04 de Outubro de 2016

**Exercicio 13 – Automação (Custom Workflow)**

Exercicio 13 – Automação

Objectivo:

* Criar de Custom Workflow Activity

Tarefas:

1. Criar Custom Workflow Activity
2. Publicar o Custom Activity
3. Usar num workflow

Instruções:

1. Criar Custom Workflow Activity
2. Para criar um custom workflow activity e exeuctar no CRM teremos que ter o Visual Studio e o SDK de CRM. Download: [SDK](https://www.microsoft.com/en-us/download/details.aspx?id=50032), [Visual Studio](https://www.visualstudio.com/)
3. Já tendo o Visual Studio instalado, vamos criar um projecto indo a File->New Project e depois escolher Activity Library e dar o nome de OnUpdateEntity e clicar Ok.
4. Nas references adicionar as seguintes dll’s: Microsoft.Xrm.Sdk e Microsoft.Xrm.Sdk.Workflow.dll da pasta SDK\Bin
5. Remover o ficheiro Activity1.xaml e adicionar uma nova class (.cs) com o nome OnUpdateEntity.cs
6. Depois mudar a o nome da class, extendemos a class adicionando à frente do nome : CodeActivity e adicionando os usings (using System.Activities; using Microsoft.Xrm.Sdk; using Microsoft.Xrm.Sdk.Workflow;).
7. É necessário também assinar o projecto para podermos publicar, indo às propriedades do projecto e criar uma chave para o projecto.
8. De seguida pode-se usar o seguinte código para executar a acção pretendida:

protected override void Execute(CodeActivityContext executionContext)

{

//Create the tracing service

ITracingService tracingService = executionContext.GetExtension<ITracingService>();

//Create the context

IWorkflowContext context = executionContext.GetExtension<IWorkflowContext>();

IOrganizationServiceFactory serviceFactory = executionContext.GetExtension<IOrganizationServiceFactory>();

IOrganizationService service = serviceFactory.CreateOrganizationService(context.UserId);

tracingService.Trace("Update Opportunity");

try

{

EntityReference acc = new EntityReference();

Entity opp = getOpportunityById(OpportunityReference.Get<EntityReference>(executionContext).Id, service);

if (opp.Attributes.Contains("parentaccountid"))

{

acc = ((EntityReference)opp.Attributes["parentaccountid"]);

EntityReference user = getUserFromAccount(acc.Id, service);

if (user != null)

if (user.Id != Guid.Empty)

if (!opp.Attributes.Contains("new\_opportunitymanagerid"))

opp.Attributes.Add("new\_opportunitymanagerid", user);

else

opp.Attributes["new\_opportunitymanagerid"] = user;

service.Update(opp);

Updated.Set(executionContext, true);

}

}

catch (Exception ex)

{

throw new InvalidPluginExecutionException("Error: " + ex.Message);

}

}

private Entity getOpportunityById(Guid oppId, IOrganizationService service)

{

string fetch = @"<fetch version='1.0' output-format='xml-platform' mapping='logical' distinct='false'>

<entity name='opportunity'>

<attribute name='parentaccountid' />

<attribute name='new\_opportunitymanagerid' />

<filter type='and'>

<condition attribute='opportunityid' operator='eq' value='" + oppId + @"' />

</filter>

</entity>

</fetch>";

EntityCollection result = service.RetrieveMultiple(new FetchExpression(fetch));

if (result != null & result.Entities != null)

if (result.Entities.Count > 0)

{

if (result.Entities.Count == 1)

return ((Entity)result.Entities[0]);

else

return null;

}

else

return null;

else

return null;

}

private EntityReference getUserFromAccount(Guid accountId, IOrganizationService service)

{

string fetch = @"<fetch version='1.0' output-format='xml-platform' mapping='logical' distinct='false'>

<entity name='account'>

<attribute name='ownerid' />

<filter type='and'>

<condition attribute='accountid' operator='eq' value='" + accountId + @"' />

</filter>

</entity>

</fetch>";

EntityCollection result = service.RetrieveMultiple(new FetchExpression(fetch));

if (result != null & result.Entities != null)

if (result.Entities.Count > 0)

{

if (result.Entities.Count == 1)

return ((EntityReference)result.Entities[0].Attributes["ownerid"]);

else

return null;

}

else

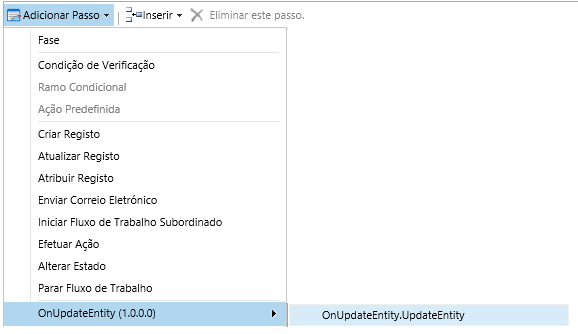
return null;

else

return null;

}

1. Publicar o Custom Workflow
   1. Ir a SDK\Tools\PluginRegistration e abrir o executavel do PluginRegistration.exe
   2. Clicar em Create New Organization e fazer login com o Office 365, escolher Europe, Middle East and Asia e colocar as credenciais de acesso. Clicar em Login.
   3. De seguida clicar em Registar e escolher Register New Assembly, clicar no botão “...” e ir até à pasta do projecto à pasta bin\Debug\ e escolher a dll do projecto. Clicar em Register Selected Plugins.
2. Usar num workflow
3. Para usar este desenvolvimento tem que ser criado um Workflow e pode-se faze-lo de duas formas, dentro da solução e ir a Process ou ir a Settings-> Process.
4. Tendo aberta a janela de workflow, podemos criar a nossa lógica e podemos usar o desenvolvimento pois está disponivel nas opções de componentes:



1. De seguida pode-se contruir o Workflow da seguinte forma:

