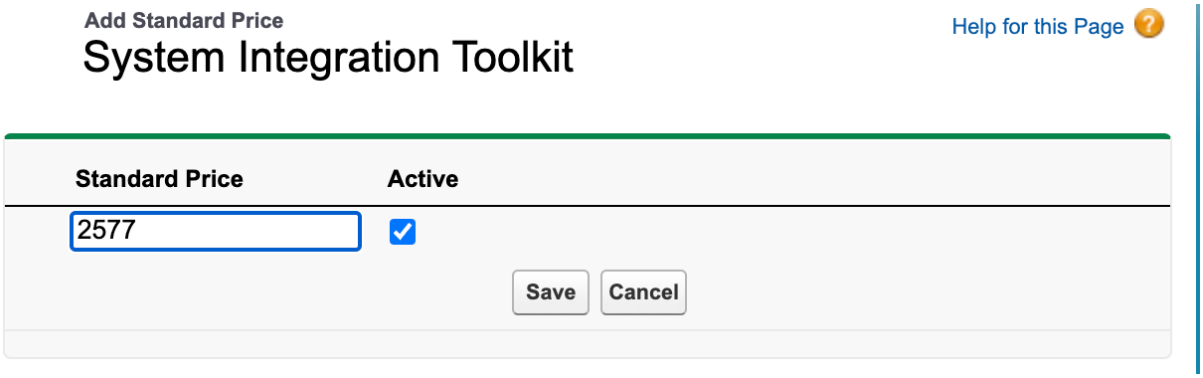


- Under **Product Name** – enter **System Integration Toolkit**
- Under **Product Code** – enter **SystInt-2025**



- Click on **Save**
- Click on **Add** under **Standard Price**

- Enter a price value (2577) as illustrated below, then click on **Save**



TODO: Create three other more products with the same product code (SystInt-2025) as you just did

Product names

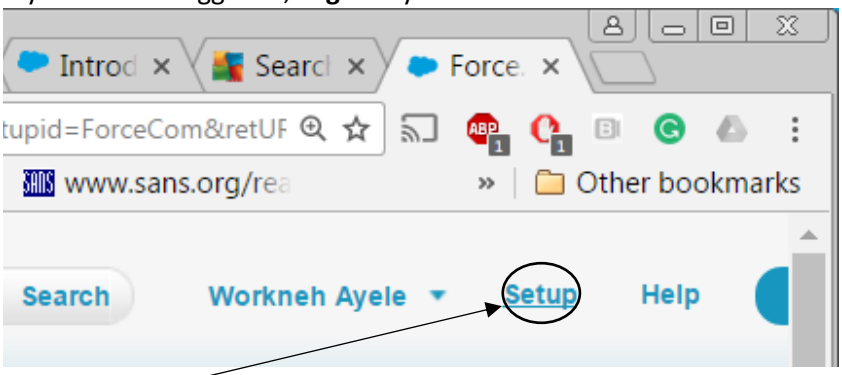
- Visual studio (1500 kr.)
- Intel Server (45000 kr.)
- Router (25000 kr.)
- Swtich (25000 kr.)

- BizTalk Server (258000 kr.)

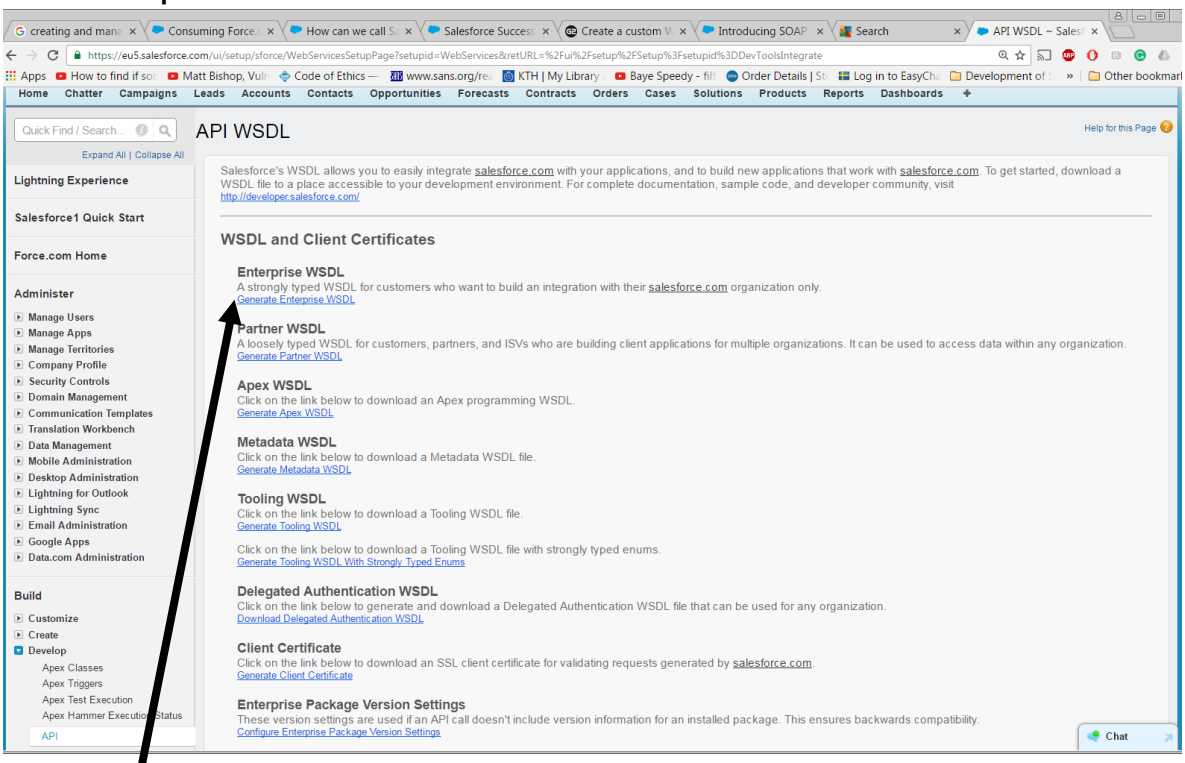
1.10 Get the WSDL SOAP API

Steps

- If you are not Logged in, **Login** to your salesforce.com account



- Select **Setup**
- Create **Task4** folder under **AllFiles** which is found under C:
- Go to **Develop** on the left sidebar under **Build** then select **API**

