Module 3 Lab A

Creating Custom Entities

Objective

Develop a MVC Web App that allows users to create new Entity types

Scenario

The finished Web App will allow a user to create an Entities and add multiple string based attributes to the entities created

Step1: Editing Model classes to support the metadata needed to create the Entity and Attributes

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

1. Open the Visual Studio Project in the Module 3 Lab A starter folder. This project is MVC based the Controller has methods Create and Update that will allow the creation and updating of entities.
2. Open the CustomEntityMetadata.cs file in the models folder and create 7 string properties named.

EntityName

EntityDisplayName

EntityDisplayCollectionName

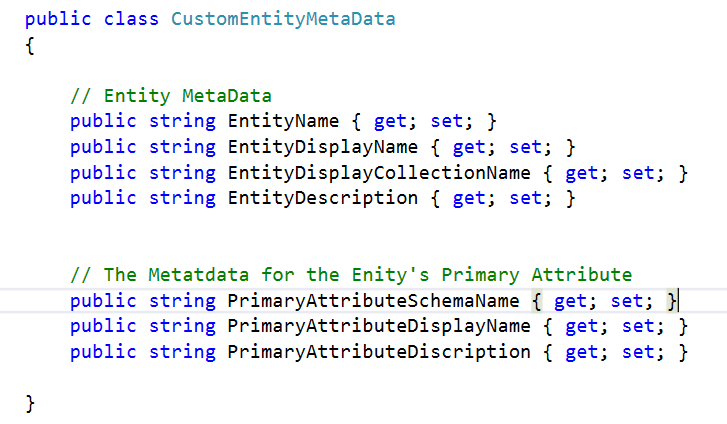
EntityDescription

PrimaryAttributeSchemaName

PrimaryAttributeDisplayName

PrimaryAttributeDiscription

The first 4 will be associated directly with the new Entity the following 3 with it's primary attribute.

