Integration Flow

# Flow Breakdown

## Step Below is the \*\*Integration Flow Document\*\* for the AIS Project implementation:

---

## \*\*Integration Flow Document\*\*

### \*\*AIS Salesforce Integration\*\*

---

### \*\*Objective\*\*

This integration handles Salesforce CollaborationGroup retrieval and Chatter FeedItem management using Azure Functions in conjunction with MediatR, Salesforce APIs, and Azure Service Bus. It ensures modularity, scalability, and strict separation of domain, application, infrastructure, and functions layers.

---

### \*\*Data Flow Overview\*\*

**The integration consists of three main flows:**

1. \*\*Retrieving Salesforce Group by Name\*\*

2. \*\*Posting Chatter FeedItem\*\*

3. \*\*Conditionally Posting or Updating Chatter FeedItem\*\*

Each flow operates autonomously via Azure Functions triggers and leverages typed configuration, Salesforce API calls, and domain-encapsulated business logic.

---

### \*\*Flow Details\*\*

#### \*\*Retrieve Salesforce Group by Name\*\*

**1. \*\*Source of Action\*\*:** Azure HTTP Trigger (`GET` at `/salesforce/group/{groupName}`).

**2. \*\*Request Details\*\*:**

**- Input:** QueryParameter `groupName`.

**- Example:** `/salesforce/group/MarketingTeam`

**3. \*\*Processing\*\*:**

- Azure Function \*\*SalesforceRetrieveGroupFunction\*\* accepts the HTTP request.

- Delegates to `RetrieveGroupCommandHandler` via MediatR.

- Sends the `RetrieveGroupCommand` to `IGroupService`.

**- Uses \*\*Salesforce API\*\* to perform an `SOQL Query`:**

```

**SELECT ID FROM CollaborationGroup WHERE Name = ':** groupName'

```

**- Response:**

**- On success:** Group ID returned.

**- On failure:** HTTP `404` returned.

**4. \*\*Destination\*\*:**

**- Returns the Group ID in JSON format:**

```json

**{ "id":** "00XXXXXX0001ABCD" }

```

---

#### \*\*Post Chatter FeedItem\*\*

**1. \*\*Source of Action\*\*:** Azure HTTP Trigger (`POST` at `/salesforce/chatter/feedItem`).

**2. \*\*Request Details\*\*:**

**- Input:** JSON payload, example:

```json

{

**"messagePieces":** [

{

**"type":** "text",

**"text":** "Hello Team, welcome onboard!"

}

],

**"recordIDToAddFeedItemTo":** "003XX00001ABCD"

}

```

**3. \*\*Processing\*\*:**

- Azure Function \*\*SalesforcePostFeedItemFunction\*\* accepts the HTTP request.

- Delegates to `PostFeedItemCommandHandler` via MediatR.

- Sends the `PostFeedItemCommand` to `IFeedItemService`.

**- Authenticates with Salesforce via \*\*GenerateAccessToken\*\* using:**

```

/services/oauth2/token

```

**- Response:** Salesforce OAuth token.

- Constructs the Chatter FeedItem payload.

**- Makes an \*\*HTTP POST request\*\* to:**

```

/services/data/v47.0/chatter/feed-elements

```

**- Input:**

```json

{

**"body":** {

**"messageSegments":** [

**{ "type":** "Text", "text": "Hello Team, welcome onboard!" }

]

},

**"feedElementType":** "FeedItem",

**"subjectId":** "003XX00001ABCD"

}

```

**- Response:**

**- On success:** HTTP `201 Created`.

**- On failure:** Error response logged internally.

**4. \*\*Destination\*\*:**

- Posts a FeedItem in Salesforce Chatter.

- HTTP status codes confirm action completion.

---

#### \*\*Conditionally Post or Update Chatter FeedItem\*\*

**1. \*\*Source of Action\*\*:** Azure HTTP Trigger (`POST` at `/salesforce/chatter/feedItem/update`).

**2. \*\*Request Details\*\*:**

**- Input:** JSON payload, same format as "Post Chatter FeedItem".

**3. \*\*Processing\*\*:**

- Azure Function \*\*SalesforcePutFeedItemFunction\*\* accepts the request.