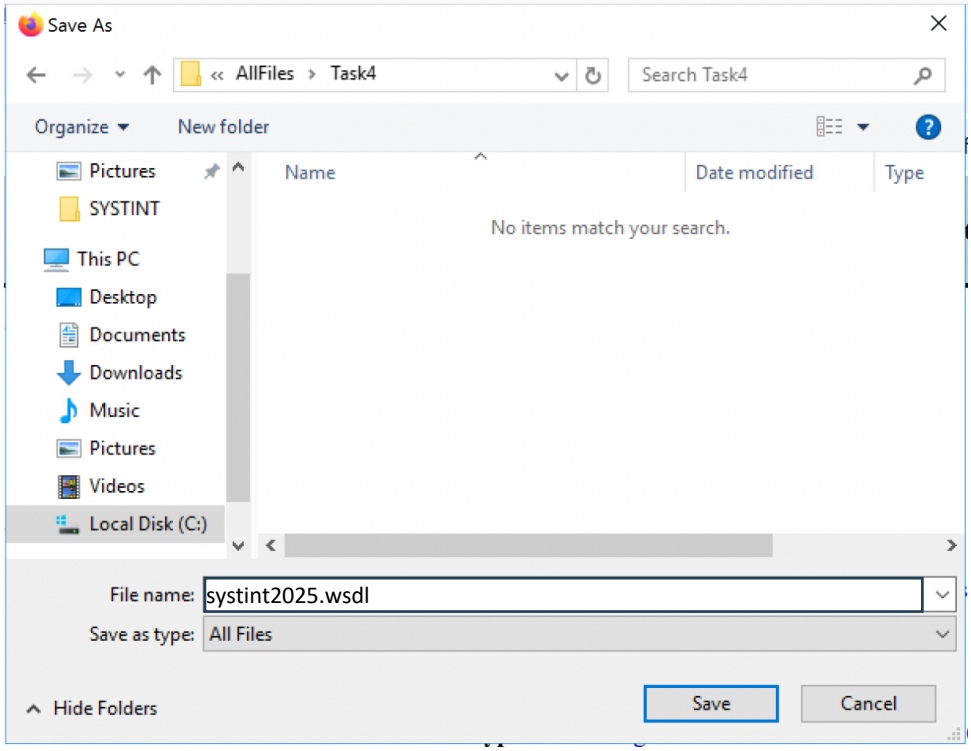


- Select **Generate Enterprise WSDL** (if you get error, re-login in another browser-I prefer Chrome)

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<!--
Salesforce.com Enterprise Web Services API Version 63.0
Generated on 2025-02-24 19:32:42 +0000.

