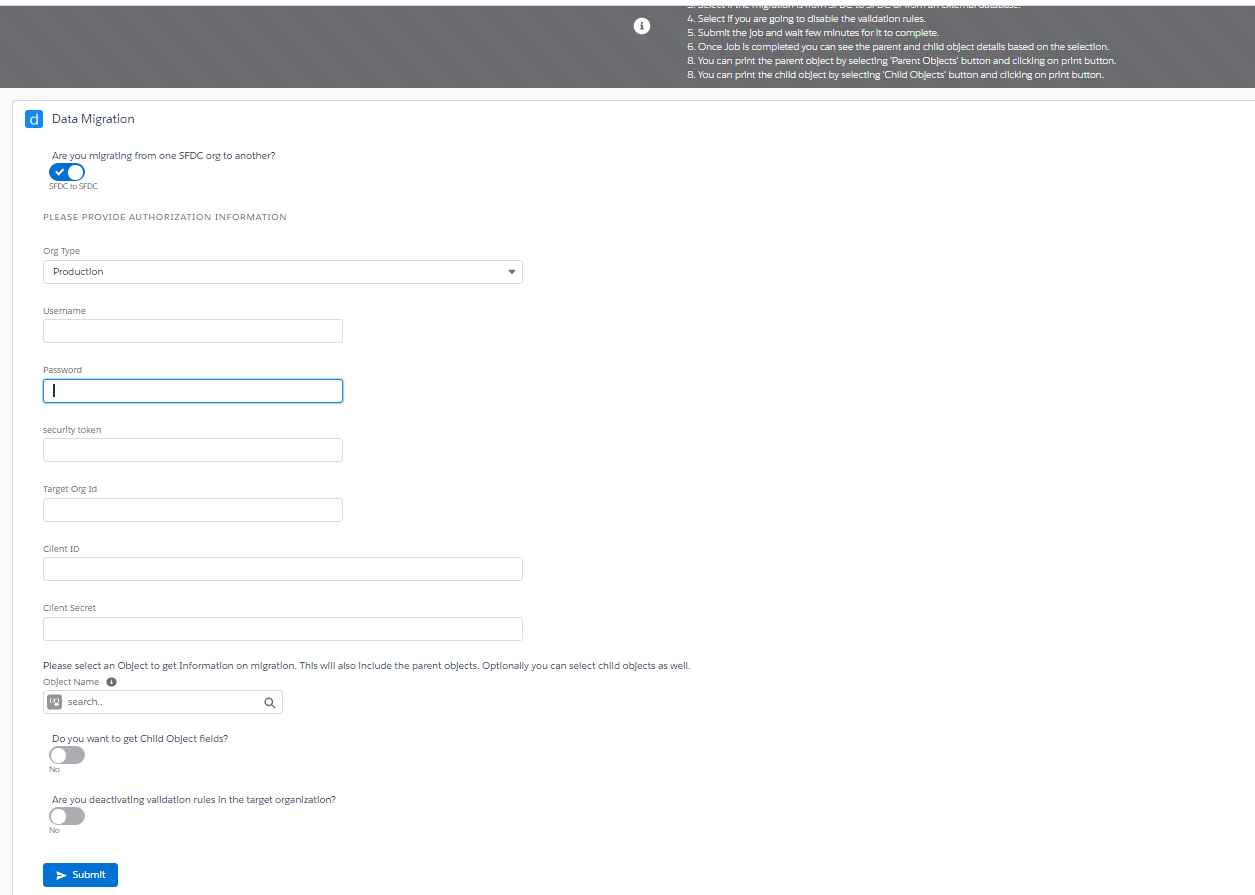
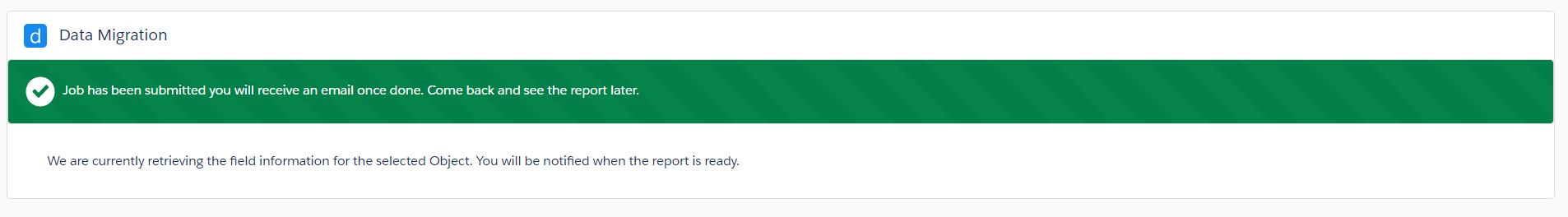
**DM Assist**

This tool allows admins to compare fields, check if the field is required in target org (validation, layout and object level) and picklist values between orgs and get a pdf document of the same. The tool allows for users to select the object and get the field and picklist details from that object’s parent and optionally child objects.



1. Select if the migration is from SFDC to SFDC or from an external database.
2. When user selects external database, only the current org object details are obtained.
3. If SFDC to SFDC is selected, user should select if the target org is production or sandbox.
4. Provide the user credentials, client Id, client Secret, target org Id, security token to perform the authentication. (For this a connected app needs to be created in the target org which we will get back in post Installation activities)
5. Select if child object details need to be retrieved.
6. Select if you are going to disable the validation rules.
7. Submit the job and wait few minutes for it to complete.
8. Once Job is completed the requestor will receive an email. You can see the parent and child object details including picklist values comparison based on the selected object.
9. User can print the parent object by selecting 'Parent Objects' button and clicking on print button.
10. User can print the child object by selecting 'Child Objects' button and clicking on print button.
11. User can print the picklist values comparison from the related parent and child objects and print them.



Once submitted a job is run in salesforce that will connect to the target org and get the object details

**Source Org Component:**

**Lightning components used:**

DataMigrationApplication – Standalone lightning app to host the components

DataMigrationDocumentGeneration – Component is used to submit the job by collecting the authentication details. It will show a message when a job is running in the background so users can’t submit duplicate jobs. It also shows the buttons (parent objects, child objects, child picklist, parent picklist, reset and print) when the job is completed.

searchComponent- To search for the objects in salesforce while user selects an object to get details

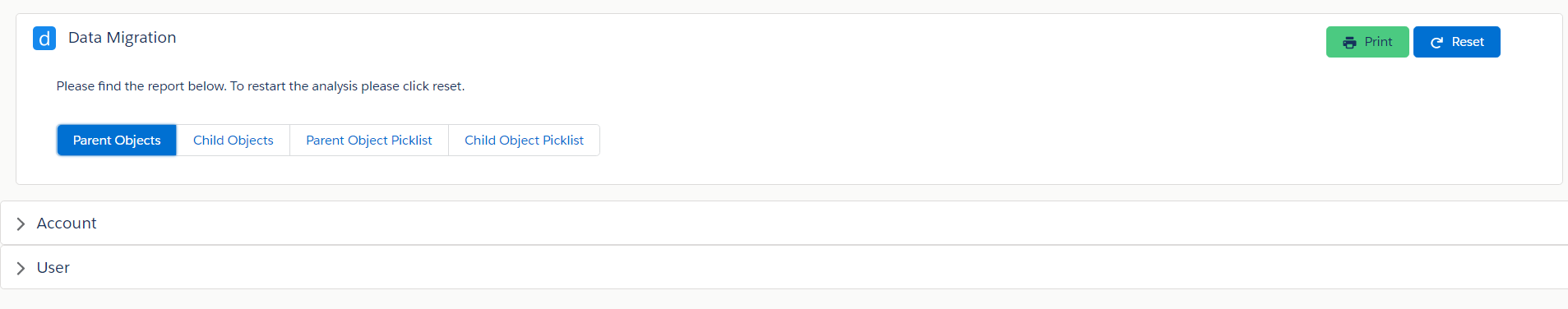
searchResults – To display search results

oSelectedValueEvent – Event is triggered when user selects an object

clearValueEvent – Event is triggered when user clears the selection

DataMigrationObjectAccordion – This component is used to show the list of objects in the form of a lightning accordion which when opened will display a table for each object with comparison with target orgs.

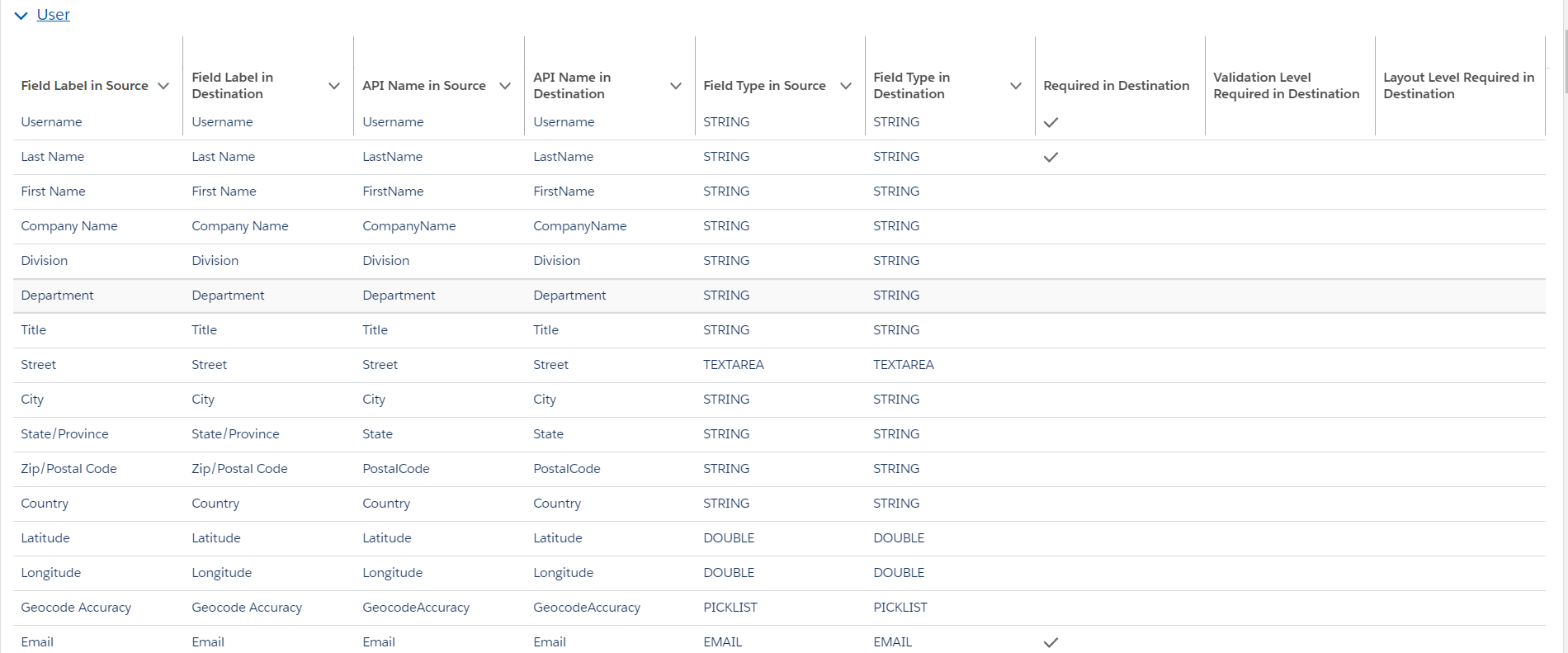
It displays the details based on whether we have selected Parent Objects/ Child Objects button where the accordion will host the table containing the object details like (field type, field label, field name, required in layout, required in object level, required in validation rule level) or if the user has selected child picklist/ parent picklist buttons then the accordion will host the table containing picklist value comparison)



