

## **Exercise Solution: Create a WCF Based Extension**

Service Layer

PUBLIC

## INTRODUCTION

In this exercise, you will perform the following tasks:

1. Configure the Service Layer service
2. Install WCF Data Services
3. Create project in Visual Studio
4. Connect to Service Layer
5. Disconnect from Service Layer

## PREREQUISITE:

- This document is using the **C Sharp** (C#) language
- This document is using the Microsoft Visual Studio 2015
- Use the demo database for SAP Business One, version for SAP HANA
- Credentials: User code: **manager**

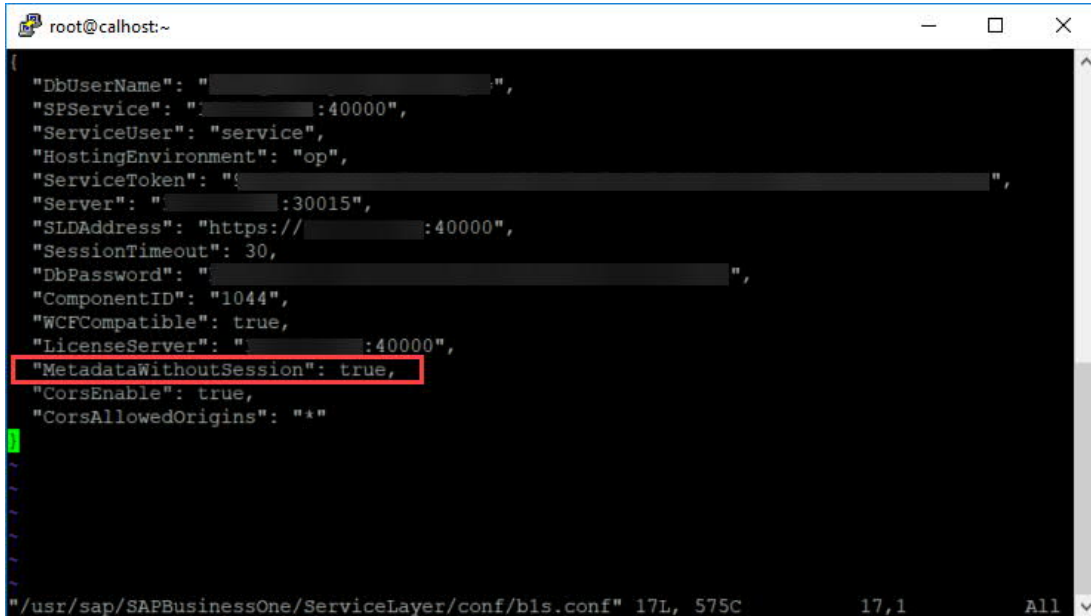
## GUIDELINES:

The screenshots provided here are for your reference only and may differ from the actual screenshots in your system.

