**Metadata\_Config** (Salesforce Application)

Version: 1.0

Date: 9/11/2019

**Overview:**

Salesforce is a technology that stores and accesses customer data in their cloud and analyze their sales by understanding the customer transactions using the data in the cloud. So, when any application get builds in Salesforce before deploying it, developer must check all metadata. For that right now there is very tedious process. This application will help developer to get view of all metadata in single glance and help to update field level security of objects.

**Features**:

         View/ Update Field level securities for selected profiles

         View/ Update Field level securities for selected objects.

         View/ Update Field level securities of managed/ unmanaged package.

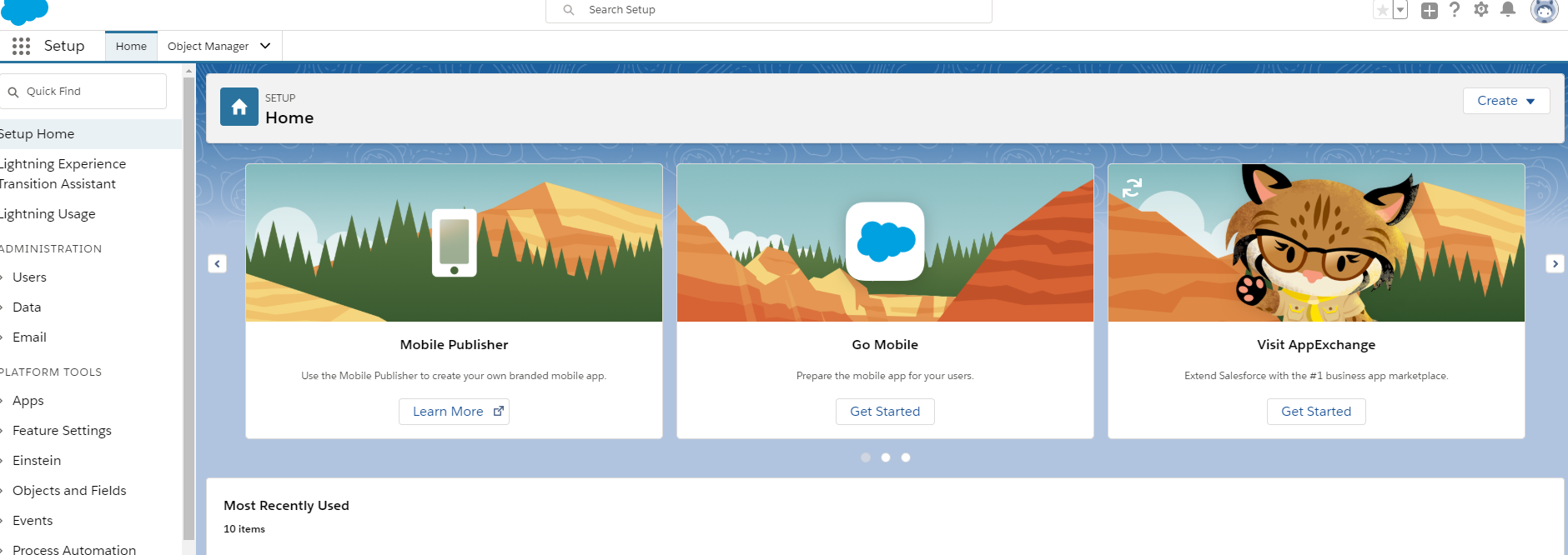
**Technologies**: Salesforce Apex, Salesforce lightning, JavaScript

**How to install application:**

1. As this application uses cloud technology, one should have salesforce org. You can create free salesforce org , follow below steps