Copyright 1999-2025 salesforce.com, inc.
All Rights Reserved
 -->
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="urn:enterprise.soap.sforce.com"
xmlns:fns="urn:fault.enterprise.soap.sforce.com" xmlns:ens="urn:sobject.enterprise.soap.sforce.com"
targetNamespace="urn:enterprise.soap.sforce.com">
  <types>
    <schema xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
    targetNamespace="urn:sobject.enterprise.soap.sforce.com">
      <import namespace="urn:enterprise.soap.sforce.com"/>
      <!--  Base sObject (abstract)  -->
      <complexType name="sObject">
        <sequence>
          <element name="fieldsToNull" type="xsd:string" nillable="true" minOccurs="0" maxOccurs="unbounded"/>
          <element name="Id" type="tns:ID" nillable="true"/>
        </sequence>
      </complexType>
      <complexType name="AggregateResult">
        <complexContent>
          <extension base="ens:sObject">
            <sequence>
              <any namespace="##targetNamespace" minOccurs="0" maxOccurs="unbounded" processContents="lax"/>
            </sequence>
          </extension>
        </complexContent>
      </complexType>
      <complexType name="AIApplication">
        <complexContent>
          <extension base="ens:sObject">
            <sequence>
              <element name="AiApplications" nillable="true" minOccurs="0" type="tns:QueryResult"/>
              <element name="CreatedBy" nillable="true" minOccurs="0" type="ens:User"/>
              <element name="CreatedById" nillable="true" minOccurs="0" type="tns:ID"/>
              <element name="CreatedDate" nillable="true" minOccurs="0" type="xsd:dateTime"/>
              <element name="DeveloperName" nillable="true" minOccurs="0" type="xsd:string"/>
              <element name="IsDeleted" nillable="true" minOccurs="0" type="xsd:boolean"/>
              <element name="Language" nillable="true" minOccurs="0" type="xsd:string"/>
              <element name="LastModifiedBy" nillable="true" minOccurs="0" type="ens:User"/>
              <element name="LastModifiedById" nillable="true" minOccurs="0" type="tns:ID"/>
              <element name="LastModifiedDate" nillable="true" minOccurs="0" type="xsd:dateTime"/>
              <element name="MasterLabel" nillable="true" minOccurs="0" type="xsd:string"/>
              <element name="MlPredictionDefinitions" nillable="true" minOccurs="0" type="tns:QueryResult"/>
              <element name="MlRecommendationDefinitions" nillable="true" minOccurs="0" type="tns:QueryResult"/>
```

- Right click on the code generated then, select All Files under **Save as type**
- Click **Save as**, to save it in your computer in your project folder - **C:\AllFiles\Task4\systint2025.wsdl**.



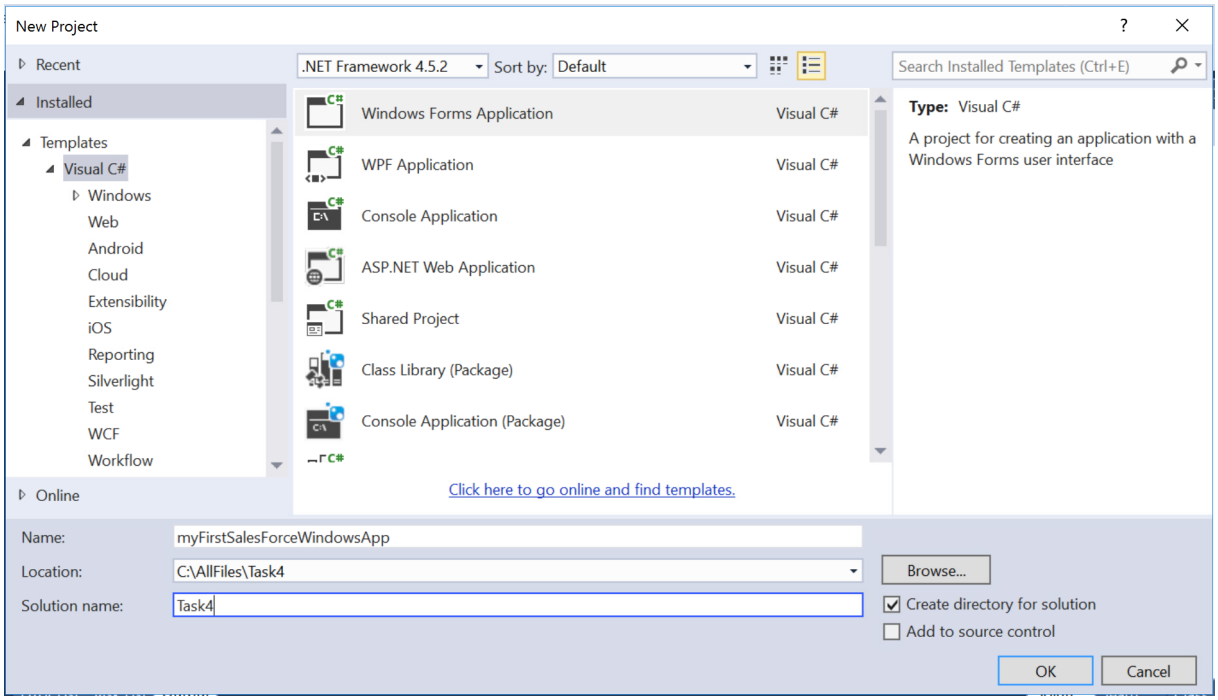
Part II

2. Creating Integrated Windows Application with Salesforce

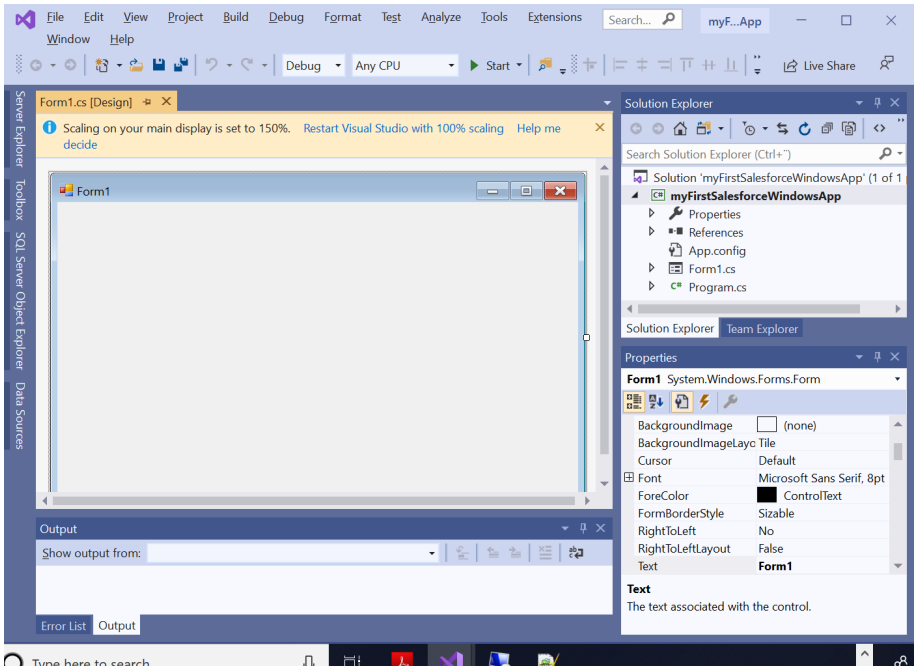
2.1. Creating Integrated Windows .NET Application

Steps