- Delegates to `PutFeedItemCommandHandler` via MediatR.

- Authenticates with Salesforce via \*\*GenerateAccessToken\*\*.

**- Checks for existing Chatter post:**

```

**SELECT body FROM FeedItem WHERE ParentID = ':** itemID'

```

**- Condition:**

**- If `messageSegments` already exist in FeedItem:**

- Returns HTTP `200 OK`.

**- Otherwise:**

- Posts a new FeedItem as per "Post Chatter FeedItem" flow.

**4. \*\*Destination\*\*:**

- Posts or skips posting based on existing Chatter content for the record.

- Provides HTTP status codes (`200`, `201`) indicating results.

---

### \*\*System Interactions\*\*

**1. \*\*Authorization\*\*:**

- All Salesforce interactions require an OAuth Token.

- Token generated via `/services/oauth2/token`.

- Retrieved credentials managed via `local.settings.json`.

**2. \*\*Salesforce Interaction Details\*\*:**

**- \*\*Retrieving Groups\*\*:** SOQL Query via `CollaborationGroup`.

**- \*\*Posting FeedItem\*\*:** HTTP POST request to Salesforce Chatter.

**3. \*\*Error Handling\*\*:**

- Errors propagate through MediatR handlers and log via \*\*Application Insights\*\*.

**- Include:**

- Generic Salesforce issues (e.g., Unauthorized, BadRequest).

- Invalid input or missing records in Salesforce.

---

### \*\*Transformations\*\*

**1. \*\*Group Name Retrieval\*\*:**

**- Input:** Simple string.

**- Output:** JSON object with `Group ID`.

**2. \*\*FeedItem Posting\*\*:**

**- Input:** JSON payload (message construction).

**- Output:** Constructed FeedItem object for Salesforce API.

**3. \*\*Conditional Posting\*\*:**

**- Input:** JSON payload (message content).

**- Processing:** Filters and checks existing FeedItems.

**- Output:** Condition-based FeedItem posting or skipped action.

---

### \*\*Data Sources\*\*

**1. \*\*Salesforce Data Locations\*\*:**

- CollaborationGroup

- FeedItem

**2. \*\*Configuration Sources\*\*:**

- Stored credentials for Salesforce API access via `local.settings.json`.

---

### \*\*Error Propagation and Handling\*\*

**1. \*\*Logging\*\*:**

- Use \*\*Application Insights\*\* for operational telemetry.

- Log transformations, API calls, and exceptions at each layer.

**2. \*\*Recovery\*\*:**

- Reconnection attempts configured for HTTP requests.

- Global error-handling strategy ensures email notifications for exceptions.

---

### \*\*Deployment and Monitoring\*\*

**1. \*\*Deployment Target\*\*:**

- Azure Functions stored in Function App (`AIS.FunctionApp`).

**2. \*\*Monitoring Tools\*\*:**

- \*\*Application Insights\*\* integrated for logs and metrics.

- Integration alerts configured for exceptions, failed API calls, and throughput.

---

### \*\*Flow Summary\*\*

| \*\*Flow Name\*\* | \*\*Trigger\*\* | \*\*External Calls\*\* | \*\*Destination\*\* |

|-----------------------------------|---------------------------------------|----------------------------|------------------------------------|

| \*\*Retrieve Salesforce Group\*\* | HTTP GET `/salesforce/group` | Salesforce SOQL Query | Group ID in JSON response |

| \*\*Post Chatter FeedItem\*\* | HTTP POST `/salesforce/chatter/feedItem` | Generate OAuth, POST FeedItem | HTTP `201 Created` |

| \*\*Condition Post Chatter FeedItem\*\*| HTTP POST `/salesforce/chatter/feedItem/update` | SOQL Query, POST FeedItem | HTTP `200 OK` or `201 Created` |

---

This document outlines all system interactions, logic, configurations, and flow behaviors, ensuring the integration adheres to Hexagonal Architecture principles while remaining scalable and maintainable.