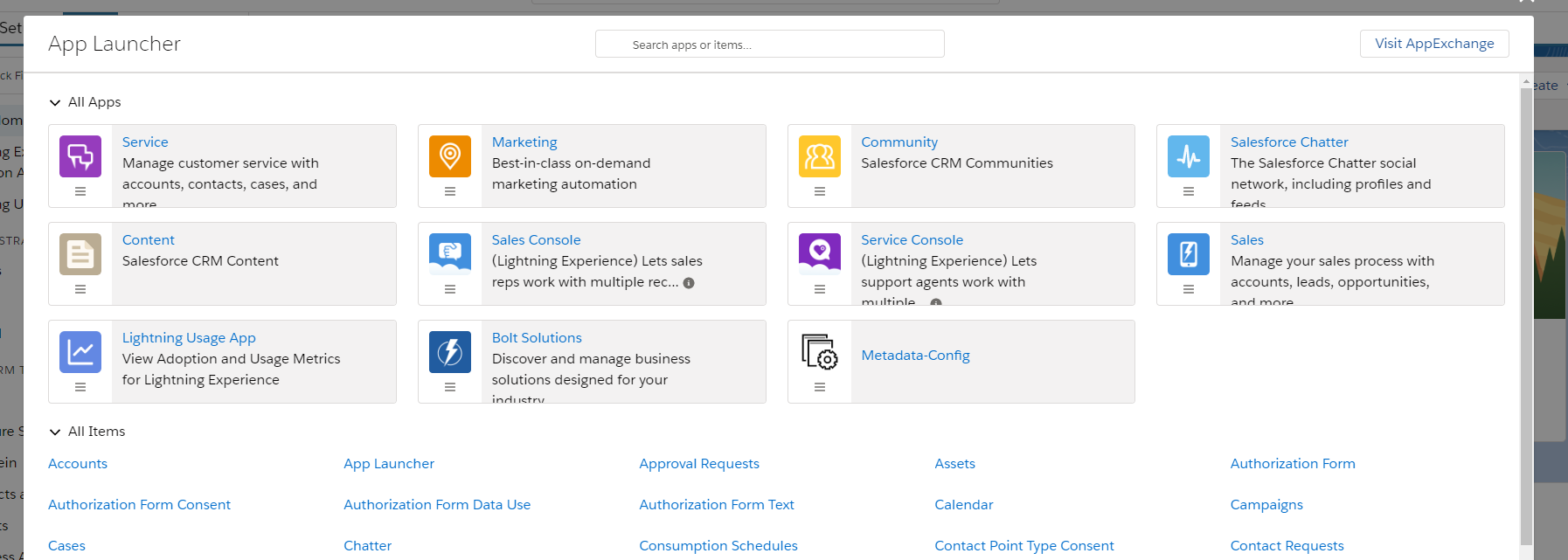
* Go to https://developer.salesforce.com/signup.
* Enter your contact information.
* Enter a unique username.
* Submit the form and wait for your welcome email.
* In the welcome email, click the link to set your password.

1. Install this application using below link. Below link will install package in your org.
2. After successful installation, go to home page of your org and click on ‘App Launcher’ located at upper left corner



