

Exercise Solution: Working with Entities

Service Layer

PUBLIC

INTRODUCTION

In this exercise, you will perform the following tasks:

1. Customize the Menu items
2. Create items on the default form
3. Add a Business Partner Master Data entry
4. Retrieve the list of the existing Business Partners

PREREQUISITE:

- This document is using the **C Sharp** (C#) language
- This document is using the Microsoft Visual Studio 2015
- Continue to work with the project finalized in previous exercise.
- This document is using the SAP Business One Studio for 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 – CUSTOMIZE THE MENU ITEMS

- 1.1. Uncomment the instantiation of the Menu class, Menu item creation function call and the Menu event subscriber.

In the **Main** function comment the following lines:

```
Menu MyMenu = new Menu();  
MyMenu.AddMenuItems();  
oApp.RegisterMenuEventHandler(MyMenu.SBO_Application_MenuEvent);
```

- 1.2. Change the menu collection to TB1_ServiceLayer. Refer to the function **AddMenuItems** in class *Menu*.

```
oCreationPackage.UniqueID = "TB1_ServiceLayer";  
oCreationPackage.String = "Service Layer";
```

- 1.3. Change the menu item to TB1_BPList. Refer to the function **AddMenuItems** in class *Menu*.

```
oMenuItem = Application.SBO_Application.Menus.Item("TB1_ServiceLayer");  
oCreationPackage.UniqueID = "TB1_BPList";  
oCreationPackage.String = "Business Partner List";
```

- 1.4. Reflect the menu related changes in the Menu item event handled function as well (**SBO_Application_MenuEvent**).

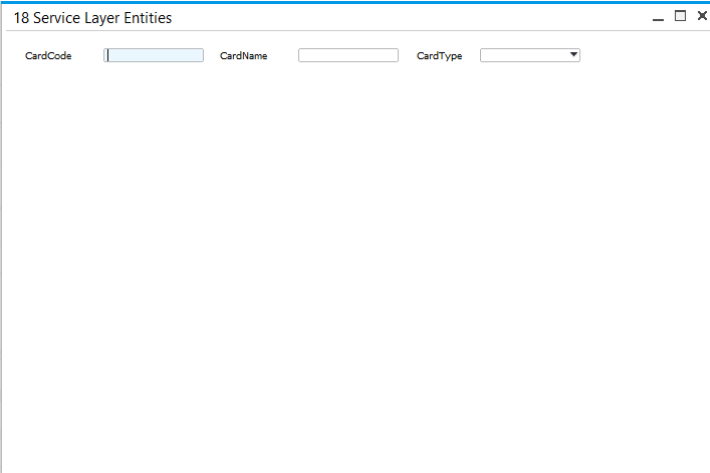
```
if (pVal.BeforeAction && pVal.MenuUID == "TB1_BPList")
```

2. TASK – CREATE ITEMS ON THE DEFAULT FORM

- 2.1. Create 2 *EditText* items and 2 *StaticText* items with captions **CardCode** (*UniqueID = txCardCode*) and **CardName** (*UniqueID = txCardName*).

The screenshot shows a window titled "18 Service Layer Entities". Inside the window, there are two input fields. The first field is labeled "CardCode" and has a light blue border. The second field is labeled "CardName" and has a light gray border. Both fields are empty and are positioned side-by-side at the top of the window.

2.2. Create *ComboBox* and *StaticText* item and set the *UniqueID* to value **cbCardType** for *ComboBox*.

The screenshot shows a window titled "18 Service Layer Entities". Inside the window, there are three input fields at the top: "CardCode" (a text box), "CardName" (a text box), and "CardType" (a dropdown menu). The rest of the window is empty.

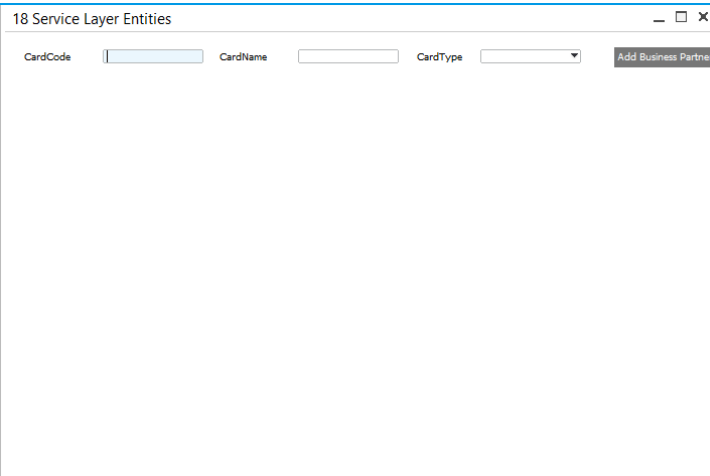
2.3. Assign default values for *ComboBox* item.

- Value = C; Description = Customer
- Value = S; Description = Supplier
- Value = L; Description = Lead

Extend the function `OnCustomInitialize` on the form.

```
private void OnCustomInitialize()
{
    cbCardType.Values.Add("C", "Customer");
    cbCardType.Values.Add("S", "Supplier");
    cbCardType.Values.Add("L", "Lead");
}
```

2.4. Create *Button* and set the *UniqueID* to value **btnAddBP** and the Caption to **Add Business Partner**.

The screenshot shows the same window as before, but now there is a button labeled "Add Business Partner" located to the right of the "CardType" dropdown menu.

- 2.5. Create *Button* and set the *UniqueID* to value **btnGetBP** and the Caption to **Get Business Partners**.

The screenshot shows a window titled '18 Service Layer Entities'. At the top, there are three input fields: 'CardCode' (with a light blue background), 'CardName', and 'CardType' (a dropdown menu). To the right of these fields is a button labeled 'Add Business Partner'. Below the input fields, centered, is a button labeled 'Get Business Partners'.

- 2.6. Create *Grid* and set the *UniqueID* to value **grResult**.

The screenshot shows the same window as before, but now a large, empty rectangular area (the grid) has been added below the 'Get Business Partners' button. The grid area is currently blank, with a light gray background.

3. TASK – ADD A BUSINESS PARTNER MASTER DATA ENTRY

- 3.1. Make the string **serviceLayerAddress** global and public in the **Program** class.

```
public static string serviceLayerAddress;
```

- 3.2. Create the function **RemoteSSLTLSCertificateValidate** for certificate validation.

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

3.3. Create the button click event for **btnAddBP**.

```
private void btnAddBP_ClickBefore(object sboObject, SAPboui COM.SBOItemEventArgs pVal,
out bool BubbleEvent)
{
    BubbleEvent = true;
}
```

3.4. Define the web request details in the function **btnAddBP_ClickBefore**.

```
var httpRequest = WebRequest.Create(Program.serviceLayerAddress +
"/BusinessPartners") as HttpWebRequest;
httpRequest.Method = "POST";
httpRequest.AllowAutoRedirect = false;
httpRequest.Timeout = 30 * 1000;
httpRequest.ServicePoint.Expect100Continue = false;
httpRequest.CookieContainer = new CookieContainer();
ServicePointManager.ServerCertificateValidationCallback +=
RemoteSSLTLSCertificateValidation;
```

3.5. Include to the function **btnAddBP_ClickBefore** the session cookie to the web request, which is stored in the string **sConnectionContext**.

```
string[] cookieItems = Program.sConnectionContext.Split(';');
foreach (var cookieItem in cookieItems)
{
    string[] parts = cookieItem.Split('=');
    if (parts.Length == 2)
    {
        httpRequest.CookieContainer.Add(httpRequest.RequestUri, new
Cookie(parts[0].Trim(), parts[1].Trim()));
    }
}
```

3.6. Add the web request body (function **btnAddBP_ClickBefore**).

```
using (var streamWriter = new StreamWriter(httpRequest.GetRequestStream()))
{
    string json = "{\"CardCode\": \"" + txCardCode.Value + "\", " +
                  "\"CardName\": \"" + txCardName.Value + "\", " +
                  "\"CardType\": \"" + cbCardType.Value + "\"}";
    streamWriter.Write(json); streamWriter.Flush(); streamWriter.Close();
}
```

3.7. Handle in function **btnAddBP_ClickBefore** the web response.

```
var httpResponse = (HttpWebResponse)httpRequest.GetResponse() as HttpWebResponse;
if (httpResponse.StatusCode != HttpStatusCode.Created)
    Application.SBO_Application.MessageBox(httpResponse.StatusDescription, 1, "Ok");
string responseContent = null;
using (var streamReader = new StreamReader(httpResponse.GetResponseStream()))
    responseContent = streamReader.ReadToEnd();

Application.SBO_Application.MessageBox(responseContent, 1, "Ok");
```

- 3.8. Hand the Business Partner Master Data by filling the CardCode, CardName and CardType. The response should be displayed in the MessageBox object.

```

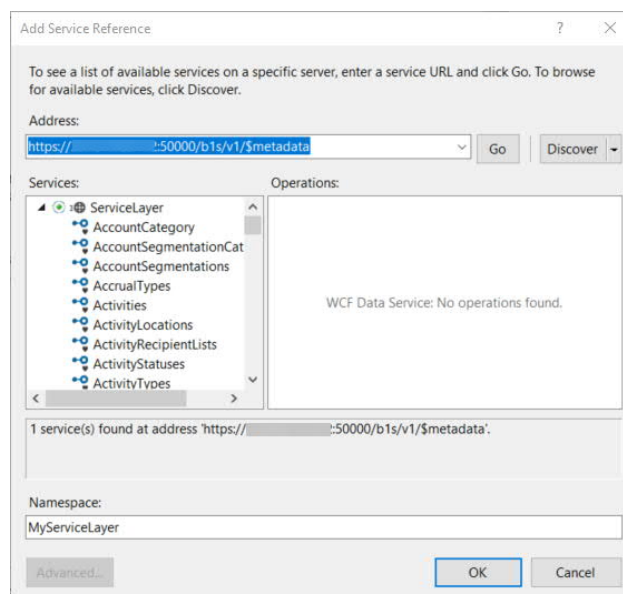
System Message
{
  "odata.metadata" : "https://10.0.9.104:50000/b1s/v1/$metadata#BusinessPartner/@Element",
  "CardCode" : "myBP16",
  "CardName" : "my business partner 16",
  "CardType" : "cld",
  "GroupCode" : 100,
  "Address" : null,
  "ZipCode" : null,
  "MailAddress" : null,
  "MailZipCode" : null,
  "Phone1" : null,
  "Phone2" : null,
  "Fax" : null,
  "ContactPerson" : null,
  "Notes" : null,
  "PayTermsGrpCode" : 1,
  "CreditLimit" : 0.0,
  "MaxCommitment" : 0.0,
  "DiscountPercent" : 0.0,
  "VatTable" : "vTable",
  "FederalTaxID" : null,
  "DeductibleAtSource" : "NO",
  "DeductionPercent" : 0.0,
  "DeductionValidUnit" : null,
  "PriceListNum" : 1,
  "InterestRatePercent" : 0.0,
  "CommissionPercent" : 0.0,
  "CommissionGroupCode" : 0,
  "FreeText" : null,
  "SalesPersonCode" : -1,
  "Currency" : "$",
  "RateDiffAccount" : null,
  "Cellular" : null,
  "AverageRate" : null,
  "City" : null,
  "Country" : null,
  "Country" : "US",
  "MailCity" : null,
  "MailCountry" : null,
  "MailCountry" : "US",
  "EmailAddress" : null,
  "Picture" : null,
}

```

4. TASK – RETRIEVE THE LIST OF THE EXISTING BUSINESS PARTNERS

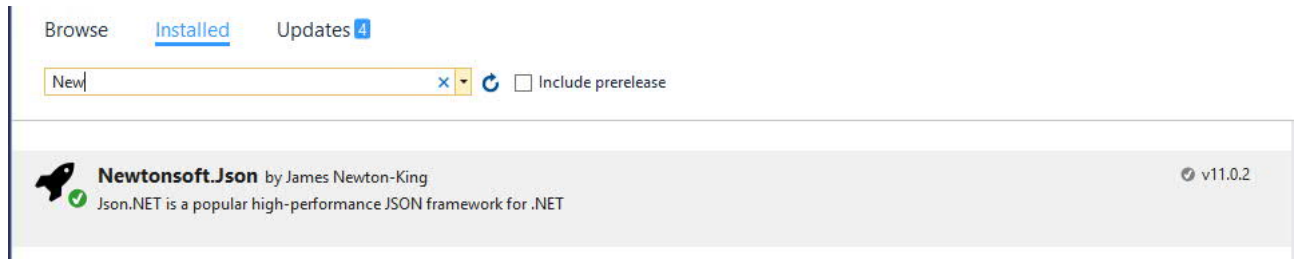
- 4.1. Add the metadata as service or file directly by selecting “Add Service Reference” into your Microsoft Visual Studio project, the metadata will be validated automatically.

Project → Add Service Reference



4.2. Install NuGet package – Newtonsoft.Json

Project → Manage NuGet Packages

4.3. Create the button click event for **btnGetBP**.

```
private void btnGetBP_ClickBefore (object sboObject, SAPboui COM. SBOItemEventArgs pVal, out bool BubbleEvent)
{
    BubbleEvent = true;
}
```

4.4. Define the web request details in the function **btnGetBP_ClickBefore**.

```
var httpWebRequest = WebRequest.Create(Program.serviceLayerAddress +
"/BusinessPartners") as HttpWebRequest;
httpWebRequest.Method = "GET";
httpWebRequest.AllowAutoRedirect = false;
httpWebRequest.Timeout = 30 * 1000;
httpWebRequest.ServicePoint.Expect100Continue = false;
httpWebRequest.CookieContainer = new CookieContainer();
httpWebRequest.Headers.Add("B1S-PageSize", "0");
ServicePointManager.ServerCertificateValidationCallback +=
RemoteSSLTLSCertificateValidation;
```

