SoqlStubProvider Class

Contains a method to create a mock test class for handling SOQL query responses for Data Cloud data model objects (DMOs).

Namespace

System

Usage

To create mock test classes, extend the `SoqlStubProvider` class and override the `handleSoqlQuery()` class method.

> **Note**
>
> SOQL `For` Loops in Apex aren't supported for SOQL stubs in static or dynamic SOQL queries against DMOs.

See Mock SOQL Tests for Data Cloud Data Model Objects in the *Apex Developer Guide*.

Example

This example shows a mock test class for the *SkyMilesForBusinessOptInController* class.

```
@IsTest
public class SkyMilesForBusinessOptInController_Test {

    @IsTest
    public static void mockSoql() {

        SoqlStubProvider stub = new UnifiedIndividualSoqlStub();
        Test.createSoqlStub(UnifiedIndividual__dlm.sObjectType, stub);

        Assert.isTrue(Test.isSoqlStubDefined(UnifiedIndividual__dlm.sObjectType));

        Test.startTest();
        string companyId = 'SampleCompanyId';
        // Performs SOQL query against Data Model Object
        List<SkyMilesMember> members = SkyMilesForBusinessOptInController.getSkyMilesProfi

        Test.stopTest();

        Assert.areEqual(1, members.size());

        SkyMilesMember member = members[0];

        Assert.areEqual(companyId, member.CompanyId);
        Assert.areEqual(5000, member.SkyMilesBalance);
    }

    class UnifiedIndividualSoqlStub extends SoqlStubProvider {
        public override List<sObject> handleSoqlQuery(sObjectType sot, string stubbedQuery
```

```
            companyId = (string)bindVars.get('tmpVar1');
        }

        UnifiedIndividual__dlm dmo = (UnifiedIndividual__dlm)Test.createStubQueryRow(
            sot,
            new Map<string, object> {
                'ssot__FirstName__c' => 'Codey',
                'ssot__LastName__c' => 'Bear',
                'ssot__Email__c' => 'developer@salesforce.com',
                'ssot__SkyMilesBalance__c' => 5000,
                'ssot__MedallionStatus__c' => 'Gold',
                'ssot__CompanyId__c' => companyId
            }
        );
        return new List<sObject> { dmo };

    }
  }
}
```

```
public with sharing class SkyMilesForBusinessOptInController {
    public static List<SkyMilesMember> getSkyMilesProfilesFromDataCloud(String companyId)
        List<UnifiedIndividual__dlm> unifiedIndividuals = [
          SELECT
            Id,
            ssot__FirstName__c,
            ssot__LastName__c,
            ssot__Email__c,
            ssot__SkyMilesBalance__c,
            ssot__MedallionStatus__c,
            ssot__CompanyId__c
          FROM UnifiedIndividual__dlm
          WHERE ssot__CompanyId__c = :companyId
        ];
        List<SkyMilesMember> skyMilesMembers = new List<SkyMilesMember>();
        for (UnifiedIndividual__dlm individual : unifiedIndividuals) {
          skyMilesMembers.add(
            new SkyMilesMember(
              individual.Id,
              individual.ssot__FirstName__c,
              individual.ssot__LastName__c,
              individual.ssot__Email__c,
              individual.ssot__SkyMilesBalance__c,
              individual.ssot__MedallionStatus__c,
              individual.ssot__CompanyId__c
            )
          );
        }
        return skyMilesMembers;
    }
}
```

- SoqlStubProvider Methods

SoqlStubProvider Methods

The following are methods for `SoqlStubProvider`.

- **handleSoqlQuery(targetType, stubbedQuery, bindMap)**
  Defines a mocked response for a SOQL query executed against the specified SObject type.

```
public List<SObject> handleSoqlQuery(Schema.SObjectType targetType, String stubbedQuery,
Map<String,Object> bindMap)
```

Parameters

targetType

Type: Schema.SObjectType

The SObject type to be stubbed. This parameter can't be null.

stubbedQuery

Type: String

The SOQL query whose response is to be stubbed. Bind variables are replaced with placeholders.

bindMap

Type: Map<String,Object>

A map that contains placeholder keys for each bind variable specified in the SOQL query string and its value.

Return Value

Type: List<SObject>

The list of stubbed SObjects resulting from the SOQL query.

See Also

- Test Class
- *Apex Developer Guide:* Mock SOQL Tests for Data Cloud Data Model Objects