FIREBRAND

Microsoft Dynamics CRM 2013 Extending Training Labs

www.firebrandtraining.com

Labs content

Introduction to the Labs

Purpose

The purpose of the labs is to reinforce the learning in the courseware.

In the main, the labs focus on the mechanics of how to use the multiple extensibility points of Dynamics CRM. The labs do not require deep programming skills and you will not be asked to write lots of code.

Case Study

The labs use a mixture of system CRM entities and the custom entities and fields you created in the Customisation course ie the NPD and Feedback solution.

You will also see the Event Management solution that your instructor will use for Demos

Location of files

The Microsoft Dynamics CRM SDK is installed under C:\CRM on the virtual machine.

The lab files are on the Extending Labs drive (F:\).

Visual Studio

Visual Studio 2010 is installed on the virtual machine and is available from the Taskbar on the Desktop

Languages used

- C#
- JavaScript
- JQuery

Lab 2A – Discovery Service

Objectives

To connect to CRM using the helper classes in the SDK

In this lab you will cover:

- Discovery web service
- Helper classes

Steps

Generate Proxy Classes

1. Browse to F:\Labs\Module2
2. Double click on earlybound.cmd

Check SDKProxyTypes.cs has been created

Locate Helper Classes

Browse to C:\CRM\SDK\SampleCode\CS\HelperCode and view the files

Create Folder for your projects

Create folder Labs under C:\CRM

Create folder Module2 under C:\CRM\Labs

Create Project

Start Visual Studio and click on New Project

Select Visual C# and then Console Application template

Change name to ConnectionLab

Change Location to C:\CRM\Labs\Module2

Click on OK

Set Target Framework

Right Click on the ConnectionLab project in the Solution Explorer pane and select Properties

Change Target framework to .NET 4 Framework

Add Helper Classes

Right Click on the ConnectionLab project in the Solution Explorer pane and select Add and then Existing Item

Browse to C:\CRM\SDK\SampleCode\CS\HelperCode and select CrmServiceHelpers.cs

Repeat for DeviceIdManager.cs

Add Proxy Class

Right Click on the ConnectionLab project in the Solution Explorer pane and select Add and then Existing Item

Browse to location of SDKProxyTypes.cs created above and select it.

Add .NET Assemblies

Right Click on the Reference in the Solution Explorer pane and select Add Reference

Select the .NET tab and select

- System.Runtime.Serialization
- System.Security
- System.ServiceModel
- System.DirectoryServices
- System.DirectoryServices.AccountManagement

Add Windows Identity Foundation Assembly

Right Click on the Reference in the Solution Explorer pane and select Add Reference

Select the Browse tab and browse to C:\Program Files\Reference Assemblies\Microsoft\Windows Identity Foundation\v3.5 and select Microsoft.IdentityModel.dll

Add CRM/XRM Assemblies

Right Click on the Reference in the Solution Explorer pane and select Add Reference

Select the Browse tab and browse to C:\CRM\SDK\bin and select Microsoft.Xrm.Sdk.dll

Repeat for Microsoft.Crm.Sdk.Proxy.dll

Add Using Statements

Add to Program.cs

```
using Microsoft.Xrm.Sdk;
using Microsoft.Xrm.Sdk.Client;
using Microsoft.Xrm.Sdk.Discovery;
using Microsoft.Crm.Sdk.Samples;
using System.ServiceModel;
using System.Runtime.Serialization;
```

