Integration Flow

# Flow Breakdown

## Step Here is the \*\*Integration Flow Documentation\*\* for the project, following the format specified in the "SWB Sales Order Integration Flow" document. --- ## \*\*Integration Flow Documentation\*\* --- ### \*\*Integration Flow Name\*\*: Group-GetByName #### \*\*Flow Overview\*\*: This integration flow retrieves the ID of a Salesforce Collaboration Group based on its name. #### \*\*Steps\*\*: 1. \*\*Trigger\*\*: The flow is triggered through an HTTP GET request with the `groupName` query parameter (`[FunctionName("GroupGetByName")]`). 2. \*\*Input Transformation\*\*: - The query parameter is validated. - A `GroupByNameQuery` object is created and sent to the handler. 3. \*\*Interaction with Salesforce\*\*: - The Azure Function `GroupGetByName` invokes the `SalesforceService.GetGroupByNameAsync` method. - A SOQL query (`SELECT ID FROM CollaborationGroup WHERE Name = '{groupName}'`) is sent to Salesforce via the API. 4. \*\*Output Transformation\*\*: - If the group exists, the JSON response is transformed into a `GroupResponse` object containing the ID: ```json { "id": "0F9XXXXXXXXXXXXXXX" } ``` - If the group is not found, it returns a `404 Not Found` status. 5. \*\*Logs\*\*: - Logs the HTTP request and response using Application Insights. --- ### \*\*Integration Flow Name\*\*: Chatter-FeedItem-Post #### \*\*Flow Overview\*\*: This integration flow posts a feed item to a Salesforce Chatter group's feed elements. #### \*\*Steps\*\*: 1. \*\*Trigger\*\*: The flow is triggered through an HTTP POST request with a JSON payload containing `messagePieces` and `recordIDToAddFeedItemTo`. 2. \*\*Input Transformation\*\*: - The JSON payload is validated and converted into a `FeedItemCommand` object. 3. \*\*Access Token Retrieval\*\*: - The `GenerateChatterAccessToken` subflow is invoked to retrieve a Salesforce OAuth token. 4. \*\*Interaction with Salesforce\*\*: - The Azure Function `ChatterFeedItemPostFunction` uses `SalesforceService.PostFeedItemAsync` to post the feed item. - The payload is transformed into the format required for Salesforce's Chatter API: ```json { "body": { "messageSegments": [...] }, "feedElementType": "FeedItem", "subjectId": "001XXXXXXXXXXXXXXX" } ``` 5. \*\*Output Transformation\*\*: - Upon a successful post, a `201 Created` HTTP status is returned. 6. \*\*Logs\*\*: - Logs access token generation and Salesforce API interactions. --- ### \*\*Integration Flow Name\*\*: Chatter-FeedItem-Put #### \*\*Flow Overview\*\*: This integration flow updates or posts a feed item to a Salesforce Chatter feed only if an identical post does not already exist. #### \*\*Steps\*\*: 1. \*\*Trigger\*\*: The flow is triggered through an HTTP PUT request with a JSON payload containing `messagePieces` and `recordIDToAddFeedItemTo`. 2. \*\*Input Transformation\*\*: - The JSON payload is validated and converted into a `FeedItemCommand` object. 3. \*\*Access Token Retrieval\*\*: - The `GenerateChatterAccessToken` subflow is invoked to retrieve a Salesforce OAuth token. 4. \*\*Interaction with Salesforce\*\*: - The Azure Function `ChatterFeedItemPutFunction` first queries Salesforce using: ```sql SELECT body FROM FeedItem WHERE ParentID = ':itemID' ``` - Checks if a duplicate message exists in the `body`. If no duplicate is found, it prepares the payload for a POST request: ```json { "body": { "messageSegments": [...] }, "feedElementType": "FeedItem", "subjectId": "001XXXXXXXXXXXXXXX" } ``` - If duplicate exists, it skips the post and returns HTTP `200 OK`. 5. \*\*Output Transformation\*\*: - Returns a `201 Created` status for a successful post. 6. \*\*Logs\*\*: - Logs both the query and the post result interaction with Salesforce. --- ### \*\*Integration Flow Name\*\*: GenerateChatterAccessToken #### \*\*Flow Overview\*\*: This subflow generates an OAuth access token using Salesforce credentials for subsequent API calls. #### \*\*Steps\*\*: 1. \*\*Trigger\*\*: Invoked by other flows such as `Chatter-FeedItem-Post` or `Chatter-FeedItem-Put` when access to Salesforce is required. 2. \*\*Input Transformation\*\*: - The flow prepares a form-data payload using environment-secured properties (`client\_id`, `client\_secret`, `username`, `password`): ```json { "grant\_type": "password", "client\_id": "...", "client\_secret": "...", "username": "...", "password": "..." } ``` 3. \*\*Interaction with Salesforce\*\*: - The Azure Function `GenerateChatterAccessTokenFunction` sends the above payload to Salesforce's OAuth API endpoint: ```url POST /services/oauth2/token ``` 4. \*\*Output Transformation\*\*: - Extracts the `access\_token` from Salesforce’s response and passes it back to the calling flow: ```json { "access\_token": "00DXXXXXXXXXXXXXXX!AQoXXXXXXXXXXXXXXX", "instance\_url": "https://yourInstance.salesforce.com" } ``` 5. \*\*Logs\*\*: - Logs the request and response using Application Insights for traceability. --- ### \*\*Global Error Handling\*\* 1. \*\*Retry Strategy\*\*: - All Salesforce API calls include retry policies defined through HTTP client configurations or Azure Resiliency Patterns. - Retry count and delay are configurable via `local.settings.json`. 2. \*\*Error Logging\*\*: - All flows log errors to Application Insights, including: - Correlation ID - Error message - Stack trace 3. \*\*Fallback Handling\*\*: - For critical failures, pre-configured fallback notifications are sent to a resolver or defined email recipient groups. --- This documentation represents the business-readable integration flow details. Let me know if specific sections need deeper elaboration!

Group-GetByName  
  
#### \*\*Flow Overview\*\*:  
This integration flow retrieves the ID of a Salesforce Collaboration Group based on its name.  
  
#### \*\*Steps\*\*:  
1. \*\*Trigger\*\*: The flow is triggered through an HTTP GET request with the `groupName` query parameter (`[FunctionName("GroupGetByName")]`).  
2. \*\*Input Transformation\*\*:  
 - The query parameter is validated.  
 - A `GroupByNameQuery` object is created and sent to the handler.  
3. \*\*Interaction with Salesforce\*\*:  
 - The Azure Function `GroupGetByName` invokes the `SalesforceService.GetGroupByNameAsync` method.  
 - A SOQL query (`SELECT ID FROM CollaborationGroup WHERE Name = '{groupName}'`) is sent to Salesforce via the API.  
4. \*\*Output Transformation\*\*:  
 - If the group exists, the JSON response is transformed into a `GroupResponse` object containing the ID:  
 ```json  
 {  
 "id": "0F9XXXXXXXXXXXXXXX"  
 }  
 ```  
 - If the group is not found, it returns a `404 Not Found` status.  
5. \*\*Logs\*\*:  
 - Logs the HTTP request and response using Application Insights.  
  
---  
  
### \*\*Integration Flow Name\*\*: Chatter-FeedItem-Post  
  
#### \*\*Flow Overview\*\*:  
This integration flow posts a feed item to a Salesforce Chatter group's feed elements.  
  
#### \*\*Steps\*\*:  
1. \*\*Trigger\*\*: The flow is triggered through an HTTP POST request with a JSON payload containing `messagePieces` and `recordIDToAddFeedItemTo`.  
2. \*\*Input Transformation\*\*:  
 - The JSON payload is validated and converted into a `FeedItemCommand` object.  
3. \*\*Access Token Retrieval\*\*:  
 - The `GenerateChatterAccessToken` subflow is invoked to retrieve a Salesforce OAuth token.  
4. \*\*Interaction with Salesforce\*\*:  
 - The Azure Function `ChatterFeedItemPostFunction` uses `SalesforceService.PostFeedItemAsync` to post the feed item.  
 - The payload is transformed into the format required for Salesforce's Chatter API:  
 ```json  
 {  
 "body": {  
 "messageSegments": [...]  
 },  
 "feedElementType": "FeedItem",  
 "subjectId": "001XXXXXXXXXXXXXXX"  
 }  
 ```  
5. \*\*Output Transformation\*\*:  
 - Upon a successful post, a `201 Created` HTTP status is returned.  
6. \*\*Logs\*\*:  
 - Logs access token generation and Salesforce API interactions.  
  