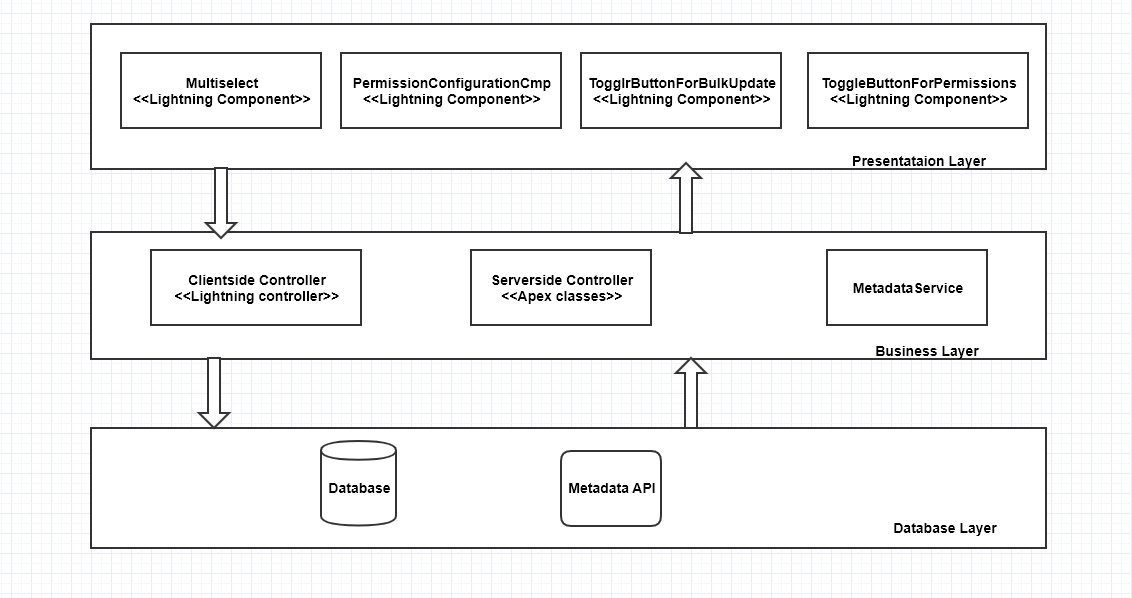


1. After that in ‘All -apps’ you can find the ‘Metadata\_config’ app.





**Architectural Diagram:**

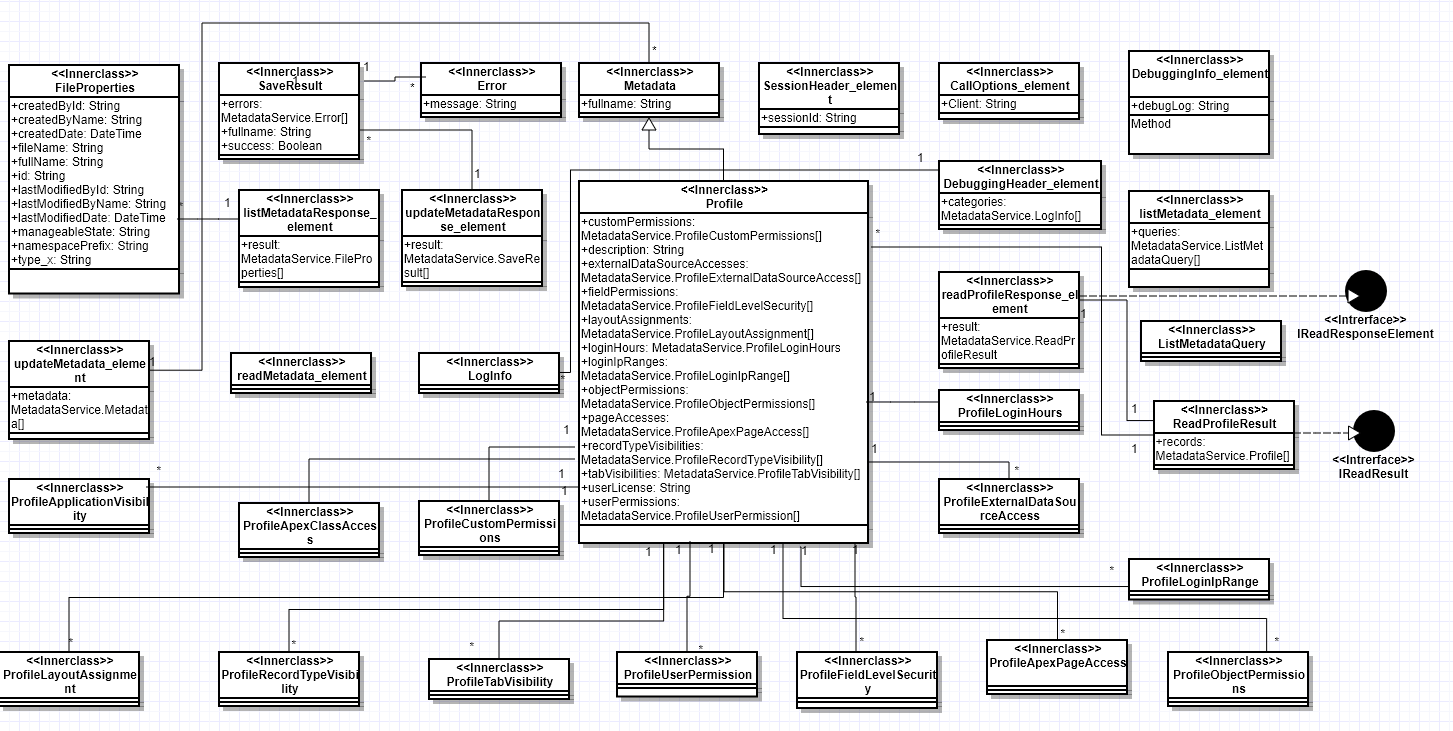


\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

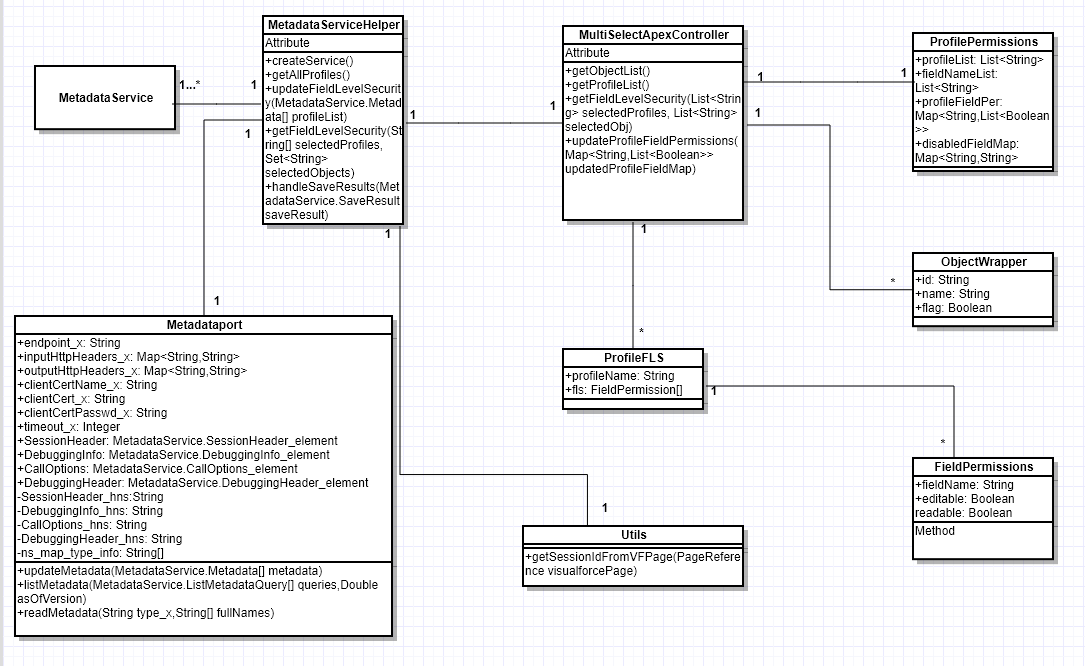
**UML Diagrams:**

**Class diagram for MetadataService class**

MetadataService class is having following inner classes

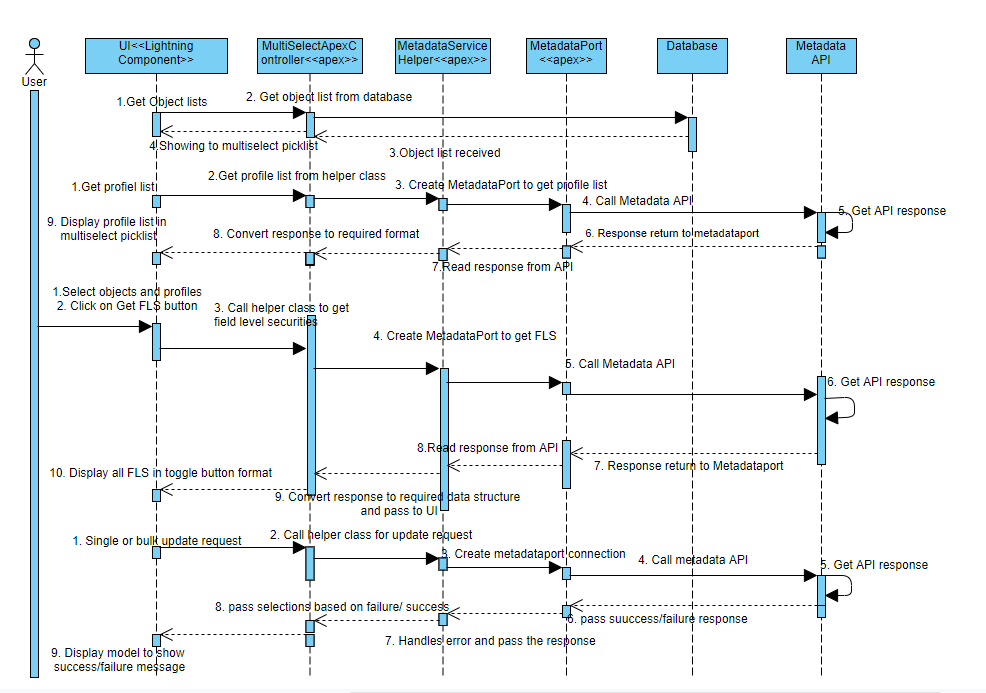


**Class diagram of application**



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Sequence Diagram:**



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Use Cases:**

|  |  |
| --- | --- |
| Use Case 1 | Select Objects |
| Actor | Admin/ User |
| Precondition | One should login to salesforce org with valid credentials and open ‘Metadata\_Config’ app. |
| Basic Flow | 1. User can select objects form multi pick list box.  2. User can select max 5 objects from multi pick list |
| Alternative Flow | If users select more than 5 objects, then message will appear saying ‘max limit is 5’. |

|  |  |
| --- | --- |
| Use Case 2 | Select Profiles |
| Actor | Admin/ User |
| Precondition | One should login to salesforce org with valid credentials and open ‘Metadata\_Config’ app. |
| Basic Flow | 1. User can select profiles form multi pick list box.  2. User can select max 5 profiles from multi pick list |
| Alternative Flow | If users select more than 5 profiles, then message will appear saying ‘max limit is 5’. |

|  |  |
| --- | --- |
| Use Case 3 | Click on ‘Get FLS’ button |
| Actor | Admin/ User |
| Precondition | 1. One should login to salesforce org with valid credentials and open ‘Metadata\_Config’ app.  2. Select objects and profiles up to max 5. |
| Basic Flow | 1. After clicking on ‘Get FLS’ button filed permissions for selected profiles and objects will display. |
| Alternative Flow | If user have not selected any object/ profile then pop up message will come saying ‘please select profile/object’. |

|  |  |
| --- | --- |
| Use Case 5 | Changing field permission one at a time |
| Actor | Admin/ User |
| Precondition | 1. One should login to salesforce org with valid credentials and open ‘Metadata\_Config’ app.  2. Select objects and profiles up to max 5 and click on ‘Get FLS’ button |
| Basic Flow | 1. Every field of selected object and selected profile is having two permissions ‘Read’, ‘Edit’ is toggle button format.  2. If User changes field permission of specific field then toggle update will change as per modifications |
| Alternative Flow | 1. If user disables ‘read’ permission of field and the same field is having ‘edit’ permission enabled, then edit will also get disabled.  2. If user enables ‘edit’ permission of any field and the same field is having ‘read’ permission disabled, then read permission will also get assign. |

Every field of selected object and selected profile is having two permissions ‘Read’, ‘Edit’ in toggle button format.