## 1. TASK – CONFIGURE THE SERVICE LAYER SERVICE

### 1.1. Set the configuration option **MetadataWithoutSession** to true.

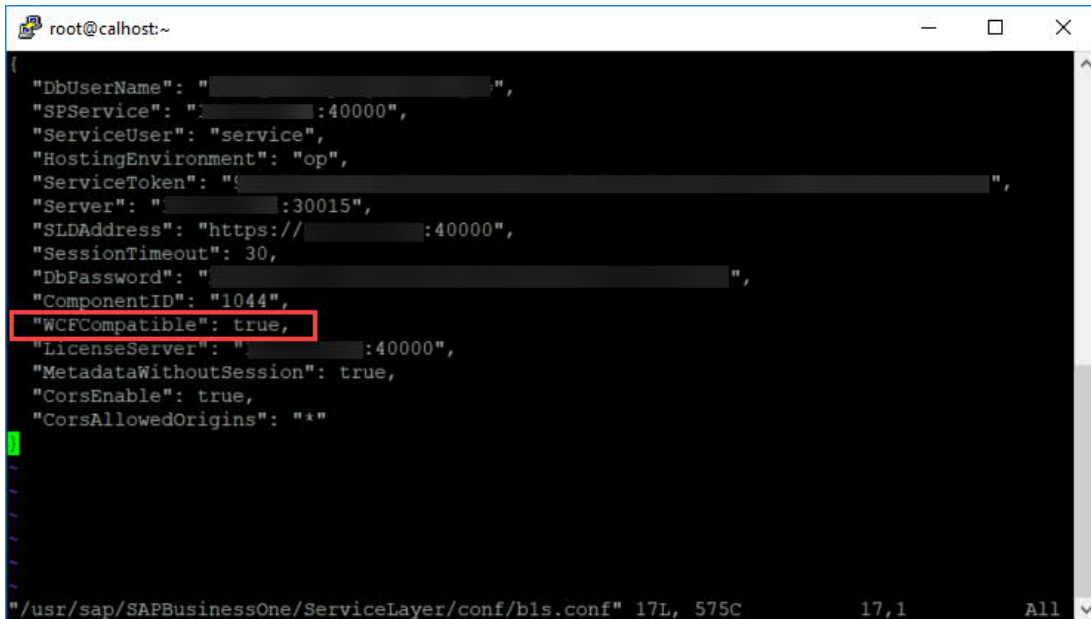
Use the **vi** command for editing the file **b1s.conf**, located on *<Installation Directory>/ServiceLayer/conf/b1s.conf*



```
root@calhost:~  
{  
  "DbUserName": " ",  
  "SPService": " :40000",  
  "ServiceUser": "service",  
  "HostingEnvironment": "op",  
  "ServiceToken": " ",  
  "Server": " :30015",  
  "SLDAddress": "https:// :40000",  
  "SessionTimeout": 30,  
  "DbPassword": " ",  
  "ComponentID": "1044",  
  "WCFCompatible": true,  
  "LicenseServer": " :40000",  
  "MetadataWithoutSession": true,  
  "CorsEnable": true,  
  "CorsAllowedOrigins": "*" }  
"/usr/sap/SAPBusinessOne/ServiceLayer/conf/b1s.conf" 17L, 575C 17,1 All
```

### 1.2. Set the configuration option **WCFCompatible** to true.

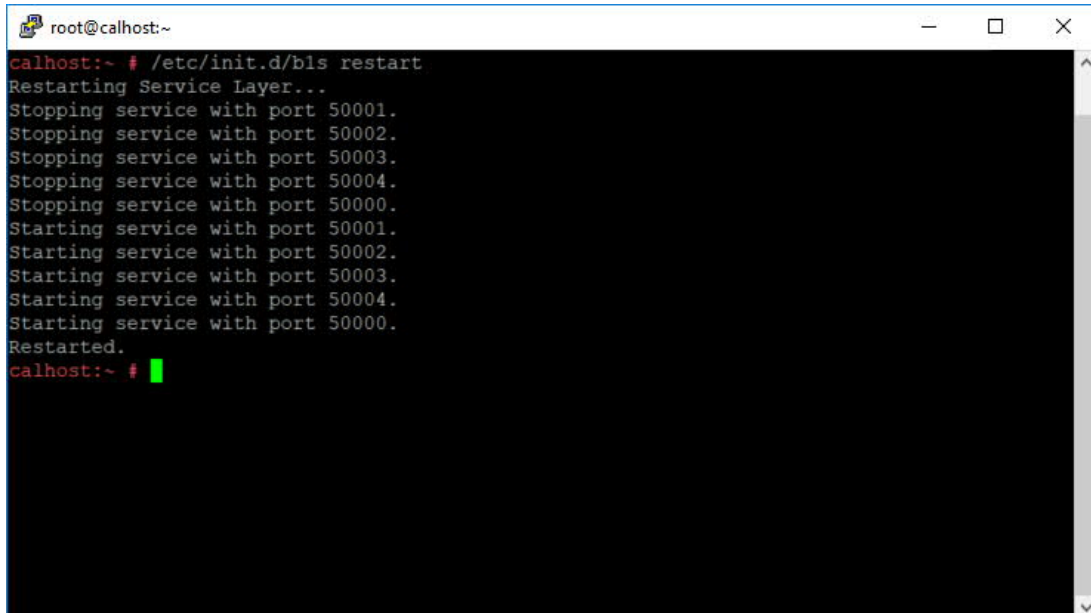
Use the **vi** command for editing the file **b1s.conf**, located on *<Installation Directory>/ServiceLayer/conf/b1s.conf*



```
root@calhost:~  
{  
  "DbUserName": " ",  
  "SPService": " :40000",  
  "ServiceUser": "service",  
  "HostingEnvironment": "op",  
  "ServiceToken": " ",  
  "Server": " :30015",  
  "SLDAddress": "https:// :40000",  
  "SessionTimeout": 30,  
  "DbPassword": " ",  
  "ComponentID": "1044",  
  "WCFCompatible": true,  
  "LicenseServer": " :40000",  
  "MetadataWithoutSession": true,  
  "CorsEnable": true,  
  "CorsAllowedOrigins": "*" }  
"/usr/sap/SAPBusinessOne/ServiceLayer/conf/b1s.conf" 17L, 575C 17,1 All
```

