Dynamic Apex

swdcworld.com

Dynamic Apex Useful Methods and check CRUD, FLS Permissions

1 Comment Thursday, 12 January 2017

Useful Schema Describe Methods & check CRUD, FLS

Dynamic apex enables developer to develop more flexible applications using describe information about the sobjects, their fields and child relationship information. Now here I'm going to show some useful methods used with Schema.DescribeSobject.

IsCustom() :

some time we need to check the object is Custom object or Standard object. It's difficult to find out. Now it's simple with **IsCustom()** method, we need to pass object name and it will return boolean true or false. See following sample code.

```
List<String> allCustomObjsList = new List<string>();
List<String> allStandardObjsList = new List<string>();
for(Schema.SobjectType obj : Schema.getGlobalDescribe().values())
    {
        Schema.DescribeSobjectResult objResult = obj.getDescribe();
          if(objResult.isCustom())
          {
              String strObjName =objResult.getname();
              allCustomObjsList.add(strObjName);
          }
        else{
             String strObjName =objResult.getname();
              allStandardObjsList.add(strObjName);
        }
    }
System.debug('all Custom Objects List'+allCustomObjsList);
System.debug('all Standard Objects List'+allStandardObjsList);
```

Above code get the all the objects in org, then separate custom objects into allCustomObjsList and Standart objects into allStandardObjsList .

IsQueryable() :

Some of the standard salesforce objects we can't query. For example take **EmailStatus** standard object, If you try to query this it will throw error message like "**entity type EmailStatus does not support query**".

Check the following code, It will return the soql query supported objects into one list and soql query not supported objects into another list.

```
List<String> queryCan = new List<string>();
List<String> queryCant = new List<string>();
for(Schema.SobjectType obj : Schema.getGlobalDescribe().values())
    {
        Schema.DescribeSobjectResult objResult = obj.getDescribe();
          if(objResult.IsQueryable())
          {
              String strObjName =objResult.getname();
              queryCan.add(strObjName);
          }
        else{
            String strObjName =objResult.getname();
            queryCant.add(strObjName);
        }
    }
System.debug('soql query supported objects'+queryCan);
System.debug('soql query Not supported objects'+queryCant);
```

If want check the single object use following piece of code.

```
Schema.DescribeSObjectResult describeAccount = Schema.sObjectType.Account;
System.debug('check Account is queryable: '+describeAccount.IsQueryable());
```

The output is boolean.

```
USER_DEBUG [2]|DEBUG|check Account is queryable: true
```

Dynamic Apex to Check the CRUD Permissions on Object:

Dynamic apex is very useful to check the CRUD permissions (Create / Read / Update / Delete permissions) on any object for current user. see following methods, all are boolean methods. It will written true or false.

- isCreateable() : Returns true if the object can be created by the current user, false otherwise.
- isAccessible() : Returns true if the current user can see this object, false otherwise.
- isUpdateable() : Returns true if the object can be updated by the current user, false otherwise.
- isDeletable() : Returns true if the object can be deleted by the current user, false otherwise.

```
Schema.DescribeSObjectResult describeAccount = Schema.sObjectType.Account;
System.debug('isCreateable: '+describeAccount.isCreateable());
System.debug('isAccessible: '+describeAccount.isAccessible());
System.debug('isUpdateable: '+describeAccount.isUpdateable());
System.debug('isDeletable: '+describeAccount.isDeletable());
```

Output:

```
USER_DEBUG [2]|DEBUG|isCreateable: true
USER_DEBUG [3]|DEBUG|isAccessible: true
USER_DEBUG [4]|DEBUG|isUpdateable: true
USER_DEBUG [5]|DEBUG|isDeletable: true
```

we can use in apex code to check the object-level permission for the contact before deleting the contact.

```
if (Schema.sObjectType.Contact.isDeletable()) {
   // Delete contact code here
}
```

Dynamic Apex to Check FLS Permissions:

We can even also check FLS Permissions(Field level Security permissions) using the Dynamic apex **Schema.DescribeFieldResult .** Following methods are all instance methods. It will return boolean type.

- isAccessible() : Returns true if the current user can see this object, false otherwise.
- isCreateable() : Returns true if the object can be created by the current user, false otherwise.
- isUpdateable() : Returns true if the object can be updated by the current user, false otherwise.

```
Schema.DescribeFieldResult describeFieldIndustry = Account.Type.getDescribe();
System.debug('isAccessible: '+describeFieldIndustry.isAccessible());
System.debug('isCreateable: '+describeFieldIndustry.isCreateable());
System.debug('isUpdateable: '+describeFieldIndustry.isUpdateable());
```

we can use To check the field-level create permission of the account's Industry field before creating a new account:

```
if (Schema.sObjectType.Account.fields.Industry.isCreateable()) {
   // Create new account code here
}
```