Also there will be some toggle buttons to do bulk changes. User can see following bulk toggle buttons

* **Bulk Toggle Button 1-** ‘Read’: to change permission of all fields of specific object
* **Bulk Toggle Button 2-** ‘Edit’: to change permission of all fields of specific object
* **Bulk Toggle Button 3-** ‘Read’: to change permission of field for all selected profiles
* **Bulk Toggle Button 4-** ‘Edit: to change permission of field for all selected profiles
* **Bulk Toggle Button 5-** ‘Read’: to change permission of all fields for selected profiles and objects
* **Bulk Toggle Button 6-** ‘Edit: to change permission of all fields for selected profiles and objects

|  |  |
| --- | --- |
| Use Case 6 | Changing **Bulk Toggle Button 1-** ‘Read’: to change permission of all fields of specific object |
| Actor | Admin/ User |
| Precondition | 1. One should login to salesforce org with valid credentials and open ‘Metadata\_Config’ app.  2. Select objects and profiles up to max 5 and click on ‘Get FLS’ button |
| Basic Flow | User can change Bulk Toggle Button 1 (‘Read’: to change permission of all fields of specific object) |
| Alternative Flow | 1. If user disables Bulk Toggle Button 1 then edit permission (if present) will also lost. |

|  |  |
| --- | --- |
| Use Case 7 | Changing **Bulk Toggle Button 2-** ‘Edit’: to change permission of all fields of specific object |
| Actor | Admin/ User |
| Precondition | 1. One should login to salesforce org with valid credentials and open ‘Metadata\_Config’ app.  2. Select objects and profiles up to max 5 and click on ‘Get FLS’ button |
| Basic Flow | User can change Bulk Toggle Button 2 ‘Edit’: to change permission of all fields of specific object) |
| Alternative Flow | If user enables Bulk Toggle Button 2 then read permission (if not present) will also get assign. |

|  |  |
| --- | --- |
| Use Case 8 | Changing **Bulk Toggle Button 3-** ‘Read’: to change permission of field for all selected profiles |
| Actor | Admin/ User |
| Precondition | 1. One should login to salesforce org with valid credentials and open ‘Metadata\_Config’ app.  2. Select objects and profiles up to max 5 and click on ‘Get FLS’ button |
| Basic Flow | User can change Bulk Toggle Button 3 (‘Read’: to change permission of field for all selected profiles) |
| Alternative Flow | If user disables Bulk Toggle Button 3 then edit permission (if present) will also lost. |

|  |  |
| --- | --- |
| Use Case 9 | Changing **Bulk Toggle Button 4-** ‘Edit: to change permission of field for all selected profiles |
| Actor | Admin/ User |
| Precondition | 1. One should login to salesforce org with valid credentials and open ‘Metadata\_Config’ app.  2. Select objects and profiles up to max 5 and click on ‘Get FLS’ button |
| Basic Flow | User can change Bulk Toggle Button 4 (‘Edit: to change permission of field for all selected profiles) |
| Alternative Flow | If user enables Bulk Toggle Button 4 then read permission (if not present) will also get assign. |

|  |  |
| --- | --- |
| Use Case 10 | Changing **Bulk Toggle Button 5-** ‘Read’: to change permission of all fields for selected profiles and objects |
| Actor | Admin/ User |
| Precondition | 1. One should login to salesforce org with valid credentials and open ‘Metadata\_Config’ app.  2. Select objects and profiles up to max 5 and click on ‘Get FLS’ button |
| Basic Flow | User can change Bulk Toggle Button 5 ‘Read’: to change permission of all fields for selected profiles and objects) |
| Alternative Flow | If user disables Bulk Toggle Button 5 then edit permission (if present) will also lost. |

|  |  |
| --- | --- |
| Use Case 11 | Changing **Bulk Toggle Button 6-** ‘Edit’: to change permission of all fields for selected profiles and objects |
| Actor | Admin/ User |
| Precondition | 1. One should login to salesforce org with valid credentials and open ‘Metadata\_Config’ app.  2. Select objects and profiles up to max 5 and click on ‘Get FLS’ button |
| Basic Flow | User can change Bulk Toggle Button 6 (‘Edit’: to change permission of all fields for selected profiles and objects) |
| Alternative Flow | If user enables Bulk Toggle Button 6 then read permission (if not present) will also get assign. |

|  |  |
| --- | --- |
| Use Case 12 | Updating the changed field permissions |
| Actor | Admin/ User |
| Precondition | 1. One should login to salesforce org with valid credentials and open ‘Metadata\_Config’ app.  2. Select objects and profiles up to max 5 and click on ‘Get FLS’ button  3. Change one or more field permission. |
| Basic Flow | 1.User can update the changes in org by clicking ‘Update’ button.  2. After clicking on update button it will update the changes in backend and will make new API call to show latest updated field permissions on screen. |
| Alternative Flow | If user do not make any changes in field permission, then ‘Update’ button will not appear. |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Code Documentation:**

**Lightning Components (UI):**

Every lightning component consist of component, controller, helper, style, documentation, renderer, design, SVG



**1.PermissionConfigurationCmp:**

This is the main component of this application to show user interface.