4.5. Include to the function **btnGetBP_ClickBefore** the session cookie to the web request, which is stored in the string **sConnectionContext**.

```
string[] cookieItems = Program.sConnectionContext.Split(';');
foreach (var cookieItem in cookieItems)
{
    string[] parts = cookieItem.Split('=');
    if (parts.Length == 2)
    {
        httpWebRequest.CookieContainer.Add(httpWebRequest.RequestUri, new
Cookie(parts[0].Trim(), parts[1].Trim()));
    }
}
```

4.6. Handle in function **btnGetBP_ClickBefore** the web response.

```
var httpResponse = (HttpWebResponse)httpWebRequest.GetResponse() as HttpWebResponse;
if (httpResponse.StatusCode != HttpStatusCode.OK)
    Application.SBO_Application.MessageBox(httpResponse.StatusDescription, 1, "Ok");
```


4.7. Display the results in the *Grid* object, the code can be handled in function **btnGetBP_ClickBefore**.

```
string responseContent = null;
using (var streamReader = new StreamReader(httpResponse.GetResponseStream()))
{
    responseContent = streamReader.ReadToEnd();
    var oResult = JsonConvert.DeserializeObject<IDictionary<string,
object>>(responseContent);

    List<MyServiceLayer.SAPB1.BusinessPartner> l_B1BusinessPartners;
    l_B1BusinessPartners =
JsonConvert.DeserializeObject<List<MyServiceLayer.SAPB1.BusinessPartner>>(oResult["value"].T
oString());

    SAPboui COM. Form oForm = Application.SBO_Application.Forms.GetForm("TB1300.Form1", 0);
    oForm.Freeze(true);
    SAPboui COM. DataTable oDataTable;

    if (grResult.Columns.Count == 0)
        oDataTable = oForm.DataSources.DataTables.Add("myBP sDT");
    else
        oDataTable = oForm.DataSources.DataTables.Item("myBP sDT");

    oDataTable.Clear();
    oDataTable.Columns.Add("CardCode", SAPboui COM. BoFieldstype.ft_AlphaNumeric);
    oDataTable.Columns.Add("CardName", SAPboui COM. BoFieldstype.ft_AlphaNumeric);
    oDataTable.Columns.Add("CardType", SAPboui COM. BoFieldstype.ft_AlphaNumeric);

    for (int i = 0; i <= l_B1BusinessPartners.Count - 1; i++)
    {
        oDataTable.Rows.Add();
        oDataTable.SetValue(0, i, l_B1BusinessPartners[i].CardCode);
        oDataTable.SetValue(1, i, l_B1BusinessPartners[i].CardName);
        oDataTable.SetValue(2, i, l_B1BusinessPartners[i].CardType);
    }
    grResult.DataTable = oForm.DataSources.DataTables.Item("myBP sDT");
    oForm.Freeze(false);
}
```

4.8. Extend the CardCode column in the *Grid* item with the *LinkButton* object.

The function **btnGetBP_ClickBefore** need to be extended, before the form gets unfreeze.

```
grResult.Columns.Item("CardCode").Type = SAPboui COM. BoGridColumnType.gct_EditText;
    SAPboui COM. EditTextColumn oEdit =
(SAPboui COM. EditTextColumn)grResult.Columns.Item("CardCode");
    oEdit.LinkObjectType = "2";
```

4.9. Try to get the Business Partner Records into the *grid* object.

18 Service Layer Entities

CardCode

CardName

CardType

Add Business Partner

Get Business Partners

CardCode	CardName	CardType
➡ C20000	Maxi-Teq	cCustomer
➡ C23900	Parameter Technology	cCustomer
➡ C25000	Star Company	cCustomer
➡ C26000	River Inc	cCustomer
➡ C30000	Microchips	cCustomer
➡ C40000	Earthshaker Corporation	cCustomer
➡ C42000	Mashina Corporation	cCustomer
➡ C50000	ADA Technologies	cCustomer
➡ C60000	SG Electronics	cCustomer
➡ C70000	Aquent Systems	cCustomer
➡ C99998	Web Customer	cCustomer
➡ C99999	One Time Customer	cCustomer
➡ L10001	Andreas Ackermann	cLid
➡ L10002	Werner Richter	cLid
➡ V10000	Acme Associates	cSupplier
➡ V1010	Far East Imports	cSupplier
➡ V20000	Lasercom	cSupplier
➡ V23000	Anthony Smith	cSupplier
➡ V30000	Blockies Corporation	cSupplier



© 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.