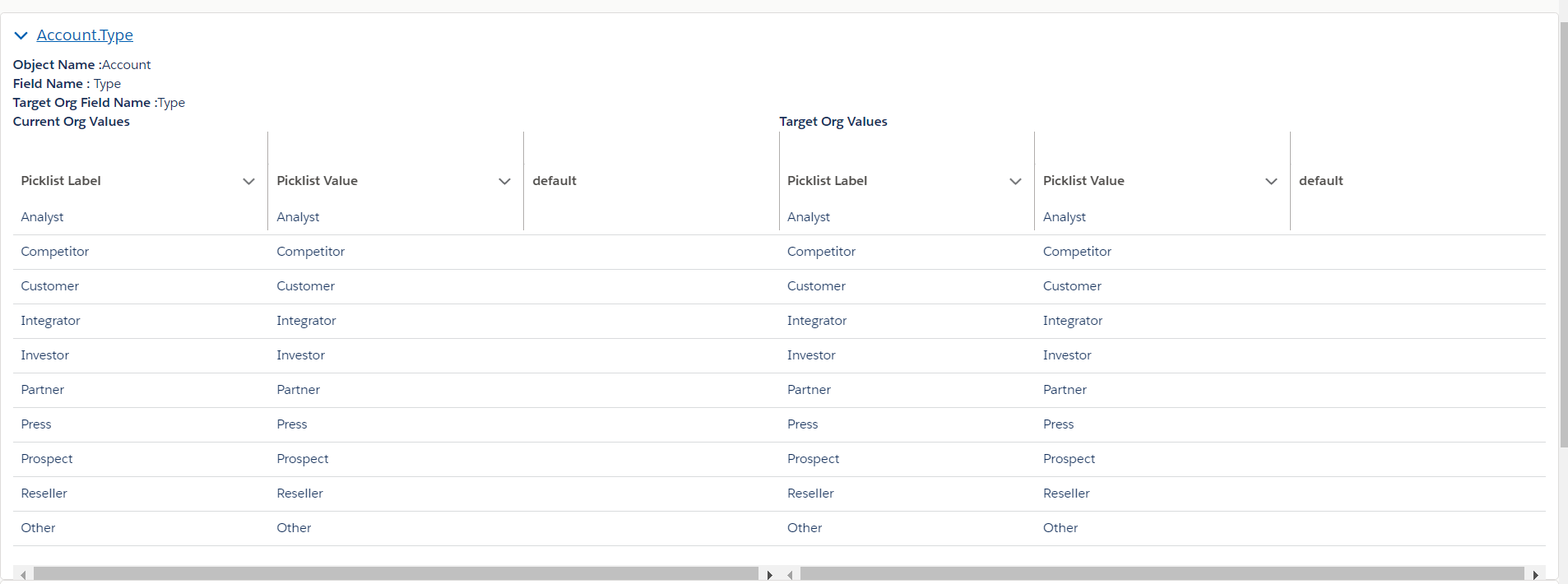
DataMigrationTableComponent – This component contains a lightning datatable which shows the object details or picklist details

PrintEvent- event is triggered when user clicks on Print button and is handled in DataMigrationObjectAccordion to open all accordions before printing.

Parent/ Child objects button is selected:

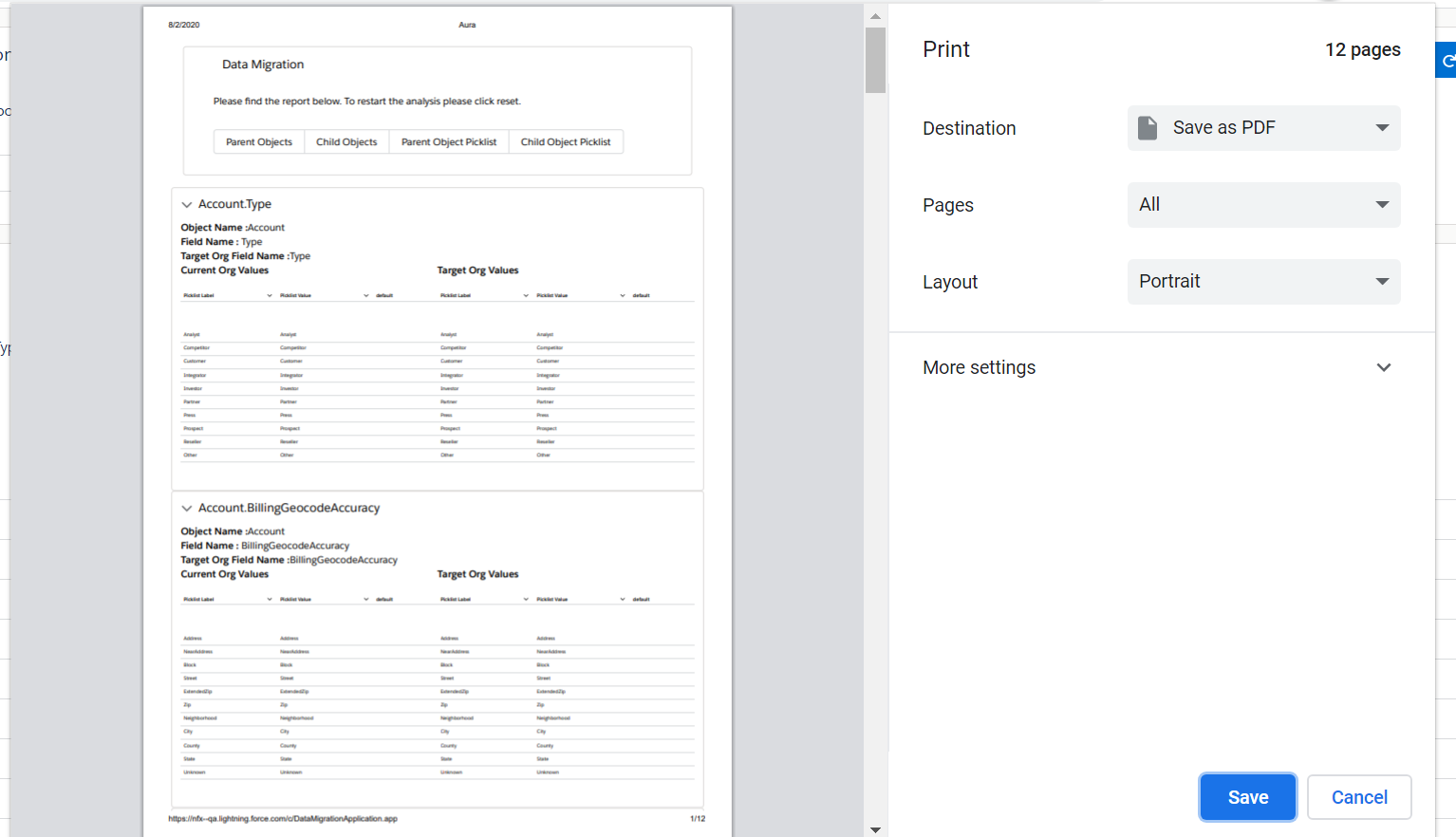


Parent/ Child picklist button is selected:



Print and reset buttons are relative to the page and will stay in place at top right, when users scroll down/ up.

Print button will print the page by opening all the accordions displaying the tables and can be saved as PDF.



Reset button will reset the object and picklist details to allow users to compare different object.



**Objects used:**

Data\_Migration\_\_b big object – Maps the object field and picklist details from current org to target org.

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Indexed** | **Purpose** |
| indexValue\_\_c | Yes | Big object index field used for maintaining uniqueness of records and faster query. |
| DateTimeStampIndex\_\_c | Yes | Big object index field used for maintaining uniqueness of records and faster query. |
| targetOrg\_\_c | Yes | Used to hold the taget org’s Id |
| Object\_\_c | Yes | Used to hold the Selected object/ Parent object/ Child object Name |
| hierarchy\_\_c |  | Indicates whether an object is parent/ child. Selected object is indicated as ‘Parent’ |
| fieldLabel\_\_c |  | Current org field label |
| fieldName\_\_c |  | Current org field api name |
| fieldType\_\_c |  | Current org field type |
| fieldLabelTarget\_\_c |  | Target org field label |
| fieldNameTarget\_\_c |  | Target org field name |
| fieldTypeTarget\_\_c |  | Target org field type |
| picklistValuesCurrent\_\_c |  | Current org Picklist values if the field type is picklist |
| picklistValuesTarget\_\_c |  | Target Org Picklist values if the field type is picklist |
| required\_\_c |  | Indicates if the field is required at object level in target org if selection on app is SFDC to SFDC. Else this will store the details of current org. |
| requiredInLayout\_\_c |  | Indicates if the field is required at layout level in target org if selection on app is SFDC to SFDC. Else this will store the details of current org. |
| requiredInValidation\_\_c |  | Indicates if the field is required at validation rule level in target org if selection on app is SFDC to SFDC. Else this will store the details of current org. |

Migration\_object\_\_c custom setting: When the form is submitted by selecting the object, the controller gets the parent objects and optionally the child objects and saves them in this custom setting records. It also holds the user selection for considering validation.