- Start a new **Visual Studio Community** instance by using the **Start** menu, Select **New Project**
- Select **Windows Forms Application** (.NET Framework) after selecting **Visual C#**

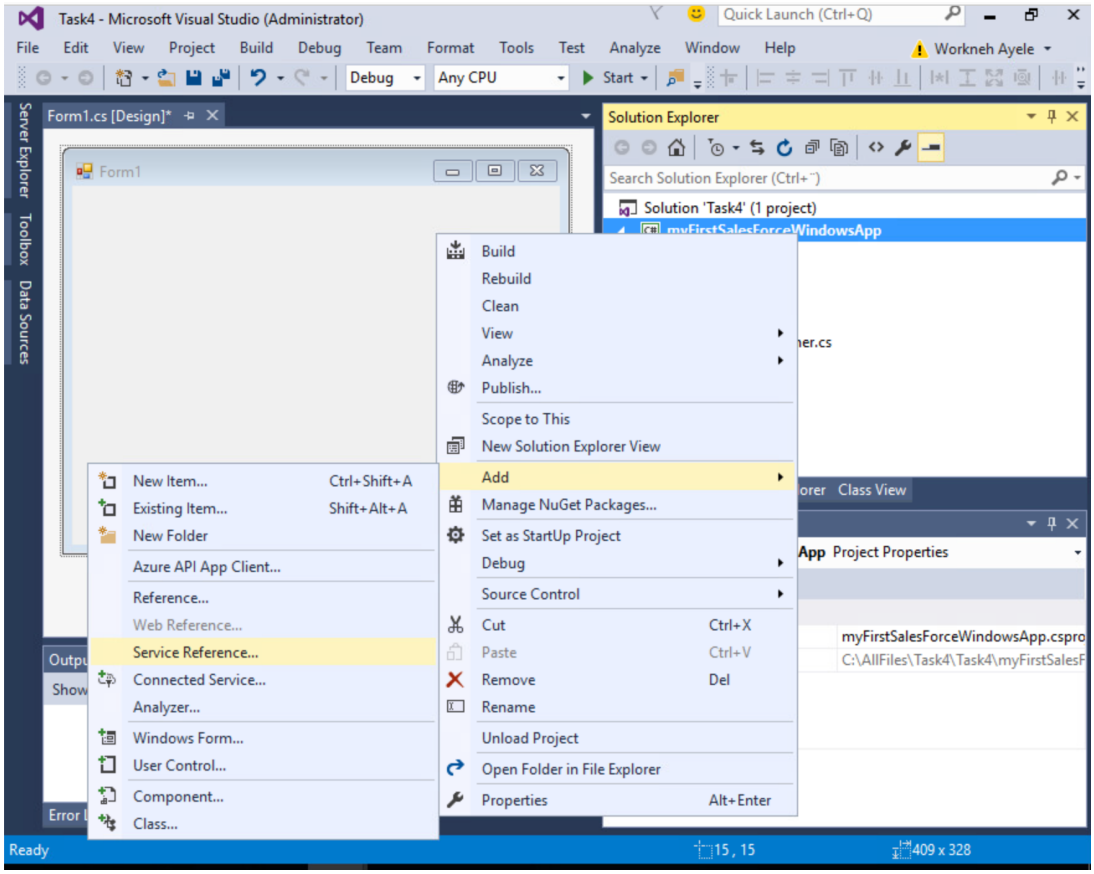


- Enter the applications name – **myFirstSalesforceWindowsApp**
- Specify **Location** to be – C:\AllFiles\Task4
- Click on **OK**, and your Windows Form will be created, as illustrated below



To create the reference to the service

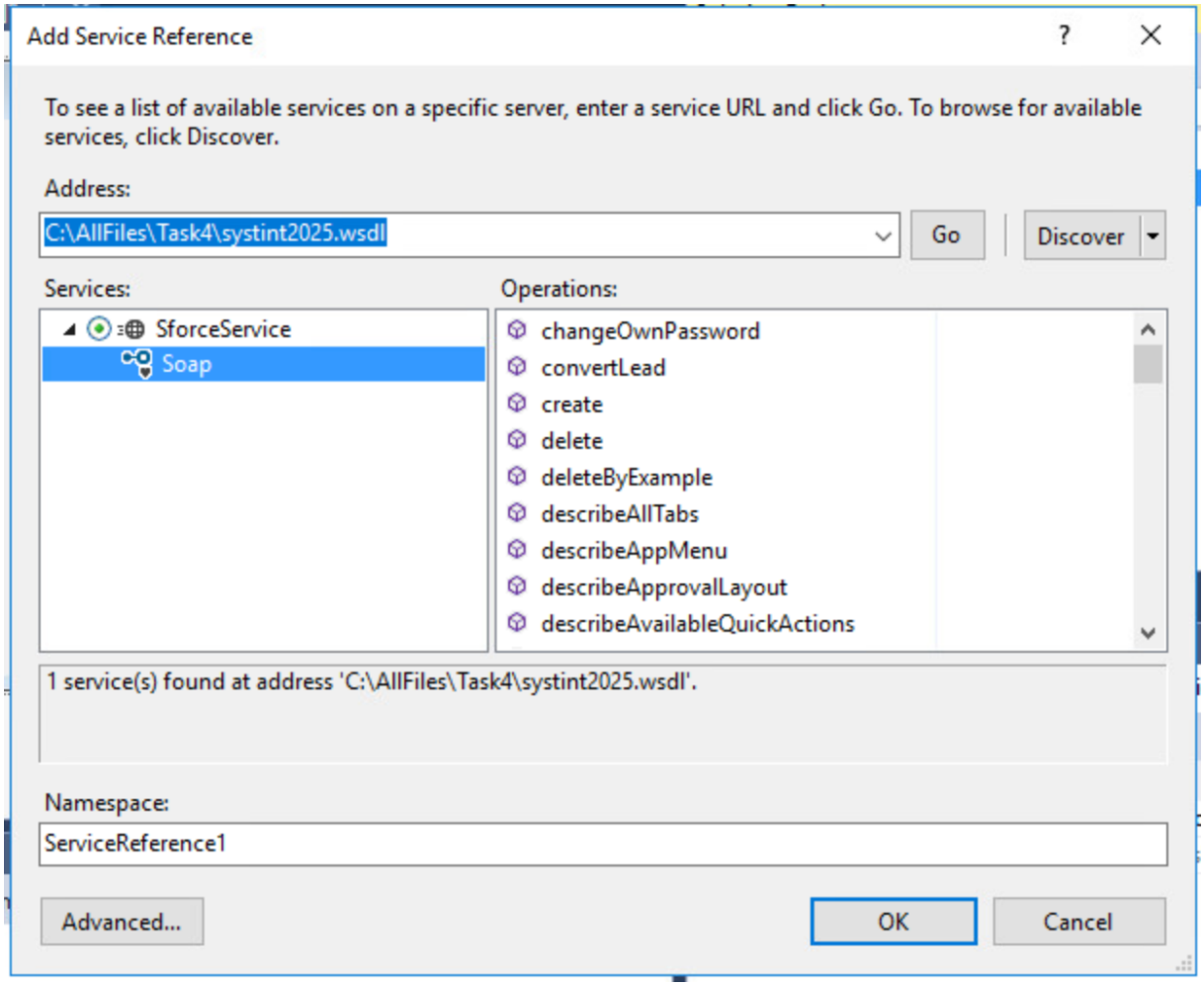
- Right-click on your **Project** > Point to **Add** > Select **Services References**



- Add the address of the WSDL file, i.e., systint2025.wsdl
  - The address is: **"C:\AllFiles\Task4\systint2025.wsdl"**
- Then click **Go**

- Enter **ServiceReference1** as the name, and for illustration, see the snapshot below

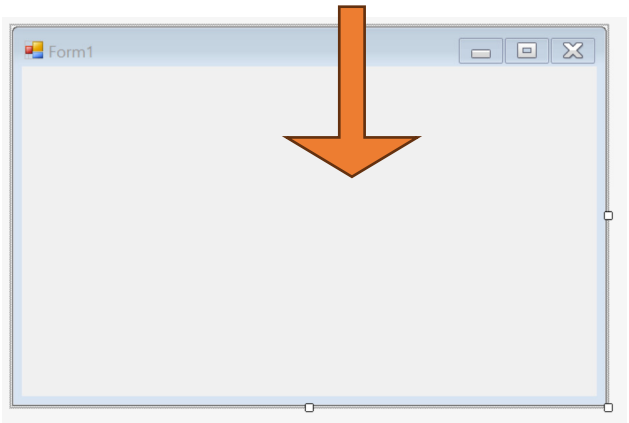
Note: You can see the functions available in the service.



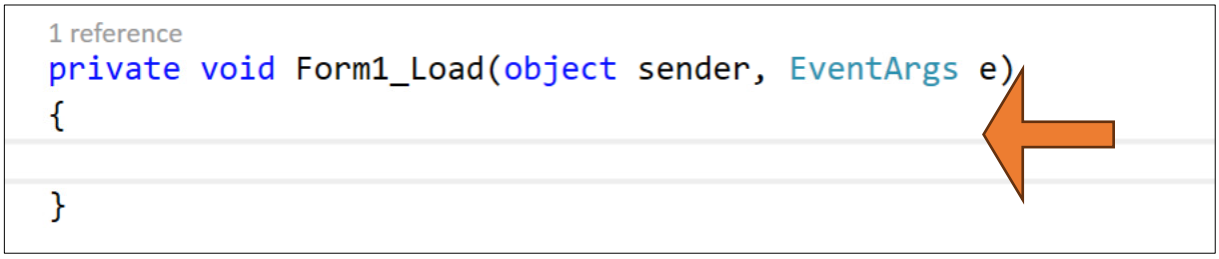
- Click **OK**

2.2. Adding the Form1 load code

Here, you will add the Form_load code. As this application is opened, it will start a new session by connecting to Salesforce using your username, password, and security token. To create this even-based method/function, **double-click** on the empty background of Form1.



- Now, the form load event will be opened in the code view. Enter the code for Form1_Load as illustrated below:

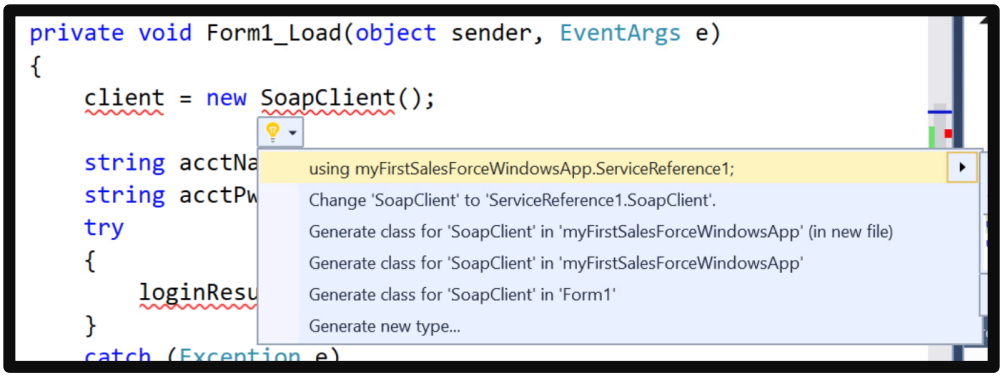


The code for this method/function:

```
private void Form1_Load(object sender, EventArgs e)
    {
                client = new SoapClient();

               ServicePointManager.Expect100Continue = true;
               ServicePointManager.SecurityProtocol = SecurityProtocolType.Tls12;

               string acctName = "youraccountemail@xxx.com"; //"YOUR DEVORG USERNAME";
               string acctPw = "YourPassword"+"Yoursecuritytoken";
               try
                {
                    loginResult = client.login(null, acctName, acctPw);
                }
                catch (Exception ex)
               {
                        MessageBox.Show(ex.Message+ " and "+ ex.StackTrace);

                    if (ex.Data != null)
                        {
                           MessageBox.Show("You failed to login");
                           Close();
                        }
                }

                    MessageBox.Show("Congratulations: Login succeeded ");
                    // success
    }
```

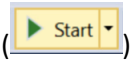
After you enter the code you will get some errors, to fix the errors follow the following step:

- Hover your mouse over **SoapClinet** as illustrated below:

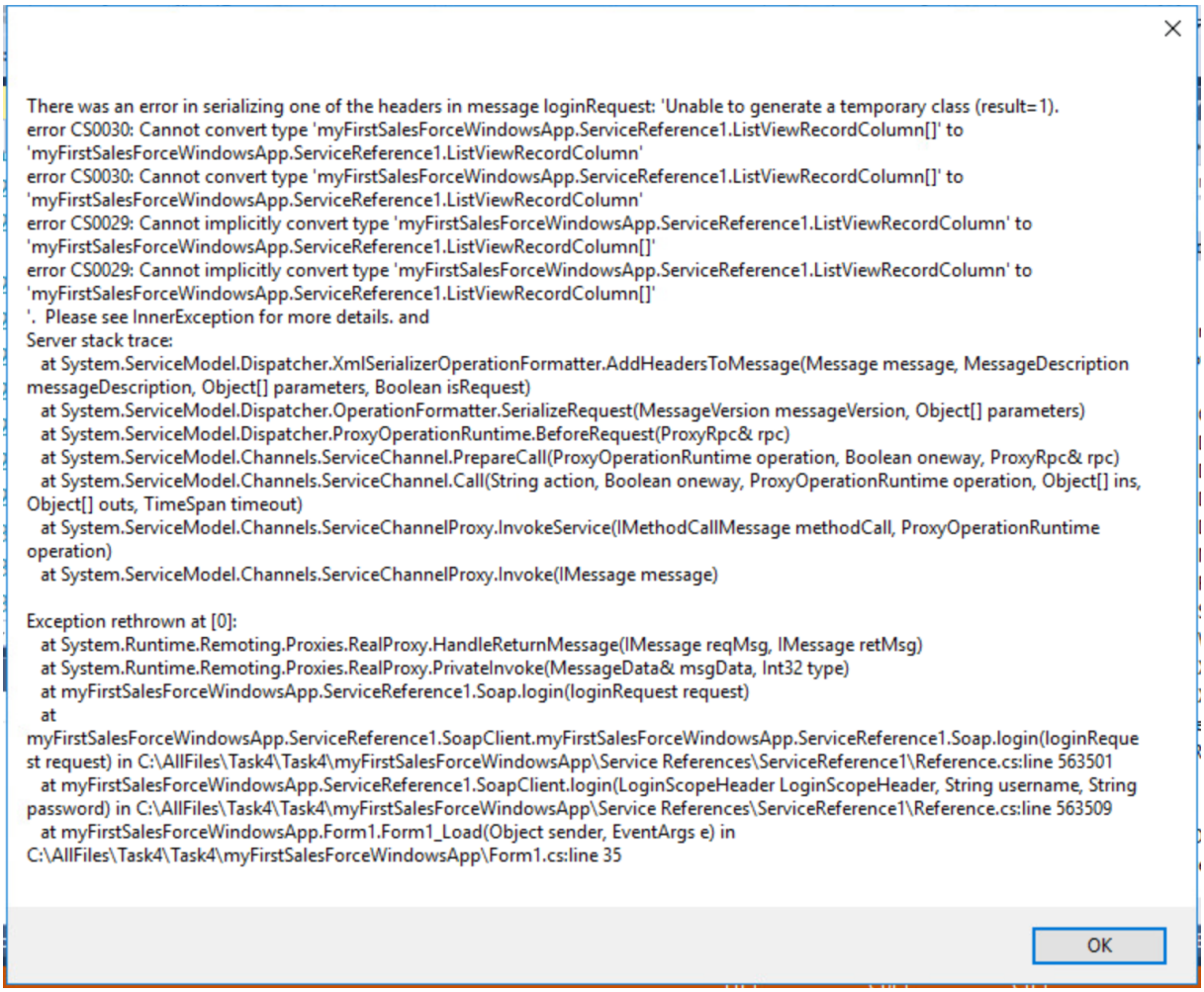


- You will still get error messages to clean the errors, add the following code above private void Form1_Load (object sender, EventArgs e) as illustrated below:

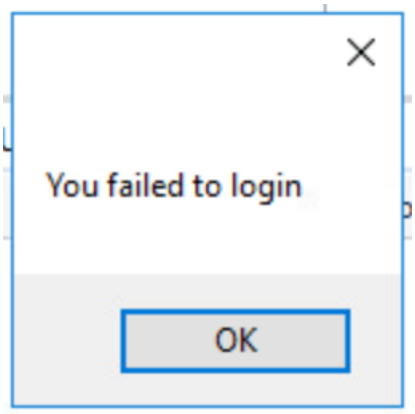
```
private static SoapClient client;
private static LoginResult loginResult;
1 reference
private void Form1_Load(object sender, EventArgs e)
{
    client = new SoapClient();
```

- Add "using System.Net" without the quotation marks in the beginning of the entire code.
- Build the project and click on **Start** (

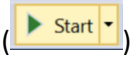
)



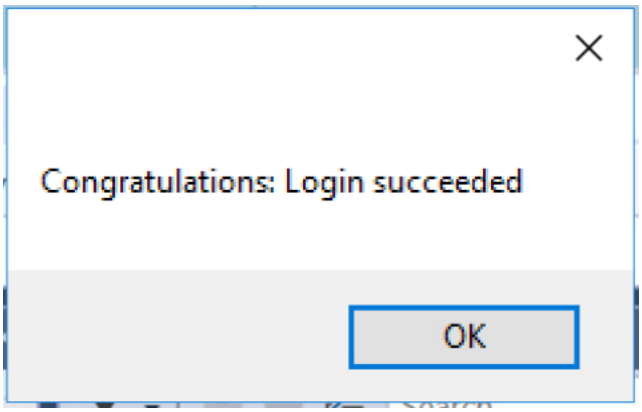
- Click OK



- Click OK again

- The error is a common error, and you can easily fix it by following the instruction in Appendix 1.
- After fixing the issue, Run your Windows Application by clicking on **Start** (

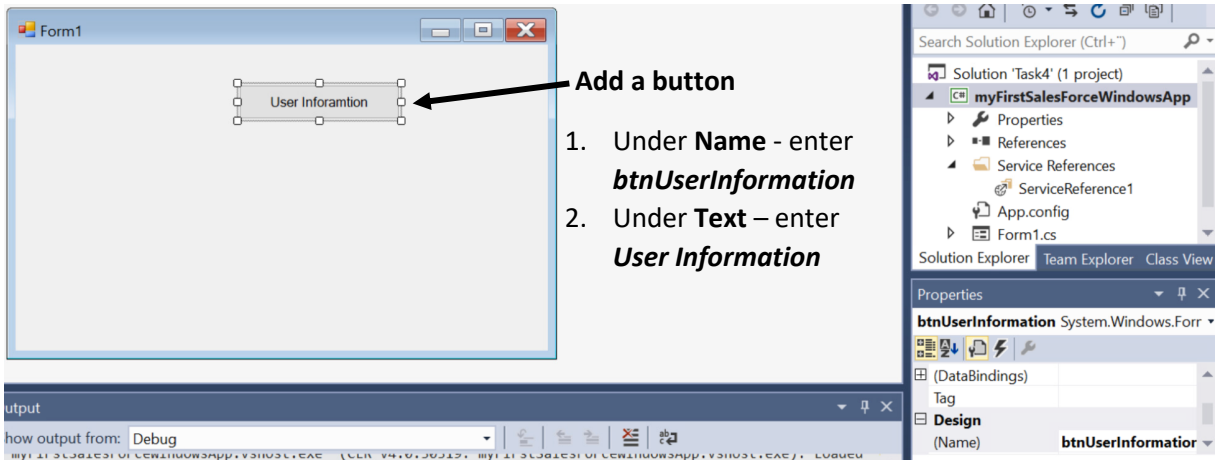
)
- If everything works fine! You will see a message box showing you the following message "**Congratulations: Login succeeded**"



- Exit the running application to continue to the next step

2.3. Adding controls to Form1 designer

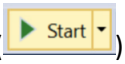
Here, you will add a button; when the user clicks on the button, it will display a message box with information about the user details of your Salesforce account.



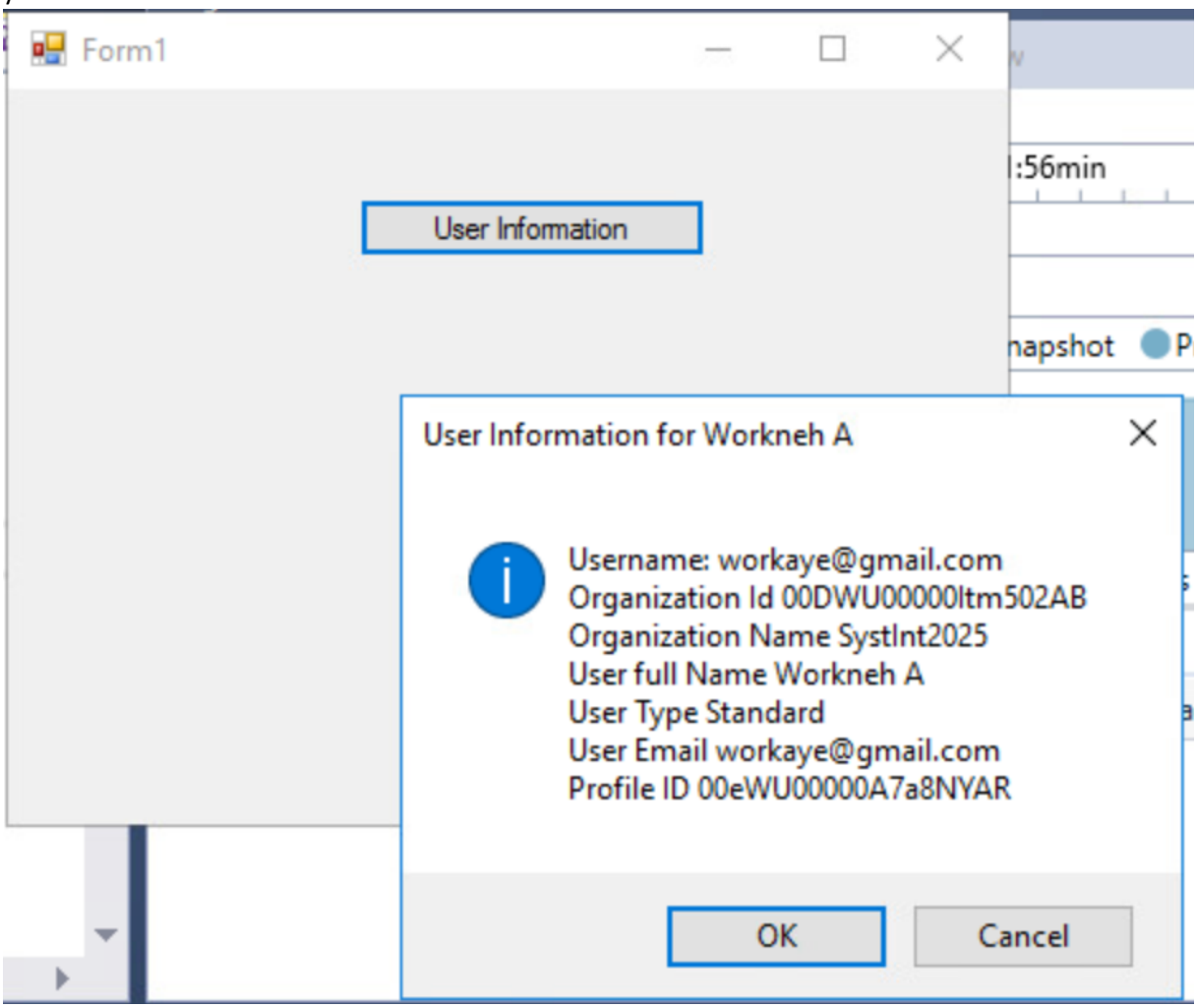
- Double click on the button: **User Information**
- Add the following code:

```
private void btnUserInformation_Click(object sender, EventArgs e)
{

    MessageBox.Show("Username: " + loginResult.userInfo.userName + "\n"
        + "Organization Id " + loginResult.userInfo.organizationId + "\n"
        + "Organization Name " + loginResult.userInfo.organizationName + "\n"
        + "User full Name " + loginResult.userInfo.userFullName + "\n"
        + "User Type " + loginResult.userInfo.userType + "\n"
        + "User Email " + loginResult.userInfo.userEmail + "\n"
        + "Profile ID " + loginResult.userInfo.profileId + "\n"
        , "User Information for"+ loginResult.userInfo.userFullName,
        MessageBoxButtons.OKCancel, MessageBoxIcon.Information);
}
```

Save your project

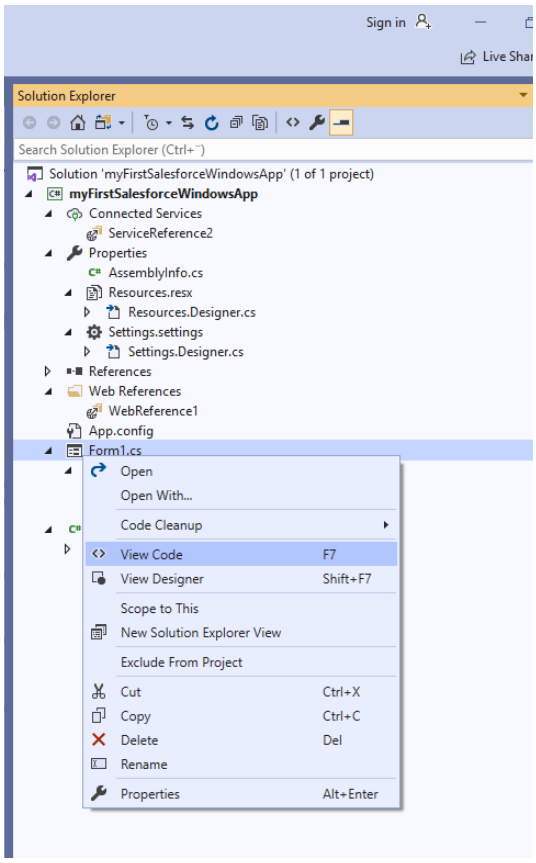
- Run your Windows Application by clicking on **Start** (

)
- Click OK on the Login success dialog box and click on User Information button, if you succeed you will see all user details as illustrated below.



2.4. Switching between Designer View and Code View

- To go back to Design View:
  - Under **Form1** in the **Solution Explorer** right click on **Form1.cs** > then click on **View Designer**
- To go back to Design View:
  - Under **Form1** in the **Solution Explorer** right click on **Form1.cs** > then click on **View Designer**



Exercise (mandatory): Add a button (**Text** = Server URL, **Name** = btnServerURL) that will be used to show ServerURL, and write the code as you did for User Information.

Exercise (optional): Add a button (**Text** = Session ID, **Name** = btnSessionID) and write the code as you did for User Information.

The final code:

- The code is illustrated below, Form1.cs – you can copy the code below or enter the code manually

```
using myFirstSalesForceWindowsApp.ServiceReference1;
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;

namespace myFirstSalesForceWindowsApp
{
   public partial class Form1 : Form
```

```
    {
        public Form1()
        {
            InitializeComponent();
        }
        private static SoapClient client;
        private static LoginResult loginResult;
        private void Form1_Load(object sender, EventArgs e)
        {
            client = new SoapClient();

            ServicePointManager.Expect100Continue = true;
            ServicePointManager.SecurityProtocol = SecurityProtocolType.Tls12;

            string acctName = "email@xyz.com"; //your username
            string acctPw = "yourPassword" + "your_token";

            try
            {
                loginResult = client.login(null, acctName, acctPw);
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message+" and "+ex.StackTrace);
                if(ex.Data !=null)
                {
                    MessageBox.Show("You failed to login");
                    Close();
                }
            }
            MessageBox.Show("Congratulations: Login succeeded"); //Success
        }

        private void btnUserInformation_Click(object sender, EventArgs e)
        {
            MessageBox.Show("Username: " + loginResult.userInfo.userName + "\n"
                + "Organization Id " + loginResult.userInfo.organizationId + "\n"
                + "Organization Name " + loginResult.userInfo.organizationName + "\n"
                + "User full Name " + loginResult.userInfo.userFullName + "\n"
                + "User Type " + loginResult.userInfo.userType + "\n"
                + "User Email " + loginResult.userInfo.userEmail+ "\n"
                + "Profile ID " + loginResult.userInfo.profileId+ "\n"
                , "User Information for " + loginResult.userInfo.userFullName, MessageBoxButtons.OKCancel,
MessageBoxIcon.Information);
        }
    }
}
```

The additional codes (the optional and the mandatory:

```
        private void btnServerURL_Click(object sender, EventArgs e)
        {
            MessageBox.Show("User name : " + loginResult.userInfo.userName + "\n"
                + "Server URL :" + loginResult.serverUrl,
                "Server URL for" + loginResult.userInfo.userFullName + "\n",
                MessageBoxButtons.OKCancel, MessageBoxIcon.Information);
        }

        private void btnSessionID_Click(object sender, EventArgs e)
```

```
    {
        MessageBox.Show("User name : " + loginResult.userInfo.userName + "\n"
            + "Session ID :" + loginResult.sessionId,
            "Session Id for" + loginResult.userInfo.userFullName + "\n",
            MessageBoxButtons.OKCancel, MessageBoxIcon.Information);
    }
```

2.5. Testing if the code works

- Select **Start** from the toolbox

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 whate you see on the screeshot

- Find **Reference.cs** file in your project folder (Open it using Notepad++) then find all instances of '[][]' and replace them with '[]'