### 1.3. Restart the Service Layer service

Use the `/etc/init.d/b1s restart` to restart the service.

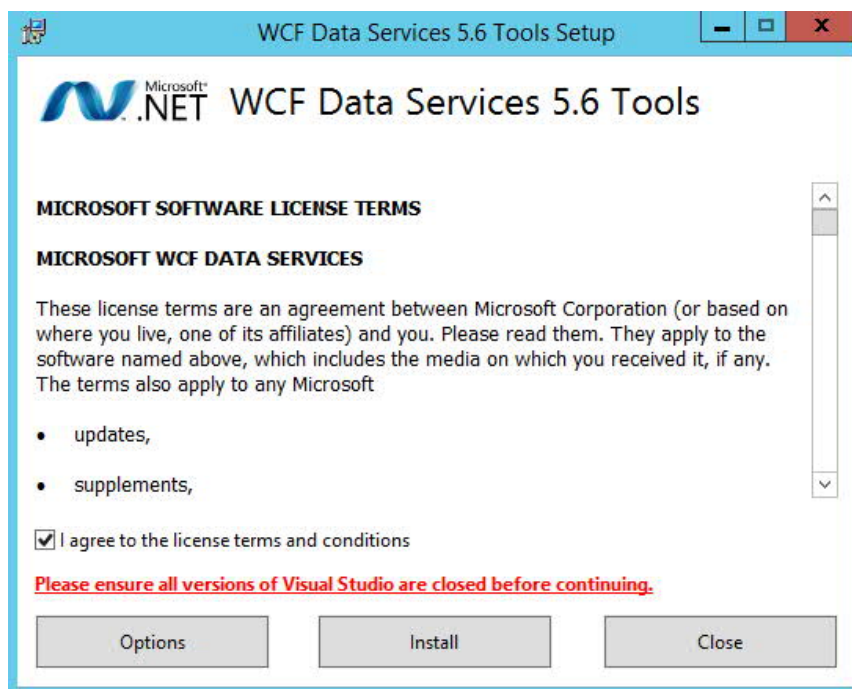


```
root@calhost:~  
calhost:~ # /etc/init.d/b1s restart  
Restarting Service Layer...  
Stopping service with port 50001.  
Stopping service with port 50002.  
Stopping service with port 50003.  
Stopping service with port 50004.  
Stopping service with port 50000.  
Starting service with port 50001.  
Starting service with port 50002.  
Starting service with port 50003.  
Starting service with port 50004.  
Starting service with port 50000.  
Restarted.  
calhost:~ #
```

## 2. TASK - INSTALL WCF DATA SERVICES

2.1. Please make sure Visual Studio is closed before running the installer.

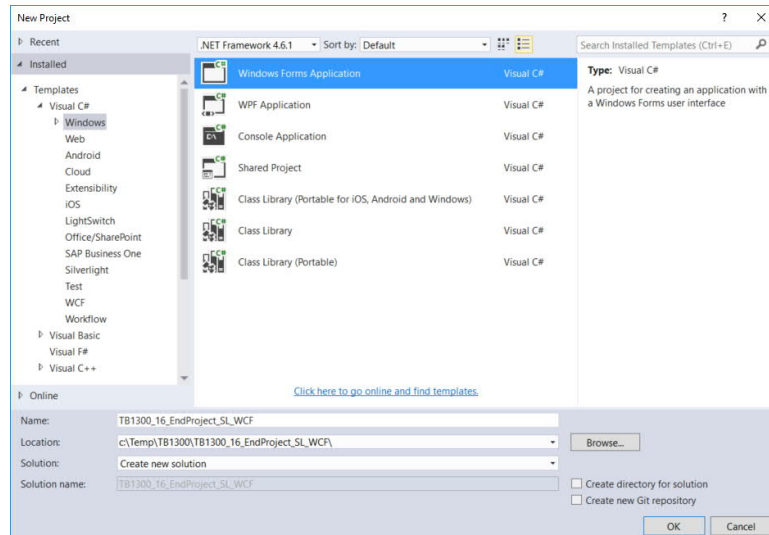
2.2. Download and run the installer to install the [WCF Data Services 5.6.0 tools](#) on your computer.



### 3. TASK – CREATE PROJECT IN VISUAL STUDIO

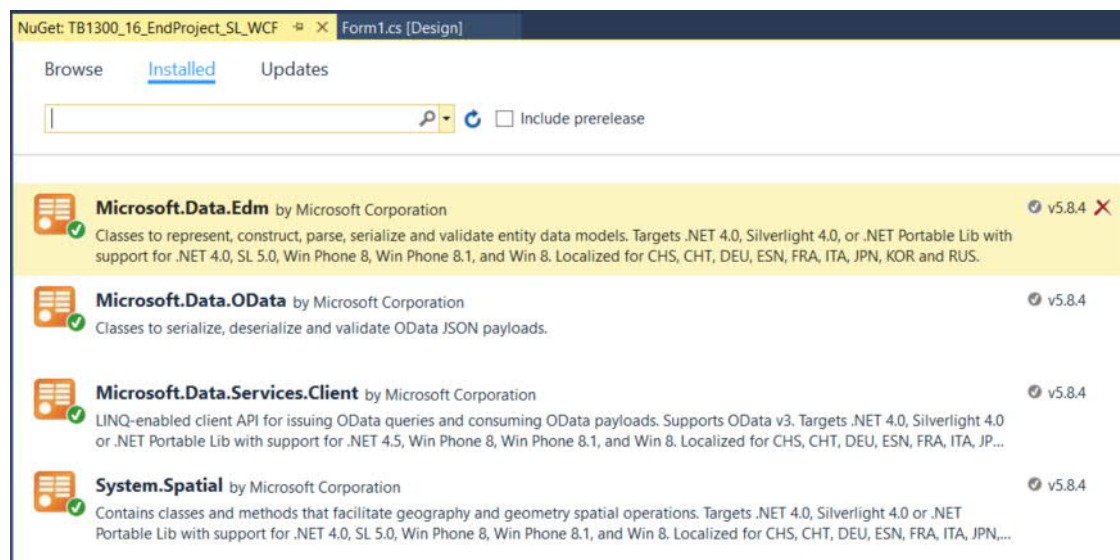
3.1. Open Visual Studio and choose *File* → *New* → *Project*