From this component other component are called.

It contains lightning layout, in first layout item it contains picklist for objects, profiles and ‘Get FLS’ button. Second layout item contains table structure to show field permissions.

After clicking ‘Get FLS’ its helper class calls the apex class. Below is code snippet for the same.



**2.** **ToggleButtonForpermissions**:

Toggle button for all field permissions are created from this component.



**3.ToggleButtonForBulkUpdate:**

Toggle button for all bulk field permissions are created from this component.



**4.MultiSelect**

Lightning don’t have any input field for multiselect picklist drop box. This component is created for customized multi select pick list using buttons.

**Apex Classes (Backend):**

**1.MultiSelectApexController**

This is client server-side controller for lightning component and it is responsible to give all required data to UI. It contains following methods:

getObjectList():

returns: List<ObjectWrapper>

This method fetch all object list of org from database.

getProfileList():

returns: List<ObjectWrapper>

This method calls MetadataServiceHelper class to retrieve all profiles of org.

getFieldLevelSecurity():

Parameters: List<String> selectedProfiles, List<String> selectedObj

returns: ProfilePermissions

This method calls MetadataServiceHelper class to get field permissions and returns

the restructured output. Also, it gives the list of fields whose field permissions are non editable.

updateProfileFieldPermissions():

Parameters: Map<String,List<Boolean>> updatedProfileFieldMap

Returns: Boolean

This method pass the field permission to be updated to MetadataServiceHelper class and

return the update result true/false.

**2.MetadataServiceHelper**

This class create Metadataport class instance and call Metadata API for various actions like update metadata, listMetadata etc.

createService():

returns: MetadataPort

This method create instance of MetadataPort class and assign sessionId to it.

getAllProfiles():

returns: List<String>

This method gives list of all profiles by calling listmetadata method of MeatadataPort.

updateFieldLevelSecurity()

parameters: MetadataService.Metadata[] profileList

returns: Boolean

This method call updateMetadata method of MeatadataPort.

getFieldLevelSecurity()

parameters: String[] selectedProfiles, Set<String> selectedObjects

returns: MetadataService.Profile[]

This method call ReadMetadata method of MeatadataPort.

handleSaveResults():

parameters: MetadataService.SaveResult saveResult

This method throw exception if result contains any error.

**3.Metadataport**

This class specifies every detail required to webservice call and invokes webservice for different actions