This will be used to track statuses of the metadata retrieval jobs which will get the details like required, picklist values, field name and field type. If any of the jobs fail the corresponding custom setting record from this object is updated to Failed and will be excluded from further processing. Can be used to display the job statuses to users as well in future enhancements.

|  |  |
| --- | --- |
| **Field Name** | **Purpose** |
| Name | Stores a random unique value based on time stamp to avoid getting any uniqueness error while saving records.  Ex:  2020-08-02 08:01:19-357 |
| objectName\_\_c | Holds the object name |
| consider\_Validation\_\_c | Checked if user doesn’t select ‘Are you deactivating validation rules in the target organization?’ toggle button to ‘ON’ |
| Hierarchy\_\_c | Used to store the values parent/ Child depending on the relationship of the object to the selected object. |
| Status\_\_c | Used to hold the job status. |

Data\_Migration\_Settings\_\_c custom settings: Used to hold a record which will store the current index value in the big object to maintain uniqueness while inserting them.

|  |  |
| --- | --- |
| Field Name | Purpose |
| Name | Should be equal to ‘currentIndexMigration’ |
| Record\_id\_\_c | Stores the record Id of the same record to update without querying |
| indexNumber\_\_c | Stores the next index number to be used for big object insertion. |

**VF Page**:

SessionIdPage : To get the session Id of the user when making a callout to the same org’s metadata API since Session Id cannot be retrieved from userInfo in Lightning Context.

**Classes**:

DataMigrationController –

Main controller for the application

Methods:

deleteBigObjectRecords – deletes records from the big object DataMigration\_\_b and the custom settings Migration\_Objects\_\_c to reset the application and selecting new object for comparing.

getAuthWrapper – returns an instance of a wrapper class to pass the authorization information to and from the lightning component

getBigObjects – returns a wrapper called InitializeWrapper which contains details like

1. Big object records exist
2. Jobs are running for getting object details and deleting big objects

Used by the lightning component to display conditionally.

submitJob – It performs authentication with the target org using the credentials and returns error if failed. If the authentication succeeds, It inserts the migration\_objects\_\_c records with parent and child objects along with the selected object and then calls processObjectMetadataBatch to get the field details of the object names in Migration\_Objects\_\_c records.

getBigObjectRecords – It returns the filtered big object records based on the user selection if it is Parent or child. Returns the details in the form of a Map<String, ObjectPayload> where object payload is a wrapper class containing details like hierarchy, List of ObjectWrapper that contains all details of the object fields that are in the DataMIgration\_\_b big object ex(field name, label, required, picklist values etc)

getBigObjectPicklist – Gets the picklist values from big objects based on user selection if it is parent or Child. It returns a list of ObjectWrapper.

searchComponentController –

Used by the searchComponent to get all sobjects in the org based on what user types.

deleteBigObjectsBatch –

deletes all the big object records to reset the application.

processObjectMetadataBatch –

Runs on Migration\_Objects\_\_c custom setting records and creates big object records mapping the object fields from current org to target org including getting the field details like (field name , type and picklist values) from target org by making a callout.

If considervalidation is false on the custom setting record then the getLayoutRequiredQueueable class is called directly or else getValidationRequiredQueueable is called and when that ends getLayoutRequiredQueueable will be called from there. This repeats till all Migration\_Objects\_\_c records are processed and the queueable executions run in different transactions.

getValidationRequiredQueueable –

Makes metadata API callout to target org to check if the fields in the big object records passed in from processObjectMetadataBatch are used in the validation rules of that object. Once this completes it calls getLayoutRequiredQueueable.

getLayoutRequiredQueueable-

Makes metadata API callout to target org to check if the fields in the big object records passed in from processObjectMetadataBatch or getValidationRequiredQueueable are made required in page layouts of that object. It also updates the Migaration\_Objects\_\_c records after all the big object records are processed.

Note – Since Metadata Api describemetadata method can’t be used to get more than 10 field details at a time, the getValidationRequiredQueueable and getLayoutRequiredQueueable splits the big object records into batches of 10 and processes it.

MetadataHandlerClass –

Methods:

getSessionId() and getSessionId(auth) – used to get the session Id from the current org or target org.

getRequest – used to create a HTTP Request with SOAPAction.

getResponse – used to make http callout and return the response

InitializeXML – creates a skeletion of XML document for any Soap method of Metadata API

sendMail – used to send an email to current user.

responseWrapper – contains session Id returned from making an authorization callout to target org.

authWrapper – contains the authentication details like (client id, client secret, endpoint etc)

**Target Org Components:**

**Classes:**

DataMigrationService –

'/getFields/{objectName} - Returns field details like (name, label, type, picklist values etc) from any object in the target org. Request is made from processObjectMetadataBatch is source org.

**Post Installation Steps:**

**Source Org:**

Create a custom setting record in Data\_migration\_auth with below details:

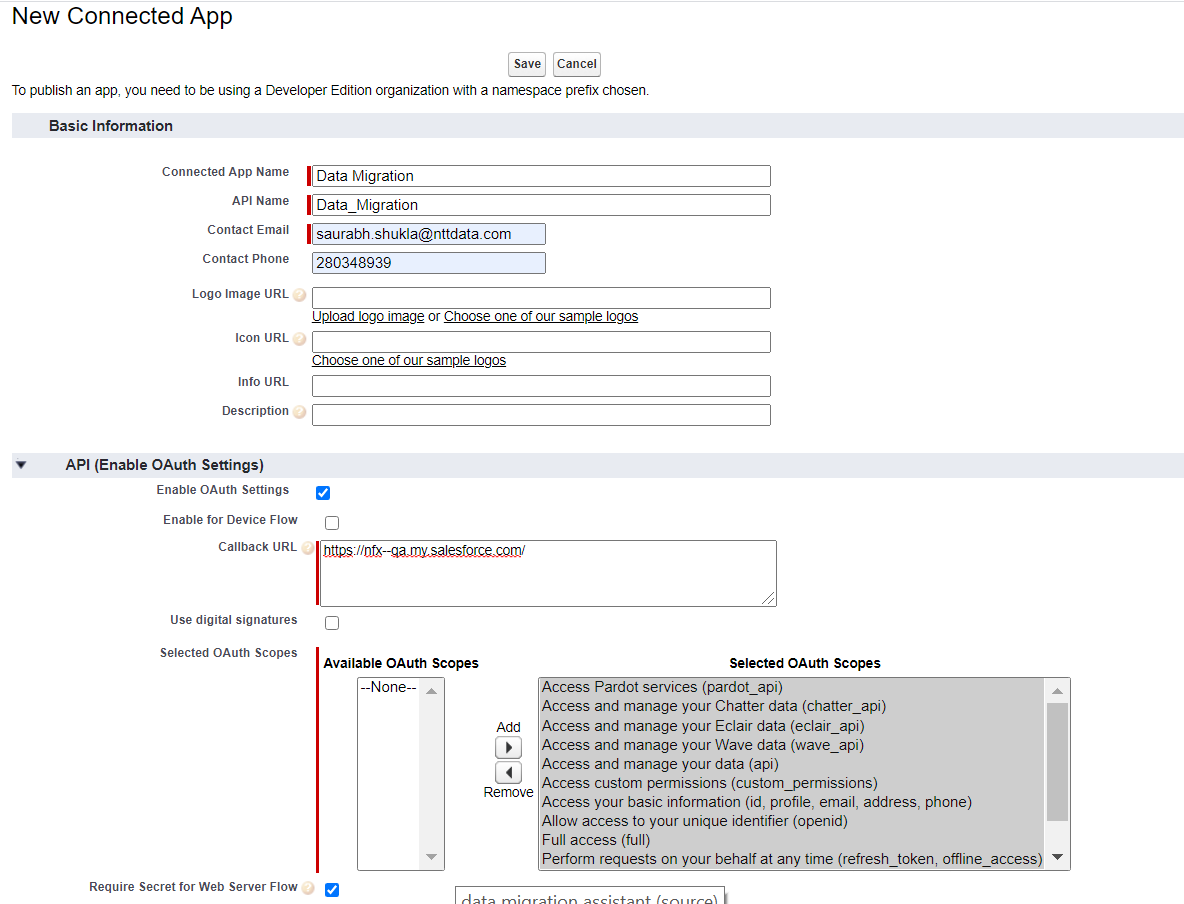
Name- currentIndexMigration

indexNumber- 1

Record Id - use the record Id of the same record after saving.

**Target Org:**

1. Create a connected app in the org with oauth enabled and below settings. Callback url can be the source org and select all oauth scopes.



1. Remove all IP restrictions from the org atleast for the user that is making the API connection.
2. Reset the security token.

The client Id, username, password, org id, client secret, security token from this org will be used for connecting to this org from DM Assist application.