3.2. Select the **Windows Forms Application** template from *Installed* → *Templates* → *Windows* section.

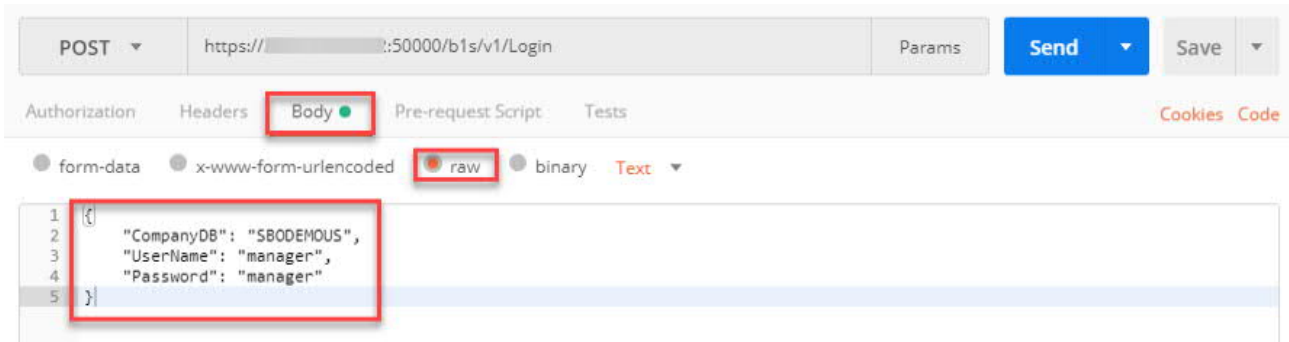


3.3. Install NuGet packages - Microsoft.Data.Edm, Microsoft.Data.Odata, Microsoft.Data.Services.Client and System.Spatial

*Project* → *Manage NuGet Packages*

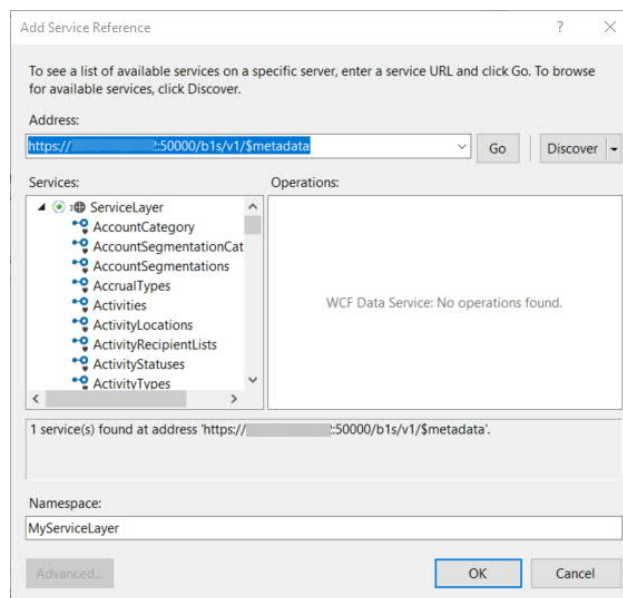


3.4. Connect to the Service Layer using a third-party tool, like the Postman.

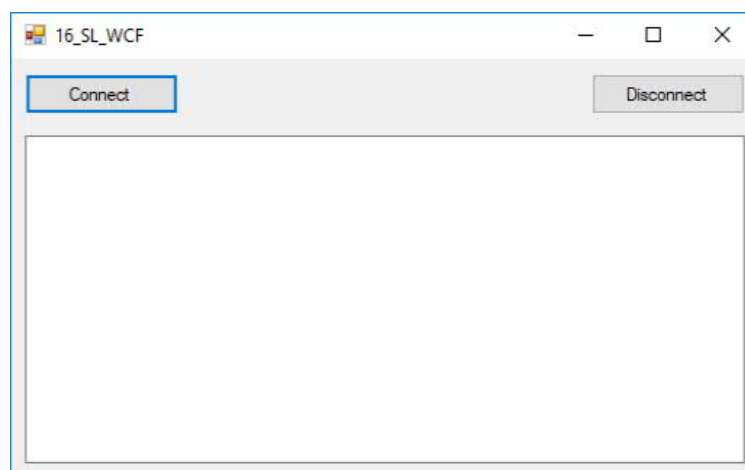


3.5. Add the metadata as service or file directly by selecting “Add Service Reference” into your Microsoft Visual Studio project, the metadata will be validated automatically.

