# Detailed API Descriptions with Authentication and Logging

## Common Step for All APIs: JWT Authentication

1. Extract the JWT token from the Authorization header.
2. Decode and validate the token using the shared secret/public key.
3. Check if the token is expired or invalid. If yes, return a 401 Unauthorized response.
4. Extract user information such as user\_id and permissions from token claims.
5. Log the authenticated user\_id and timestamp.

## 1. Create Memo API

1. Authenticate the request using JWT.
2. Receive JSON body with user\_id, client\_name, and content\_inputs.
3. Log the request with user\_id and timestamp.
4. Validate presence and correctness of input fields.
5. Insert a new memo into the PostgreSQL table with 'Processing' status.
6. Log the generated memo\_id.
7. Trigger asynchronous content generation logic in the background.
8. Once completed, update memo content and status to 'Completed'.
9. Log the completion update.
10. Return a JSON response with memo\_id, client\_name, and status.
11. Handle errors and log any failures.

## 2. Check Processing Status API

1. Authenticate the request using JWT.
2. Receive user\_id and memo\_ids as path parameters.
3. Log user\_id and requested memo\_ids.
4. Validate the user\_id and memo\_ids.
5. Query PostgreSQL for memo status.
6. Return a JSON mapping of memo\_id to status.
7. Handle and log input or server errors.

## 3. Get Memo Data API

1. Authenticate the request using JWT.
2. Receive memo\_id as a path parameter.
3. Log memo\_id and user\_id.
4. Validate that the memo belongs to the authenticated user.
5. Query the memos table for full content.
6. Update the last\_opened\_at timestamp.
7. Return full memo content.
8. Handle and log memo not found or access errors.

## 4. List Memos API

1. Authenticate the request using JWT.
2. Receive user\_id as a path parameter.
3. Log the request.
4. Validate user\_id matches token claims.
5. Query PostgreSQL for all memos belonging to user.
6. Return a list of memos.
7. Log and handle database errors.

## 5. List Recent Memos API

1. Authenticate the request using JWT.
2. Receive user\_id as a path parameter.
3. Log recent memo list request.
4. Validate user\_id ownership.
5. Query PostgreSQL for last two updated memos.
6. Return recent memo data as JSON.
7. Handle and log errors.

## 6. Get Client Names API

1. Authenticate the request using JWT.
2. Log request event.
3. Query Snowflake to get distinct client names from client\_attributes\_view.
4. Return list of client names.
5. Handle and log errors from Snowflake.

## 7. Get Client Contacts API

1. Authenticate the request using JWT.
2. Receive client\_name as a query parameter.
3. Log the query.
4. Validate the presence of client\_name.
5. Query Snowflake client\_contacts\_view filtering by client\_name and type.
6. Return list of external contacts.
7. Handle and log errors.

## 8. Update Memo Content API

1. Authenticate the request using JWT.
2. Receive JSON with user\_id, memo\_id, section\_name, and updated\_content.
3. Log update request.
4. Validate that section is editable and memo belongs to user.
5. Update the specific section in PostgreSQL.
6. Insert new version into versions table with updated content.
7. Return success response JSON.
8. Handle validation and DB errors and log appropriately.