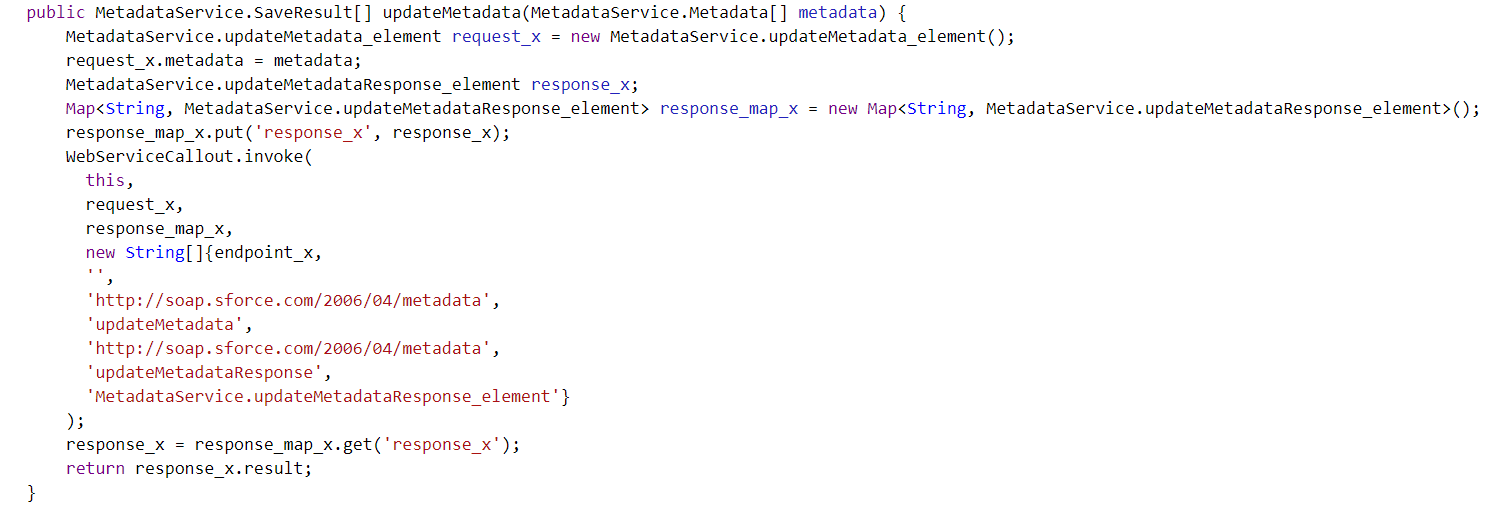


updateMetadata()

parameters: MetadataService.Metadata[] metadata

returns: MetadataService.SaveResult[]

This method makes call to API for updating metadata.



listMetadata()

parameters: MetadataService.ListMetadataQuery[] queries,Double asOfVersion

returns: MetadataService.FileProperties[]

This method make call to API for getting list of all metadata



readMetadata()

parameters: String type\_x,String[] fullNames

returns: IReadResult

This method make call to API for read all metadata



**4.MetadataService**

When we make call to API or when API gives any output, the structure of input and output can be read using this MetadataService class.

This class contains following inner classes:

FileProperties, SaveResult, Error, Metadata, SessionHeader\_element, DebuggingInfo\_element, CallOptions\_element, DebuggingHeader\_element, updateMetadata\_element, updateMetadataResponse\_element, ListMetadataQuery, listMetadata\_element, listMetadataResponse\_element, readMetadata\_element, LogInfo, Profile, ProfileFieldLevelSecurity, ProfileObjectPermissions, ReadProfileResult, readProfileResponse\_element, ProfileApplicationVisibility, ProfileApexClassAccess, ProfileCustomPermissions, ProfileExternalDataSourceAccess, ProfileLayoutAssignment, ProfileLoginHours, ProfileLoginIpRange, ProfileApexPageAccess, ProfileRecordTypeVisibility, ProfileTabVisibility, ProfileUserPermission,

**5.ProfileFLS**

This class contains structure for field level security of profiles.

Parameters: String profileName, FieldPermission[] fls

**6.FieldPermissions**

This class contains structure for individual field permssion

Parameters: String fieldname, Boolean editable, Boolean readable

**7.ObjectWrapper**

This class contains structure for multi pick list drop box values

Parameters: String id, String name, Boolean flag

compareTo()

parameters: Object compareTo

returns: Integer

This method compares two object of type ObjectWrapper and return the output positive/negative/0.

**8.ProfilePermissions**

This class contains structure of profile permissions.

Parameters: List<String> profileList, List<String> fieldNameList, Map<String,List<Boolean>> profileFieldPer, Map<String,String> disabledFieldMap = new Map<String,String>()

**9.Utils:**

This class retrieve session id from visual force page.

getSessionIdFromVFPage()

parameters: PageReference visualforcePage

return: String

This method returns session id of passed visual force page.

**10.IReadResult**

This interface contains only one method. Whoever wants to get records in form of MetadataService.Metadata can implement it.

getRecords():

returns: MetadataService.Metadata[]

**11.IReadResponseElement**

This interface contains only one method. Whoever wants to get records in form of IReadResult can implement it.

getResult ():

returns: IReadResult

**Developer:**

connectojyotsna@gmail.com