Security Class

Contains methods to securely implement Apex applications.

Namespace

System

Usage

In the context of the current user's create, read, update, or upsert access permission, use the Security class methods to:

- Strip fields that aren't visible from query and subquery results
- Remove inaccessible fields before a DML operation without causing an exception
- Sanitize SObjects that have been deserialized from an untrusted source
- Security Methods

Security Methods

The following are methods for `Security`.

- **stripInaccessible(accessCheckType, sourceRecords, enforceRootObjectCRUD)**
  Creates a list of sObjects from the source records, which are stripped of fields that fail the field-level security checks for the current user. The method also provides an option to enforce an object-level access check.
- **stripInaccessible(accessCheckType, sourceRecords)**
  Creates a list of sObjects from the source records, which are stripped of fields that fail the field-level security checks for the current user.
- **stripInaccessible(accessCheckType, sourceRecords, enforceRootObjectCRUD, permissionSetId)(Developer Preview)**
  Creates a list of sObjects from the source records, which are stripped of fields that fail field-level and object-level access checks. Apex enforces field-level security (FLS) and object permissions as per the specified permission set, in addition to the running user's permissions.

stripInaccessible(accessCheckType, sourceRecords, enforceRootObjectCRUD)

Creates a list of sObjects from the source records, which are stripped of fields that fail the field-level security checks for the current user. The method also provides an option to enforce an object-level access check.

Signature

```
public static System.SObjectAccessDecision stripInaccessible(System.AccessType accessCheckType,
List<SObject> sourceRecords, Boolean enforceRootObjectCRUD)
```

Parameters

accessCheckType

sourceRecords

Type: List<SObject>

A list of sObjects to be checked for fields that aren't accessible in the context of the current user's operation.

enforceRootObjectCRUD

Type: Boolean

Indicates whether an object-level access check is performed. If this parameter is set to `true` and the access check fails, the method throws an exception. The default value of this optional parameter is `true`.

Return Value

Type: System.SObjectAccessDecision

Example

In this example, the user doesn't have permission to create the `Probability` field of an Opportunity.

```
List<Opportunity> opportunities = new List<Opportunity>{
    new Opportunity(Name='Opportunity1'),
    new Opportunity(Name='Opportunity2', Probability=95)
};

// Strip fields that are not creatable
SObjectAccessDecision decision = Security.stripInaccessible(
    AccessType.CREATABLE,
    opportunities);

// Print stripped records
for (SObject strippedOpportunity : decision.getRecords()) {
    System.debug(strippedOpportunity);
}

// Print modified indexes
System.debug(decision.getModifiedIndexes());

// Print removed fields
System.debug(decision.getRemovedFields());

//Lines from output log
//|DEBUG|Opportunity:{Name=Opportunity1}
//|DEBUG|Opportunity:{Name=Opportunity2}
//|DEBUG|{1}
//|DEBUG|{Opportunity={Probability}}
```

stripInaccessible(accessCheckType, sourceRecords)

Creates a list of sObjects from the source records, which are stripped of fields that fail the field-level security checks for the current user.

Signature

```
public static System.SObjectAccessDecision stripInaccessible(System.AccessType accessCheckType,
List<SObject> sourceRecords)
```

Parameters

accessCheckType

Type: System.AccessType

Type: List<SObject>

A list of sObjects to be checked for fields that aren't accessible in the context of the current user's operation.

Return Value

Type: System.SObjectAccessDecision

Example

In this example, the user doesn't have permission to read the `ActualCost` field of a Campaign.

```
List<Campaign> campaigns = new List<Campaign>{
    new Campaign(Name='Campaign1', BudgetedCost=1000, ActualCost=2000),
    new Campaign(Name='Campaign2', BudgetedCost=4000, ActualCost=1500)
};
insert campaigns;

// Strip fields that are not readable
SObjectAccessDecision decision = Security.stripInaccessible(
    AccessType.READABLE,
    [SELECT Name, BudgetedCost, ActualCost from Campaign]);

// Print stripped records
for (SObject strippedCampaign : decision.getRecords()) {
    System.debug(strippedCampaign); // Does not display ActualCost
}

// Print modified indexes
System.debug(decision.getModifiedIndexes());

// Print removed fields
System.debug(decision.getRemovedFields());

//Lines from output log
//|DEBUG|Campaign:{Name=Campaign1, BudgetedCost=1000, Id=701xx00000011nhAAA}
//|DEBUG|Campaign:{Name=Campaign2, BudgetedCost=4000, Id=701xx00000011niAAA}
//|DEBUG|{0, 1}
//|DEBUG|{Campaign={ActualCost}}
```

stripInaccessible(accessCheckType, sourceRecords, enforceRootObjectCRUD, permissionSetId)(Developer Preview)

Creates a list of sObjects from the source records, which are stripped of fields that fail field-level and object-level access checks. Apex enforces field-level security (FLS) and object permissions as per the specified permission set, in addition to the running user's permissions.

Signature

> **Note**
>
> Feature is available as a developer preview. Feature isn't generally available unless or until Salesforce announces its general availability in documentation or in press releases or public statements. All commands, parameters, and other features are subject to change or deprecation at any time, with or without notice. Don't implement functionality developed with these commands or tools in a production environment. You can provide feedback and suggestions for the "Permission Sets with User Mode" feature in the Trailblazer Community.

Parameters

accessCheckType

Type: System.AccessType

Uses values from the AccessType enum. This parameter determines the type of field-level access check to be performed. To check the current user's field-level access, use the Schema.DescribeFieldResult methods – `isCreatable()`, `isAccessible()`, or `isUpdatable()`.

sourceRecords

Type: List<SObject>

A list of sObjects to be checked for fields that aren't accessible in the context of the current user's operation.

enforceRootObjectCRUD

Type: Boolean

Indicates whether an object-level access check is performed. If this parameter is set to `true` and the access check fails, the method throws an exception. The default value of this optional parameter is `true`.

permissionSetId

Type: Id

Permissions in the specified permission set are enforced in additon to the running user's permissions.

Return Value

Type: System.SObjectAccessDecision