Integration Flow

# Flow Breakdown

## Step ### Integration Flow Documentation: AIS Integration Flows --- ### \*\*Integration Flow for Group-GetByName\*\* #### \*\*Summary\*\* This integration flow handles retrieving a group's ID from Salesforce based on its name. The flow validates user authorization, queries the Salesforce CollaborationGroup for the provided group name, and returns the corresponding group ID or a 404 error response. #### \*\*Data Flow\*\* 1. \*\*Azure Function Trigger\*\*: - \*\*Trigger\*\*: HTTP GET request with query parameter `groupName` (e.g., `/group/{groupName}`). - Validates user authorization via Salesforce authorization API. 2. \*\*Command and Handler\*\*: - Trigger delegates processing to the `GetGroupIdCommand` using MediatR. - The command invokes the `GetGroupIdHandler` which sends the query to the `SalesforceAdapter`. 3. \*\*Adapter Execution\*\*: - \*\*Query\*\*: Sends a Salesforce query `SELECT ID FROM CollaborationGroup WHERE Name = ':groupName'`. - Salesforce returns the group ID or throws an exception if not found. 4. \*\*Decision and Transformation\*\*: - \*\*Condition\*\*: Validates if a valid group ID is found. - \*\*Transformation\*\*: Maps payload from Salesforce response to JSON format `{ "id": "GroupID" }`. 5. \*\*Response\*\*: - \*\*Success\*\*: Returns HTTP 200 with the group ID in the payload. - \*\*Failure\*\*: Returns HTTP 404 with an error message "Group not found". --- ### \*\*Integration Flow for Chatter-FeedItem-Post\*\* #### \*\*Summary\*\* This integration flow posts a message to Salesforce Chatter's FeedItem API. It validates user authorization, retrieves an access token, and sends the message payload to the appropriate subject in the Chatter feed. #### \*\*Data Flow\*\* 1. \*\*Azure Function Trigger\*\*: - \*\*Trigger\*\*: HTTP POST request with request body containing payload to be posted (e.g., record ID and message segments). 2. \*\*Command and Handler\*\*: - Trigger delegates processing to the `PostChatterMessageCommand` using MediatR. - The command invokes the `PostChatterMessageHandler` which prepares the payload, retrieves an access token, and posts the message. 3. \*\*Adapter Execution\*\*: - \*\*Authorization\*\*: Validates user authorization via Salesforce authorization API. - \*\*Access Token Retrieval\*\*: Requests an access token via Salesforce OAuth API. - \*\*Payload Posting\*\*: Sends a POST request to the Salesforce Chatter FeedElements API `services/data/v47.0/chatter/feed-elements` with the prepared JSON payload. 4. \*\*Transformation\*\*: - \*\*Output JSON\*\*: `{ "body": { "messageSegments": vars.originalPayload.messagePieces }, "feedElementType": "FeedItem", "subjectId": vars.originalPayload.recordIDToAddFeedItemTo }`. 5. \*\*Response\*\*: - \*\*Success\*\*: Returns HTTP 201 with no payload. - \*\*Failure\*\*: Returns HTTP 400 with an error message indicating the issue. --- ### \*\*Integration Flow for Chatter-FeedItem-Put\*\* #### \*\*Summary\*\* This integration flow updates or adds a Chatter FeedItem message based on the existence of a previous post. If the specified record does not have the given message, it posts a new message. Otherwise, it returns a 200 status indicating no action was performed. #### \*\*Data Flow\*\* 1. \*\*Azure Function Trigger\*\*: - \*\*Trigger\*\*: HTTP PUT request with request body containing record ID and message segments. 2. \*\*Command and Handler\*\*: - Trigger delegates processing to the `PutChatterMessageCommand` using MediatR. - The command invokes the `PutChatterMessageHandler` which validates payload existence, retrieves an access token, and performs logic to decide if a post should be added. 3. \*\*Adapter Execution\*\*: - \*\*Validation\*\*: Checks for existing posts via Salesforce FeedItem query: `SELECT body FROM FeedItem WHERE ParentID = ':itemID'`. - \*\*Access Token Retrieval\*\*: Requests an access token via Salesforce OAuth API. - \*\*Condition\*\*: Compares message content in the existing FeedItems with the incoming payload. 4. \*\*Decision and Transformation\*\*: - \*\*Condition\*\*: Posts a new message if no existing message matches. - \*\*Payload\*\*: Transforms incoming payload to JSON format. 5. \*\*Response\*\*: - \*\*Success\*\*: - \*\*New Post\*\*: Returns HTTP 201 with no payload. - \*\*Existing Post\*\*: Returns HTTP 200 indicating no new post was added. - \*\*Failure\*\*: Returns HTTP 400 with an error message. --- ### \*\*Flow Summary Comparison\*\* | \*\*Flow Name\*\* | \*\*Trigger\*\* | \*\*External Actions\*\* | \*\*Destination API\*\* | \*\*Final Response\*\* | |--------------------------|--------------------------|---------------------------------------------|---------------------------------------------------------|-----------------------------------| | \*\*Group-GetByName\*\* | HTTP GET `/group/{name}` | Salesforce Authorization, Salesforce Query | Salesforce CollaborationGroup `/SELECT ID...` | HTTP 200 (Success) or HTTP 404 (Fail) | | \*\*Chatter-FeedItem-Post\*\* | HTTP POST `/chatter/post` | Salesforce Authorization, Access Token | Salesforce FeedElements `POST /chatter/feed-elements` | HTTP 201 (Success) or HTTP 400 (Fail) | | \*\*Chatter-FeedItem-Put\*\* | HTTP PUT `/chatter/put` | Salesforce Authorization, Query for Posts | Salesforce FeedElements `PUT /chatter/feed-elements` | HTTP 201 (New) or HTTP 200 (Exists) | --- ### Observability and Monitoring - All flows log the following metrics to Azure Application Insights: - \*\*Trigger Invocations\*\*: Number of incoming HTTP requests to each function. - \*\*Transformation Times\*\*: Latency in processing payloads within handlers. - \*\*External API Success/Failures\*\*: Number of successful API responses and retries for Salesforce queries. - Alerts setup for: - High rate of HTTP 404 for "Group not found". - Frequent duplicate message scenarios in `Chatter-FeedItem-Put`. - Errors in authorization or access token retrieval for Salesforce. --- ### Key Assumptions: 1. Salesforce APIs used in the flows are stable and accessible during runtime. 2. Necessary configurations (e.g., `local.settings.json`) are set up correctly for Salesforce credentials. 3. Monitoring for system health includes retries and fallback mechanisms for network issues. This documentation provides a clear overview for developers, integrators, and business users to understand the integration flows and their key components. Let me know if you'd like refinements or additional details.

AIS Integration Flows  
  
---  
  
### \*\*Integration Flow for Group-GetByName\*\*  
#### \*\*Summary\*\*  
This integration flow handles retrieving a group's ID from Salesforce based on its name. The flow validates user authorization, queries the Salesforce CollaborationGroup for the provided group name, and returns the corresponding group ID or a 404 error response.  
  
#### \*\*Data Flow\*\*  
1. \*\*Azure Function Trigger\*\*:  
 - \*\*Trigger\*\*: HTTP GET request with query parameter `groupName` (e.g., `/group/{groupName}`).  
 - Validates user authorization via Salesforce authorization API.  
   
2. \*\*Command and Handler\*\*:  
 - Trigger delegates processing to the `GetGroupIdCommand` using MediatR.  
 - The command invokes the `GetGroupIdHandler` which sends the query to the `SalesforceAdapter`.  
  
3. \*\*Adapter Execution\*\*:  
 - \*\*Query\*\*: Sends a Salesforce query `SELECT ID FROM CollaborationGroup WHERE Name = ':groupName'`.  
 - Salesforce returns the group ID or throws an exception if not found.  
  
4. \*\*Decision and Transformation\*\*:  
 - \*\*Condition\*\*: Validates if a valid group ID is found.  
 - \*\*Transformation\*\*: Maps payload from Salesforce response to JSON format `{ "id": "GroupID" }`.  
  
5. \*\*Response\*\*:  
 - \*\*Success\*\*: Returns HTTP 200 with the group ID in the payload.  
 - \*\*Failure\*\*: Returns HTTP 404 with an error message "Group not found".  
  
---  
  
### \*\*Integration Flow for Chatter-FeedItem-Post\*\*  
#### \*\*Summary\*\*  
This integration flow posts a message to Salesforce Chatter's FeedItem API. It validates user authorization, retrieves an access token, and sends the message payload to the appropriate subject in the Chatter feed.  
  
#### \*\*Data Flow\*\*  
1. \*\*Azure Function Trigger\*\*:  
 - \*\*Trigger\*\*: HTTP POST request with request body containing payload to be posted (e.g., record ID and message segments).  
  
2. \*\*Command and Handler\*\*:  
 - Trigger delegates processing to the `PostChatterMessageCommand` using MediatR.  
 - The command invokes the `PostChatterMessageHandler` which prepares the payload, retrieves an access token, and posts the message.  
  