*Project → Add Service Reference*



3.6. Create two buttons on the default form the first one should be used for handling the Connect and the second one should handle the Disconnect event. Add a *Textbox* item for displaying the results.



#### 4. TASK - CONNECT TO SERVICE LAYER

- 4.1. Create the variable **slContext** for ServiceLayer object – ensure it is defined as a member of the extension application class or globally.

```
public static MyServiceLayer.SAPB1.ServiceLayer slContext = null;
```

- 4.2. Create the variable **CookieString** for Cookies – ensure it is defined as a member of the extension application class or globally.

```
private string CookieString;
```

- 4.3. Create the variable **ServiceLayerSession** for Service Layer Session – ensure it is defined as a member of the extension application class or globally.

```
public MyServiceLayer.SAPB1.B1Session ServiceLayerSession = null;
```

- 4.4. Create the function **SLSendingRequest** for specifying the web service call request details.

```
void SLSendingRequest(object sender, System.Data.Services.Client.SendingRequestEventArgs e)
{
    HttpRequest request = (HttpRequest)e.Request;
    if (null != request)
    {
        request.Accept = "application/json; odata=minimal metadata";
        request.KeepAlive = true; //keep alive
        request.ServicePoint.Expect100Continue = false; //content
        request.AllowAutoRedirect = true;
        request.ContentType = "application/json; odata=minimal metadata; charset=utf8";
        request.Timeout = 100000; //100 seconds.
        if (!string.IsNullOrEmpty(CookieString))
        {
            e.RequestHeaders.Add("Cookie", CookieString);
        }
    }
    else
    {
        throw new Exception("Failed to intercept the sending request");
    }
}
```

- 4.5. Create the function **SLReceivingResponse** to work with the web service response.

```
void SLReceivingResponse(object sender, ReceivingResponseEventArgs e)
{
    if (null == e.ResponseMessage)
        return;

    string strMessage = e.ResponseMessage.GetHeader("Set-Cookie");

    if (!string.IsNullOrEmpty(strMessage))
    {
        CookieString = strMessage.Replace(',', ';');
    }
}
```



4.6. Create the function **RemoteSSLTLSCertificateValidate** for certificate validation.

```
private static bool RemoteSSLTLSCertificateValidate(object sender, X509Certificate
cert, X509Chain chain, SslPolicyErrors ssl)
{
    return true;
}
```

4.7. Write the code for connecting to the Service Layer, which should be executed after pressing the **Connect** button.

```
private void btnConnect_Click(object sender, EventArgs e)
{
    slContext = new MyServiceLayer.SAPB1.ServiceLayer(new
Uri ("https://YourHanaServerAddress:50000/b1s/v1/"));
    slContext.Format.UseJson();

    slContext.SendingRequest += SLSendingRequest;
    slContext.ReceivingResponse += SLReceivingResponse;
    slContext.MergeOption = MergeOption.OverwriteChanges;
    ServicePointManager.ServerCertificateValidationCallback +=
RemoteSSLTLSCertificateValidate;

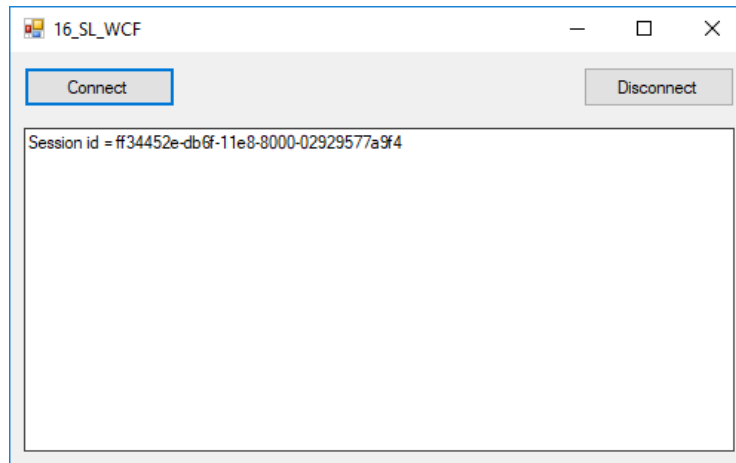
    try
    {
        Uri login = new Uri ("/Login", UriKind.Relative);
        BodyOperationParameter[] body = new BodyOperationParameter[3];
        body[0] = new BodyOperationParameter("UserName", "manager");
        body[1] = new BodyOperationParameter("Password", "manager");
        body[2] = new BodyOperationParameter("CompanyDB", "SBODEMOUS");
        if (ServiceLayerSession == null)
            ServiceLayerSession =
(MyServiceLayer.SAPB1.B1Session)slContext.Execute<MyServiceLayer.SAPB1.B1Session>(log
in, "POST", true, body).SingleOrDefault();
    }
    catch (Exception ex)
    {
        txtMain.AppendText(ex + System.Environment.NewLine);
        return;
    }
}
```