Add Discovery Service Code

Add to Main in Program.cs

to:

```
ServerConnection serverConnect = new ServerConnection();
ServerConnection.Configuration serverconfig = serverConnect.GetServerConfiguration();

using (DiscoveryServiceProxy _serviceProxy = new DiscoveryServiceProxy(serverConfig.DiscoveryUri,
                                    serverConfig.HomeRealmUri,
                                    serverConfig.Credentials,
                                    serverConfig.DeviceCredentials))
        {
          // You can choose to use the interface instead of the proxy.
          IDiscoveryService service = _serviceProxy;

          // Retrieve details about all organizations discoverable via the Discovery service.
          RetrieveOrganizationsRequest orgsRequest =
            new RetrieveOrganizationsRequest()
            {
              AccessType = EndpointAccessType.Default,
              Release = OrganizationRelease.Current
            };
          RetrieveOrganizationsResponse organizations =
            (RetrieveOrganizationsResponse)service.Execute(orgsRequest);

          // Print each organization's friendly name, unique name and URLs for each of its
endpoints.
          Console.WriteLine();
          Console.WriteLine("Retrieving details of each organization:");
          foreach (OrganizationDetail organization in organizations.Details)
          {
            Console.WriteLine("Organization Name: {0}", organization.FriendlyName);
            Console.WriteLine("Unique Name: {0}", organization.UniqueName);
            Console.WriteLine("Endpoints:");
            foreach (var endpoint in organization.Endpoints)
            {
              Console.WriteLine("  Name: {0}", endpoint.Key);
              Console.WriteLine("  URL: {0}", endpoint.Value);
            }
          }
          Console.WriteLine("End of listing");
          Console.WriteLine();

          // Retrieve details about a single organization discoverable via the Discovery service.
          RetrieveOrganizationRequest orgRequest =
            new RetrieveOrganizationRequest()
```

```
      {
        UniqueName = organizations.Details[organizations.Details.Count -1].UniqueName,
        AccessType = EndpointAccessType.Default,
        Release = OrganizationRelease.Current
      };
    RetrieveOrganizationResponse org =
      (RetrieveOrganizationResponse)service.Execute(orgRequest);

    // Print the organization's friendly name, unique name and URLs for each of its endpoints.
    Console.WriteLine();
    Console.WriteLine("Retrieving details of specific organization:");
    Console.WriteLine("Organization Name: {0}", org.Detail.FriendlyName);
    Console.WriteLine("Unique Name: {0}", org.Detail.UniqueName);
    Console.WriteLine("Endpoints:");
    foreach (KeyValuePair<EndpointType, string> endpoint in org.Detail.Endpoints)
    {
      Console.WriteLine("  Name: {0}", endpoint.Key);
      Console.WriteLine("  URL: {0}", endpoint.Value);
    }
    Console.WriteLine("End of listing");
    Console.WriteLine();
    Console.ReadLine();
  }
```

This code is in F:\Labs\Module2\Discovery.cs

Build and Test

Build project and run

Close

Close Visual Studio

Lab 2B – Organisation Service and Basic Operations

Objectives

To connect to CRM using the helper classes in the SDK and amend data

In this lab you will cover:

- Organisation web service
- Helper classes
- Early Binding
- Create method
- Retrieve method
- Update method

Steps

Copy Previous Project

In C:\CRM\Labs\Module2

Copy ConnectionLab Folder and call it CRUDLab

Open Project

In C:\CRM\Labs\Module2\CRUDLab

Double click on ConnectionLab.sln

Rename Solution and Project

Right click on Solution in Solution Explorer and rename as CRUDLab

Do the same for the Project ie rename

Right click on the Project and select Properties and change the AssemblyName and Namespace to CRUDLab

Open Program.cs and change Namespace to CRUDLab

Remove Code

Remove all code within Main

Add Using Statements

Add to Program.cs

```
using Microsoft.Xrm.Sdk.Query;
```

Add Code

Add to Main in Program.cs to:

- Connect to Organization Service
- Use Early Bound
- Create an idea record
- Retrieve the idea record
- Update the idea record

```
ServerConnection serverConnect = new ServerConnection();
ServerConnection.Configuration serverConfig = serverConnect.GetServerConfiguration();

using (OrganizationServiceProxy _serviceProxy =
ServerConnection.GetOrganizationProxy(serverConfig))
        {
          // This statement is required to enable early-bound type support.
          _serviceProxy.EnableProxyTypes();
          IOrganizationService _service = (IOrganizationService)_serviceProxy;

          awcnpd_idea Idea = new awcnpd_idea
          {
             awcnpd_name = "CRUD Idea",
             awcnpd_targetmarketsize = 5000,
             awcnpd_description = "Description"
          };
          Guid _ideaId = _service.Create(Idea);

          awcnpd_idea retrievedIdea =
(awcnpd_idea)_service.Retrieve(awcnpd_idea.EntityLogicalName, _ideaId, new ColumnSet(true));

          awcnpd_idea updateIdea = new awcnpd_idea
          {
             awcnpd_ideaId = _ideaId,
             awcnpd_targetmarketsize = null,
             awcnpd_description = string.Empty
          };
          _service.Update(updateIdea);
        }
```

This code is in F:\Labs\Module2\CRUD.cs

Enable Auditing

Enable Auditing on Idea entity and Publish

Build and Test

Build project and run

Check Audit History

Close

Close Visual Studio

Lab 3A – QueryExpression

Objectives

To perform basic query to retrieve all Ideas that have a prototype

In this lab you will cover:

- QueryExpression

Steps

Create Folder for your projects

Create folder Module3 under C:\CRM\Labs

Copy Previous Project

In C:\CRM\Labs\Module3

Copy CRUDLab Folder from C:\CRM\Labs\Module2 and call it QELab

Open Project

In C:\CRM\Labs\Module3\QELab

Double click on CRUDLab.sln

Rename Solution and Project

Right click on Solution in Solution Explorer and rename as QELab

Do the same for the Project ie rename

Right click on the Project and select Properties and change the AssemblyName and Namespace to QELab

Open Program.cs and change Namespace to QELab

Remove Code

Remove all code within Main that creates, retrieves and updates.

Add Code

Add to Main in Program.cs to:

- Connect to Organization Service
- Use Early Bound
- Use Query Expression to retrieve all active ideas
- Loop and list all idea names

```
            ServerConnection serverConnect = new ServerConnection();
            ServerConnection.Configuration serverConfig = serverConnect.GetServerConfiguration();
            using (OrganizationServiceProxy _serviceProxy =
ServerConnection.GetOrganizationProxy(serverConfig))
            {
                // This statement is required to enable early-bound type support.
                _serviceProxy.EnableProxyTypes();
                IOrganizationService _service = (IOrganizationService)_serviceProxy;

                QueryExpression ideaquery = new QueryExpression
                {
                    EntityName = awcnpd_idea.EntityLogicalName,
                };
                ideaquery.ColumnSet.AddColumn("awcnpd_name");
                ideaquery.ColumnSet.AddColumn("awcnpd_targetmarketsize");
                ideaquery.ColumnSet.AddColumn("awcnpd_description");
                ideaquery.Criteria.AddCondition("statecode", ConditionOperator.Equal,
(int)awcnpd_ideaState.Active);
                ideaquery.PageInfo.ReturnTotalRecordCount = true;
                ideaquery.Distinct = true;
                ideaquery.AddLink(awcnpd_prototype.EntityLogicalName, "awcnpd_ideaid",
"awcnpd_originatingideaid", JoinOperator.Inner);
                EntityCollection ideas = _service.RetrieveMultiple(ideaquery);
                Console.WriteLine("Number of Ideas=" + ideas.TotalRecordCount.ToString());
                foreach (awcnpd_idea idea in ideas.Entities)
                {
                    Console.WriteLine(idea.awcnpd_name);
                }
                Console.WriteLine();
            }
```

This code is in F:\Labs\Module3\QE.cs

Build and Test

Build project and run

Close

Close Visual Studio

Lab 3B – LINQ

Objectives

To perform LINQ operations

In this lab you will cover:

- LINQ
- Early Binding

Steps

Generate Proxy Classes

1. Browse to F:\Labs\Module3
2. Double click on earlyboundwithLINQ.cmd

Check SDKProxyTypeswithLinq.cs has been created

Copy Previous Project

In C:\CRM\Labs\Module3

Copy QELab Folder and call it LINQLab

Open Project

In C:\CRM\Labs\Module3\LINQLab

Double click on QELab.sln

Rename Solution and Project

Right click on Solution in Solution Explorer and rename as LINQLab

Do the same for the Project ie rename

Right click on the Project and select Properties and change the AssemblyName and Namespace to LINQLab

Open Program.cs and change Namespace to LINQLab

Add New Proxy Class

Right Click on SDKProxyTypes.cs and Exclude from Project

Right Click on the project in the Solution Explorer pane and select Add and then Existing Item

Browse to location of SDKProxyTypeswithLinq.cs created above and select it.

Remove Code

Remove all code within Main

Add Code

Add to Main in Program.cs to:

- Connect to Organization Service
- Use Early Bound
- Create Data Context
- Use LINQ to retrieve all ideas with prototypes
- Loop and list all idea and prototype names
- Create Data Context
- Add three news idea to the Data Context
- Save Changes

```
var oContext = new MyDataContext(_service);

var linqquery = from i in oContext.awcnpd_ideaSet
            join p in oContext.awcnpd_prototypeSet
        on i.awcnpd_ideaId equals p.awcnpd_originatingideaid.Id
        select new { idea_name = i.awcnpd_name, prototype_name = p.awcnpd_name};
int j = 0;
foreach (var record in linqquery)
{
   Console.WriteLine(record.idea_name + " - " + record.prototype_name);
   j++;
}
Console.WriteLine("Number of Ideas=" + j.ToString());
Console.WriteLine();

var _Context = new MyDataContext(_service);

awcnpd_idea newidea1 = new awcnpd_idea
{
   awcnpd_name = "LINQ Idea 1",
   awcnpd_targetmarketsize = 1,
   awcnpd_description = "Description"
};
_Context.AddObject(newidea1);

awcnpd_idea newidea2 = new awcnpd_idea
{
   awcnpd_name = "LINQ Idea 2",
    awcnpd_targetmarketsize = 2,
    awcnpd_description = "Description"
};
_Context.AddObject(newidea2);

awcnpd_idea newidea3 = new awcnpd_idea
{
    awcnpd_name = "LINQ Idea 3",
    awcnpd_targetmarketsize = 3,
    awcnpd_description = "Description"
};
_Context.AddObject(newidea3);

_Context.SaveChanges();
```

This code is in F:\Labs\Module3\LINQ.cs

Build and Test

Build project and run

Close

Close Visual Studio

Lab 5A – Developer Toolkit

Objectives

To install Developer Toolkit and build proxy class

In this lab you will cover:

- Developer Toolkit
- Early Binding

Steps

Install Toolkit

Open Command Prompt as Administrator

Change Directory to C:\CRM\SDK\Tools\DeveloperToolkit

Run CrmDevelperTools_Installer.msi

Create Project from CRM Template

Open Visual Studio

Create a New Project with Dynamics CRM Template – Dynamics CRM 2013 Plugin Library – called Firebrand.CRM.NPD in C:\CRM\Labs\Module5

Connect

Enter the following information when prompted

- CRM Server: lon-dc1
- Port: 5555
- Protocol: HTTP
- Authentication Details: Use Default Credentials
- Organisation: Adventure Works Cycles
- Solution: NPD and Feedback

Generate Entities.cs

Expand Tree in CRM Explorer

Right click on Entities and click Generate Wrapper to create Entities.cs

Lab 5B – AutoNumber Plugin

Objectives

Create a plugin to generate sequential number for each new idea

In this lab you will cover:

- Use Developer Toolkit to change Schema
- Create Plugin with Developer Toolkit
- Deploy Plugin with Plugin Registration Tool
- Debug

Steps

Add Field

Use Open Solution Icon on CRM Explorer bar to open the NPD and Feedback solution

Add field called Reference (awcnpd_refeence) to Idea entity with type Whole Number

Add Reference field to Idea form in the Header

Save form and Publish

Regenerate Entities.cs

Right click on Entities and click Generate Wrapper to update Entities.cs

Create Plugin

Right click on Idea entity and select Create Plug-in

Set Message to Create

Set Stage = Pre-Operation

Click on OK

This will create new class PreIdeaCreate.cs

Add code

Edit PreIdeaCreate.cs and add statements

```
using Microsoft.Xrm.Sdk.Query;
using Firebrand.CRM.NPD.Entities;
```

add the code to the ExecutePreIdeaCreate to:

- Retrieve most recently created Idea record
- If no records returned set add attribute awcnpd_reference set to 1
- If a record returned set add attribute awcnpd_reference set to increment the value on existing record

```
Entity idea = (Entity)localContext.PluginExecutionContext.InputParameters["Target"];

QueryExpression qe = new QueryExpression
{
   EntityName = awcnpd_idea.EntityLogicalName
};
qe.ColumnSet.AddColumn("awcnpd_reference");
OrderExpression oe = new OrderExpression
{
   AttributeName = "createdon",
   OrderType = OrderType.Descending
};
qe.Orders.Add(oe);
qe.PageInfo = new PagingInfo
{
   Count = 1,
   PageNumber = 1,
   PagingCookie = null
};

EntityCollection ec = localContext.OrganizationService.RetrieveMultiple(qe);

if (ec.Entities.Count == 0)
{
   idea.Attributes.Add("awcnpd_reference", 1);
}
else
{
   Entity latestidea = (Entity)ec[0];
   if (latestidea.Attributes.Contains("awcnpd_reference"))
      idea.Attributes.Add("awcnpd_reference", (int)latestidea["awcnpd_reference"] + 1);
   else
      idea.Attributes.Add("awcnpd_reference", 1);
}
```

This code is in F:\Labs\Module5\PreIdeaCreate.cs

Sign the Assembly

Right click on the Project and select Properties

Select Signing Tab

Check Sign the assembly

Select <New>

Enter filename as Firebrand.CRM.NPD.snk

Uncheck Protect my key file with a password

Click on OK

Build

Build the project

Deploy Assembly

Browse to C:\CRM\SDK\Bin

Run PluginRegistration.exe

Click on Create New Connection button

Enter the following information when prompted

- Label: AWC
- Discovery URL: http://lon-dc1:5555
- User Name: blank
- Use Saved Credentials: Checked

Click on Connect

Select Adventure Works Cycles and click on Connect

Click on Register New Assembly

Browse to your Firebrand.CRM.NPD.dll file

Select Sandbox

Select Database

Click on Register Selected Plugins

Add Step

Expand (Assembly) Firebrand.CRM.NPD

Right Click on (Plugin) Firebrand.CRM.NPD.PreIdeaCreate and select Register New Step

Enter the following information when prompted

- Message: Create
- Primary Entity: awcnpd_idea
- Pipeline Stage: Pre-operation

Click on Register New Step

Copy PDB file

Copy Firebrand.CRM.NPD.pdb to C:\Program Files\Microsoft Dynamics CRM\Server\bin\assembly

Prevent Sandbox Worker Processes from shutting down

Start regedit

Add new DWORD SandboxDebugPlugins to HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSCRM\

Set SandboxDebugPlugins to 1

Debug

In Visual Studio

Select Debug Attach to Process

Check Show processes from all users and all sessions

If no Microsoft.Crm.Sandbox.WorkerProcess.exe processes are showing you need to create a new Idea record to force the worker process to start

If more than one worker process attach to all of them

Set Breakpoint

Create new idea record

Add Plugin to Solution

Open NPD and Feedback solution

Click on Plug-in Assemblies

Click on Add Existing

Select Firebrand.CRM.NPD

Click on SDK Message Processing Steps

Click on Add Existing

Select Firebrand.CRM.NPD.PreIdeaCreate

Lab 5C – Random Number Custom Workflow Assembly

Objectives

To install Developer Toolkit and build proxy class

In this lab you will cover:

- Create Custom Workflow Activity with Developer Toolkit
- Deploy Custom Workflow Activity with Plugin Registration Tool
- Debug

Steps

Create Project from CRM Template

Open Visual Studio

Create a New Project with Dynamics CRM Template – Dynamics CRM 2013 Workflow Library – called Firebrand.CRM.NPDActivites in C:\CRM\Labs\Module5

Add .NET Assemblies

Right Click on the Reference in the Solution Explorer pane and select Add Reference

Select the .NET tab and select

- System.Activities
- System.ServiceModel
- System.Runtime.Serialization

Add Class

Right click on Project and select Add New Class

Select Dynamics CRM templates and Workflow Activity Class template

Name the class RandomNumber.cs

Enter the following information when prompted

- Name: RandomNumber
- Friendly Name: Random Number
- Activity Group Name: NPD

Add Input Parameter

Edit RandomNumber.cs and add code

```
[Input("Max Value")]

[RequiredArgument]

public InArgument<int> MaxValue { get; set; }
```

Add Output Parameter

Edit RandomNumber.cs and add code

```
[Output("Result")]

public OutArgument<int> Result { get; set; }
```

Add code

Edit RandomNumber.cs and add code under TODO

- Get MaxValue
- Generate random number between 1 and MaxValue
- Set Result

```
int _max = MaxValue.Get(executionContext);
Random _rnd = new Random();
int _result = _rnd.Next(1, _max);
Result.Set(executionContext, _result);
```

This code is in F:\Labs\Module5\RandomNumber.cs

Sign the Assembly

Right click on the Project and select Properties

Select Signing Tab

Check Sign the assembly

Select <New>

Enter filename as Firebrand.CRM.NPDActivities.snk

Uncheck Protect my key file with a password

Click on OK

Deploy Assembly

Run PluginRegistration.exe

Connect

Select Adventure Works Cycles and click on Connect

Click on Register New Assembly

Browse to your Firebrand.CRM.NPDActivities.dll file

Click on Register Selected Plugins

Create Workflow Process

Create background workflow process for Idea entity called Add Reference

Set to Run On Demand only

Uncheck Automatically delete completed workflow jobs option

Add Check Condition

```
    If Reference contains data then
        Stop
```

Add Step: Firebrand.CRM.NPDActivities (1.0.0.0) -> Firebrand.CRM.NPDActivities.RandomNumber

Click on Set Properties

Set Max Value to 99

Save and Close

Add Step to Update Idea record

Click on Set Properties

Click in Reference field in Additional fields

Use Form Assistant to set Reference to Firebrand.CRM.NPDActivities.RandomNumber.Result

Save and Close

Activate process

Copy PDB file

Copy Firebrand.CRM.NPDActivities.pdb to C:\Program Files\Microsoft Dynamics CRM\Server\bin\assembly