---  
  
### \*\*Integration Flow Name\*\*: Chatter-FeedItem-Put  
  
#### \*\*Flow Overview\*\*:  
This integration flow updates or posts a feed item to a Salesforce Chatter feed only if an identical post does not already exist.  
  
#### \*\*Steps\*\*:  
1. \*\*Trigger\*\*: The flow is triggered through an HTTP PUT request with a JSON payload containing `messagePieces` and `recordIDToAddFeedItemTo`.  
2. \*\*Input Transformation\*\*:  
 - The JSON payload is validated and converted into a `FeedItemCommand` object.  
3. \*\*Access Token Retrieval\*\*:  
 - The `GenerateChatterAccessToken` subflow is invoked to retrieve a Salesforce OAuth token.  
4. \*\*Interaction with Salesforce\*\*:  
 - The Azure Function `ChatterFeedItemPutFunction` first queries Salesforce using:  
 ```sql  
 SELECT body FROM FeedItem WHERE ParentID = ':itemID'  
 ```  
 - Checks if a duplicate message exists in the `body`. If no duplicate is found, it prepares the payload for a POST request:  
 ```json  
 {  
 "body": {  
 "messageSegments": [...]  
 },  
 "feedElementType": "FeedItem",  
 "subjectId": "001XXXXXXXXXXXXXXX"  
 }  
 ```  
 - If duplicate exists, it skips the post and returns HTTP `200 OK`.  
5. \*\*Output Transformation\*\*:  
 - Returns a `201 Created` status for a successful post.  
6. \*\*Logs\*\*:  
 - Logs both the query and the post result interaction with Salesforce.  
  
---  
  
### \*\*Integration Flow Name\*\*: GenerateChatterAccessToken  
  
#### \*\*Flow Overview\*\*:  
This subflow generates an OAuth access token using Salesforce credentials for subsequent API calls.  
  
#### \*\*Steps\*\*:  
1. \*\*Trigger\*\*: Invoked by other flows such as `Chatter-FeedItem-Post` or `Chatter-FeedItem-Put` when access to Salesforce is required.  
2. \*\*Input Transformation\*\*:  
 - The flow prepares a form-data payload using environment-secured properties (`client\_id`, `client\_secret`, `username`, `password`):  
 ```json  
 {  
 "grant\_type": "password",  
 "client\_id": "...",  
 "client\_secret": "...",  
 "username": "...",  
 "password": "..."  
 }  
 ```  
3. \*\*Interaction with Salesforce\*\*:  
 - The Azure Function `GenerateChatterAccessTokenFunction` sends the above payload to Salesforce's OAuth API endpoint:  
 ```url  
 POST /services/oauth2/token  
 ```  
4. \*\*Output Transformation\*\*:  
 - Extracts the `access\_token` from Salesforce’s response and passes it back to the calling flow:  
 ```json  
 {  
 "access\_token": "00DXXXXXXXXXXXXXXX!AQoXXXXXXXXXXXXXXX",  
 "instance\_url": "https://yourInstance.salesforce.com"  
 }  
 ```  
5. \*\*Logs\*\*:  
 - Logs the request and response using Application Insights for traceability.  
  
---  
  
### \*\*Global Error Handling\*\*  
  
1. \*\*Retry Strategy\*\*:  
 - All Salesforce API calls include retry policies defined through HTTP client configurations or Azure Resiliency Patterns.  
 - Retry count and delay are configurable via `local.settings.json`.  
  
2. \*\*Error Logging\*\*:  
 - All flows log errors to Application Insights, including:  
 - Correlation ID  
 - Error message  
 - Stack trace  
  
3. \*\*Fallback Handling\*\*:  
 - For critical failures, pre-configured fallback notifications are sent to a resolver or defined email recipient groups.  
  
---  
  
This documentation represents the business-readable integration flow details. Let me know if specific sections need deeper elaboration!