3. \*\*Adapter Execution\*\*:  
 - \*\*Authorization\*\*: Validates user authorization via Salesforce authorization API.  
 - \*\*Access Token Retrieval\*\*: Requests an access token via Salesforce OAuth API.  
 - \*\*Payload Posting\*\*: Sends a POST request to the Salesforce Chatter FeedElements API `services/data/v47.0/chatter/feed-elements` with the prepared JSON payload.  
  
4. \*\*Transformation\*\*:  
 - \*\*Output JSON\*\*: `{ "body": { "messageSegments": vars.originalPayload.messagePieces }, "feedElementType": "FeedItem", "subjectId": vars.originalPayload.recordIDToAddFeedItemTo }`.  
  
5. \*\*Response\*\*:  
 - \*\*Success\*\*: Returns HTTP 201 with no payload.  
 - \*\*Failure\*\*: Returns HTTP 400 with an error message indicating the issue.  
  
---  
  
### \*\*Integration Flow for Chatter-FeedItem-Put\*\*  
#### \*\*Summary\*\*  
This integration flow updates or adds a Chatter FeedItem message based on the existence of a previous post. If the specified record does not have the given message, it posts a new message. Otherwise, it returns a 200 status indicating no action was performed.  
  
#### \*\*Data Flow\*\*  
1. \*\*Azure Function Trigger\*\*:  
 - \*\*Trigger\*\*: HTTP PUT request with request body containing record ID and message segments.  
  
2. \*\*Command and Handler\*\*:  
 - Trigger delegates processing to the `PutChatterMessageCommand` using MediatR.  
 - The command invokes the `PutChatterMessageHandler` which validates payload existence, retrieves an access token, and performs logic to decide if a post should be added.  
  
3. \*\*Adapter Execution\*\*:  
 - \*\*Validation\*\*: Checks for existing posts via Salesforce FeedItem query: `SELECT body FROM FeedItem WHERE ParentID = ':itemID'`.  
 - \*\*Access Token Retrieval\*\*: Requests an access token via Salesforce OAuth API.  
 - \*\*Condition\*\*: Compares message content in the existing FeedItems with the incoming payload.  
  
4. \*\*Decision and Transformation\*\*:  
 - \*\*Condition\*\*: Posts a new message if no existing message matches.  
 - \*\*Payload\*\*: Transforms incoming payload to JSON format.  
  
5. \*\*Response\*\*:  
 - \*\*Success\*\*:  
 - \*\*New Post\*\*: Returns HTTP 201 with no payload.  
 - \*\*Existing Post\*\*: Returns HTTP 200 indicating no new post was added.  
 - \*\*Failure\*\*: Returns HTTP 400 with an error message.  
  
---  
  
### \*\*Flow Summary Comparison\*\*  
  
| \*\*Flow Name\*\* | \*\*Trigger\*\* | \*\*External Actions\*\* | \*\*Destination API\*\* | \*\*Final Response\*\* |  
|--------------------------|--------------------------|---------------------------------------------|---------------------------------------------------------|-----------------------------------|  
| \*\*Group-GetByName\*\* | HTTP GET `/group/{name}` | Salesforce Authorization, Salesforce Query | Salesforce CollaborationGroup `/SELECT ID...` | HTTP 200 (Success) or HTTP 404 (Fail) |  
| \*\*Chatter-FeedItem-Post\*\* | HTTP POST `/chatter/post` | Salesforce Authorization, Access Token | Salesforce FeedElements `POST /chatter/feed-elements` | HTTP 201 (Success) or HTTP 400 (Fail) |  
| \*\*Chatter-FeedItem-Put\*\* | HTTP PUT `/chatter/put` | Salesforce Authorization, Query for Posts | Salesforce FeedElements `PUT /chatter/feed-elements` | HTTP 201 (New) or HTTP 200 (Exists) |  
  
---  
  
### Observability and Monitoring  
- All flows log the following metrics to Azure Application Insights:  
 - \*\*Trigger Invocations\*\*: Number of incoming HTTP requests to each function.  
 - \*\*Transformation Times\*\*: Latency in processing payloads within handlers.  
 - \*\*External API Success/Failures\*\*: Number of successful API responses and retries for Salesforce queries.  
- Alerts setup for:  
 - High rate of HTTP 404 for "Group not found".  
 - Frequent duplicate message scenarios in `Chatter-FeedItem-Put`.  
 - Errors in authorization or access token retrieval for Salesforce.  
  
---  
  
### Key Assumptions:  
1. Salesforce APIs used in the flows are stable and accessible during runtime.  
2. Necessary configurations (e.g., `local.settings.json`) are set up correctly for Salesforce credentials.  
3. Monitoring for system health includes retries and fallback mechanisms for network issues.  
  
This documentation provides a clear overview for developers, integrators, and business users to understand the integration flows and their key components. Let me know if you'd like refinements or additional details.