4.8. To Test the function, connect to the Service Layer and display the SessionID into the TextBox.

The following line can be added to the **btnConnect\_Click** function.

```
txtMain.AppendText("Session id = " + ServiceLayerSession.SessionId +  
System.Environment.NewLine);
```

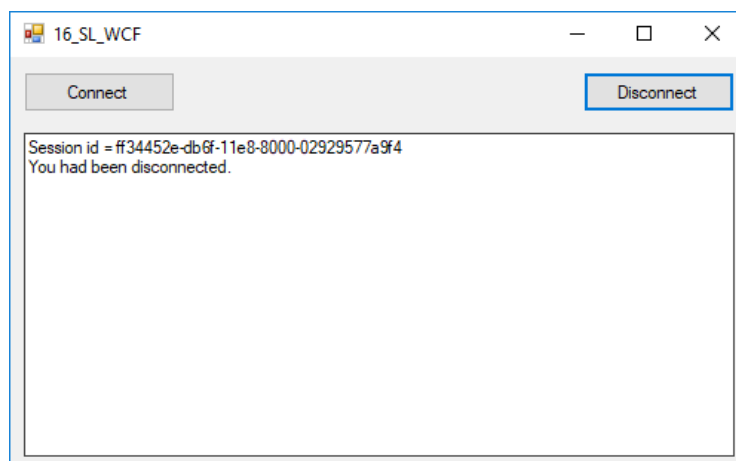


## 5. TASK - DISCONNECT FROM SERVICE LAYER

5.1. Write the code for disconnecting from the Service Layer, which should be executed after pressing the **Disconnect** button.

```
private void btnDisconnect_Click(object sender, EventArgs e)  
{  
    Uri Logout = new Uri("/Logout", UriKind.Relative);  
    if (ServiceLayerSession != null)  
        ServiceLayerSession =  
slContext.Execute<MyServiceLayer.SAPB1.B1Session>(Logout, "POST", true).SingleOrDefault();  
    if (ServiceLayerSession == null)  
        txtMain.AppendText("You had been disconnected." +  
System.Environment.NewLine);  
}
```

5.2. To test the disconnect from the Service Layer.





© 2018 SAP SE or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. Please see

<http://www.sap.com/corporate-en/legal/copyright/index.epx#trademark> for additional trademark information and notices. Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors.

National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP SE or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP SE or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

SAP SE or its affiliated companies have no obligation to pursue any course of business outlined in this document or any related presentation, or to develop or release any functionality mentioned therein. This document, or any related presentation, and SAP SE's or its affiliated companies' strategy and possible future developments, products, and/or platform directions and functionality are all subject to change and may be changed by SAP SE or its affiliated companies at any time for any reason without notice. The information in this document is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, which speak only as of their dates, and they should not be relied upon in making purchasing decisions.