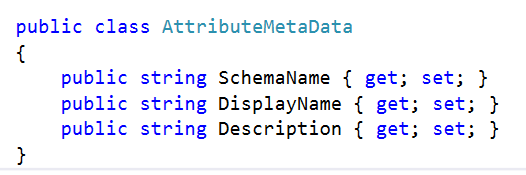


1. Edit the class in the AttributeMetaData.cs adding three string attributes named

SchemaName

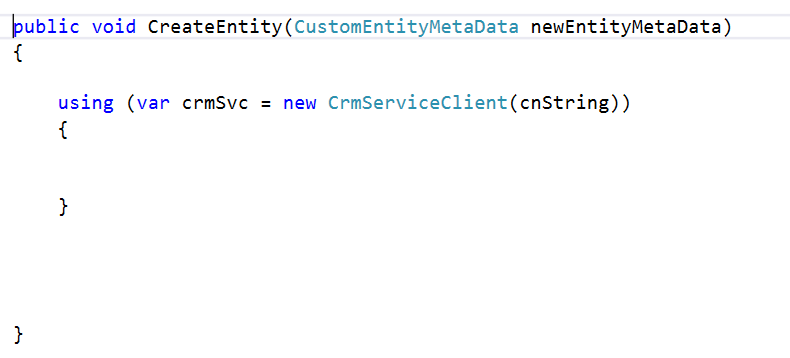
DisplayName

Description

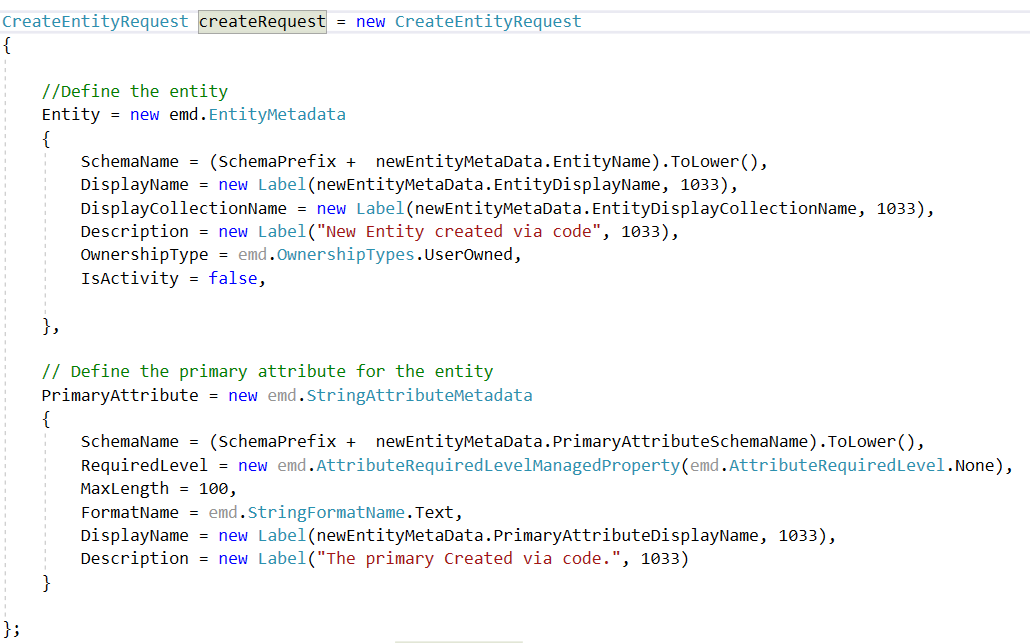


Exercise 2 The EntityMetaDataUtlis.cs file will implement the logic that creates and updates the Entities. This class will be called from the controller. We will now implement the Create and Update methods.

1. Modify the connection string named CrmOnline so that it is valid for your Dynamics 365 organisation. (you can copy it from the previous lab).
2. Within the Class you the following method.

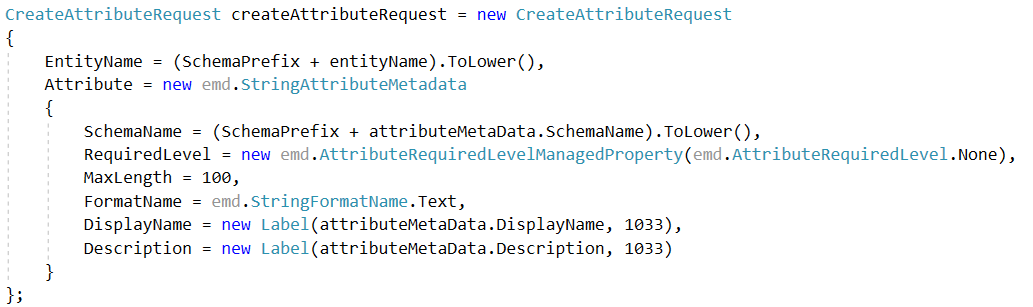


1. Inside the using statement Create a new instance of a CreateEntityRequest object and assign it to a variable named createRequest.
2. Initialise the object as shown below, note that the properties named Entity and PrimaryAttribute are both being set. The Custom Entity Metadata is passed through a parameter, named newEntityMetaData, to the CreateEntity method.



The Schema Prefix is declared already as a constant and will need to be modified if your schema name is different.

1. Following this call the crmSvc.OrganizationServiceProxy.Execute method passing the createRequest variable as the parameter value.
2. The UpdateEntity method will be called to add attributes to an existing entity. For simplicity we have assumed that the attributes' datatype will be limited to text. Modify the method so that within the using statement, an object of type CreateAttributRequest is created and assigned to variable named createAttributeRequest. Initialise the Entity and Attribute properties as shown below.



1. Following this call the crmSvc.OrganizationServiceProxy.Execute method passing the createAttributeRequest variable as the parameter value.
2. The changes need to be publ ished so following the call in step 10 create an instance to the PublishAllXmlRequest object and then pass this to the crmSvc.OrganizationServiceProxy.Execute method.
3. The Controlllers folder in the project has a controller named EntityMetaDataController created for you that calls the CreateEntity and Update entity methods for you. Within the CreateEntity Method, uncomment the line // Session["EntityName"] = newEntityMetaData.EntityName;
4. Views for this controller have been created in the appropriate views folder. Open these files and have a look at the code that has been provided noting how calls to the methods have been implemented.
5. Run the application and test it by creating an entity and adding 1 or more attributes to it using the app.