Restart Microsoft Dynamics CRM Asynchronous Processing Service (CRMAsyncService.exe)

Debug

In Visual Studio

Select Debug Attach to Process

Check Show processes from all users and all sessions

Attach to CRMAsyncService.exe

If more than such process attach to all of them

Set Breakpoint

Select existing idea record and run the workflow

Add CWA to Solution

Open NPD and Feedback solution

Click on Plug-in Assemblies

Click on Add Existing

Select Firebrand.CRM.NPDActivities

Lab 6A – Form Scripts

Objectives

To show form script

In this lab you will cover:

- Script Web Resource
- OnLoad event
- Notifications

Steps

Create Web Resource

In the NPD and Feedback Solution create a new JavaScript web resource called awcnpd_/scripts/idea.js with display name idea.js

Create function

Add following script to the web resource

```
function awcnpd_ideaOnLoad()
{
        var targetAttribute = Xrm.Page.getAttribute('awcnpd_targetmarketsize');
        if (targetAttribute.getValue() == null)
                Xrm.Page.ui.setFormNotification('Target Market', 'WARNING');
}
```

Save and Publish the web resource

Add Library to Form

Edit the Idea form

Open Form Properties

Add the Javascript library web resource

Associate function with OnLoad event

Under Event Handlers

Select Form and OnLoad and click on Add

Select idea.js library

Enter awcnpd_ideaOnLoad as function name

Save and Publish the form

Test

Open up Idea record with no value in Target Market

Lab 6B – URL Addressable Forms

Objective

To show how parameters can be passed into forms

In this lab you will cover:

- HTML Web Resource
- Calling HTML Web Resource passing single parameter Data
- Form Parameters

Steps

SDK Sample

This lab is taken from the SDK: Pass multiple values to a web resource through the data parameter

Files can be found in C:\CRM\SDK\samplecode\js\webresources and F:\Labs\Module6

Create Web Resource

In the NPD and Feedback Solution create a new web resource called awcnpd_/content/ShowDataParams.html with display name ShowDataParams.html

Upload HTML

Browse and upload ShowDataParams.htm from F:\Labs\Module6

Save and Publish

Test Web Resource

http://lon-dc1:5555/AdventureWorksCycles/WebResources/awcnpd_/content/ShowDataParams.html?Data=first%3DFirst%20Value%26second%3DSecond%20Value%26third%3DThird%20Value

Add Web Resource to Form

Edit the Idea form and add the web resource to the form, named ShowDataParams.

Set Number of Rows = 8

Set Custom Parameter (Data) in the web resource to first=A&second=B&third=C

Save and Publish

Test

http://lon-dc1:5555/AdventureWorksCycles/userdefined/edit.aspx?etc=10006&id={8A029BEC-BFF5-E311-9403-00155D027311}

Add Form Parameters

Edit the Idea form and in Form Properties add three string field parameters:

- awcnpd_param1
- awcnpd_param2
- awcnpd_param3

Remove the Custom Parameter (Data) in the web resource

Save and Publish

Edit JavScript function to read the paramaters

Add the following JavaScript to the awcnpd_ideaOnLoad function in web resource awcnpd_/scripts/idea.js

```
var urlParams = Xrm.Page.context.getQueryStringParameters();
var urlParam1 = urlParams[‘awcnpd_param1’];
var urlParam2 = urlParams[‘awcnpd_param2’];
var urlParam3 = urlParams[‘awcnpd_param3’];
Xrm.Page.ui.setFormNotification(urlParam1, ‘INFO’);
Xrm.Page.ui.setFormNotification(urlParam2, ‘INFO’);
Xrm.Page.ui.setFormNotification(urlParam3, ‘INFO’);
```

Save and Publish the web resource

Test Form Parameters on new record

http://lon-dc1:5555/AdventureWorksCycles/main.aspx?etc=10006&pagetype=entityrecord&extraqs=awcnpd_param1%3DOne%26awcnpd_param2%3DTwo%26awcnpd_param3%3DThree

Test Form Schema Names as Parameters on new record

http://lon-dc1:5555/AdventureWorksCycles/main.aspx?etc=10006&pagetype=entityrecord&extraqs=awcnpd_name%3DMy%20Idea%26awcnpd_targetmarketsize%3D999

Lab 7A – SDK.REST.js

Objective

To show how OData can be used with SDK.REST.js library without JQUERY

In this lab you will cover:

- How to perform the most basic data operations using the REST endpoint and JavaScript.
- Use the XMLHttpRequest object
- CRUD operations on a new account record.

Steps

SDK Sample

This lab is taken from the SDK, Sample: Create, retrieve, update, and delete using the OData endpoint with JavaScript

Files can be found in C:\CRM\SDK\SampleCode\JS\RESTEndpoint\JavaScriptRESTDataOperations and F:\Labs\Module7

Create Web Resources

This sample uses the following web resources:

- sample_/JavaScriptRESTDataOperationsSample.htm - An HTML page that contains the UI elements for this sample.
- sample_/Scripts/JavaScriptRESTDataOperationsSample.js - A library that contains functions to manage the UI elements on the page and includes the actions performed by the sample.
- sample_/Scripts/SDK.REST.js - A reusable, generic library that simplifies asynchronous data operations using the REST Endpoint for web resources.
- sample_/Scripts/json2.js

Import the managed solution JavaScriptRESTDataOperations_1_0_0_1_managed.zip

Review Code

Review the sample_/JavaScriptRESTDataOperationsSample.htm and sample_/Scripts/JavaScriptRESTDataOperationsSample.js files either from Default Solution or by opening Visual Studio solution in F:\Labs\Module7.

Test

View the solution configuration page to observe the actions performed

Lab 7B – JQUERY

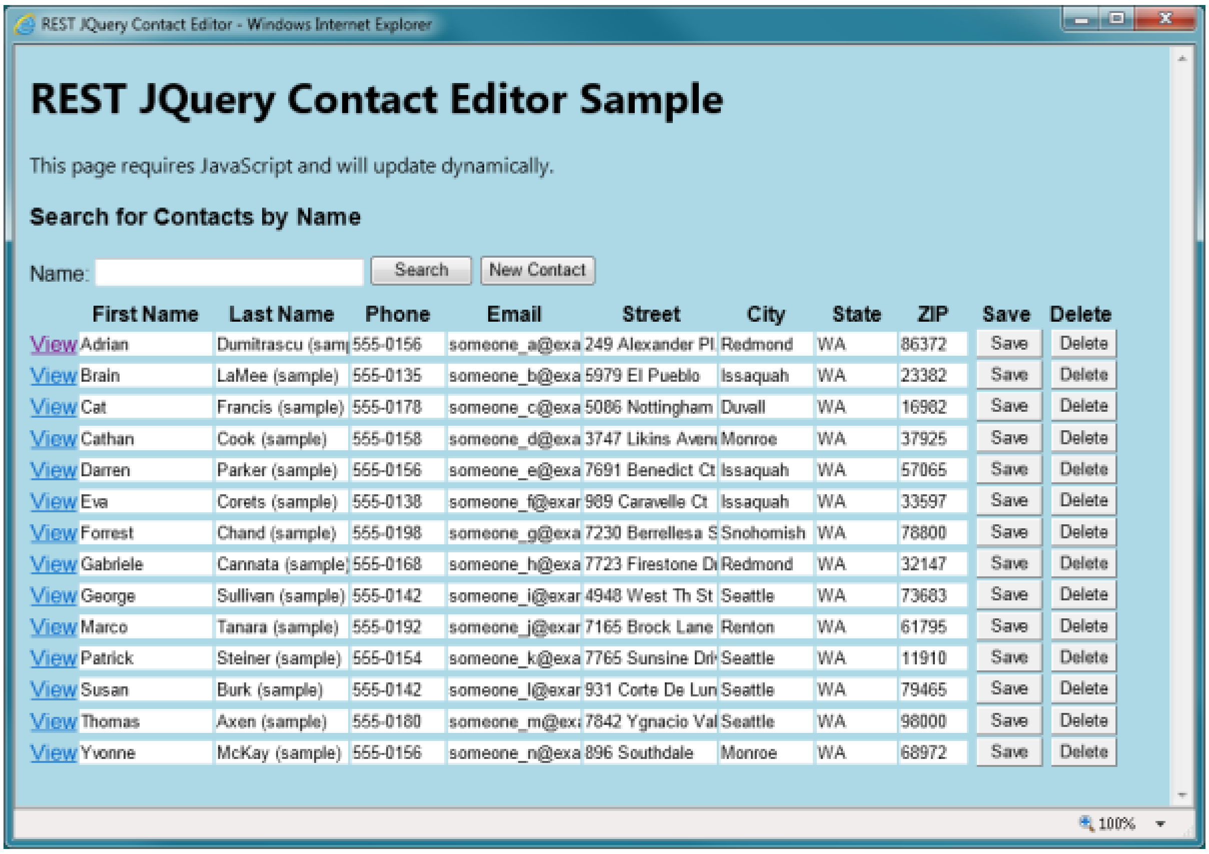
Objective

To show how OData can be used with JQUERY

In this lab you will cover:

- This sample shows how to use of the REST endpoint in a basic application that provides interactive user interface using HTML, JScript, and JQuery

This screenshot shows the user interface for the REST Endpoint JQuery Contact Editor.



REST JQuery Contact Editor Sample

This page requires JavaScript and will update dynamically.

Search for Contacts by Name

Name: [Search] [New Contact]

	First Name	Last Name	Phone	Email	Street	City	State	ZIP	Save	Delete
View	Adrian	Dumitrascu (sam	555-0156	someone_a@exa	249 Alexander Pl	Redmond	WA	86372	Save	Delete
View	Brain	LaMee (sample)	555-0135	someone_b@exa	5979 El Pueblo	Issaquah	WA	23382	Save	Delete
View	Cat	Francis (sample)	555-0178	someone_c@exa	5086 Nottingham	Duvall	WA	16982	Save	Delete
View	Cathan	Cook (sample)	555-0158	someone_d@exa	3747 Likins Aven	Monroe	WA	37925	Save	Delete
View	Darren	Parker (sample)	555-0156	someone_e@exa	7691 Benedict Ct	Issaquah	WA	57065	Save	Delete
View	Eva	Corets (sample)	555-0138	someone_f@exar	989 Caravelle Ct	Issaquah	WA	33597	Save	Delete
View	Forrest	Chand (sample)	555-0198	someone_g@exa	7230 Berrellesa S	Snohomish	WA	78800	Save	Delete
View	Gabriele	Cannata (sample)	555-0168	someone_h@exa	7723 Firestone Di	Redmond	WA	32147	Save	Delete
View	George	Sullivan (sample)	555-0142	someone_i@exar	4948 West Th St	Seattle	WA	73683	Save	Delete
View	Marco	Tanara (sample)	555-0192	someone_j@exar	7165 Brock Lane	Renton	WA	61795	Save	Delete
View	Patrick	Steiner (sample)	555-0154	someone_k@exa	7765 Sunsine Dri	Seattle	WA	11910	Save	Delete
View	Susan	Burk (sample)	555-0142	someone_l@exar	931 Corte De Lun	Seattle	WA	79465	Save	Delete
View	Thomas	Axen (sample)	555-0180	someone_m@ex	7842 Ygnacio Val	Seattle	WA	98000	Save	Delete
View	Yvonne	McKay (sample)	555-0156	someone_n@exa	896 Southdale	Monroe	WA	68972	Save	Delete

Steps

SDK Sample

This lab is taken from the SDK, Sample: OData endpoint jQuery contact editor

Files can be found in C:\CRM\SDK/SampleCode/JS/RESTEndpoint/RESTJQueryContactEditor and F:\Labs\Module7

Create Web Resources

This sample uses the following web resources:

- sample_/RESTJQueryContactEditor.htm – An HTML file that references all the needed JScript files and displays output to the user.
- sample_/Scripts/jquery1.4.1.min.js - A popular JScript library that includes capabilities to perform asynchronous data operations using the $.ajax object. jQuery1.4.1.js comes standard in a Visual Studio 2010 Web Application.
- sample_/Scripts/json2.js - a public domain library created by Douglas Crockford and distributed from http://www.JSON.org/json2.js. This file has been minified and all comments are removed. See the original file to view information about the JSON.stringify and JSON.parse methods this library provides.
- sample_/Scripts/RESTJQueryContactEditor.js – A generic library that can perform Create, Retrieve, Update and Delete operations using jquery1.4.1.min.js.

Import the managed solution RESTJQueryContactEditor_1_0_0_1_managed.zip

Review Code

Review the sample_/RESTJQueryContactEditor.htm and sample_/Scripts/RESTJQueryContactEditor.js files either from Default Solution or by opening Visual Studio solution in F:\Labs\Module7.

Test

Open the sample_/RESTJQueryContactEditor.htm Web Resource and click Preview

Lab 8A - SiteMap

Objective

To explore how SiteMap changes

Steps

Create Solution

Create new Solution called SiteMap

Add Component

Under Client Extensions, select Add Existing and then SiteMap

Export Solution

Export solution to ZIP file

Make safe copy of ZIP file

Edit SiteMap

Extract customizations.xml from ZIP file

Edit customizations.xml

Swap Areas around

Add customizations.xml back into ZIP file

Import Solution

Import solution from ZIP file

Refresh CRM

Press Ctrl-F5 to force refresh