

# JobAssist: Backend API Documentation

## (Version 1.2.14 - Finalized)

This document describes the current Flask-based RESTful API, implemented in app.py. It manages all data and documents for job applications. All endpoints are numbered sequentially from 1.0 to 12.0.

## 1. Configuration & Environment

The backend connects to PostgreSQL using the following settings:

Parameter	Value (Example/Default)	Notes
dbname	contact_db	The name of the database.
user	jobert	The PostgreSQL user.
password	linkedin	<b>SECURITY NOTE:</b> This password must be changed in a production environment.
host	localhost	The database server host.
UPLOAD_FOLDER	/home/jobert/webapp/contact_app/filestore	Location where files are stored on the server.

## 2. Company Management Endpoints (1.0 - 3.0)

These endpoints manage standardized company profiles in the companies table.

### 1.0 GET /api/companies (Retrieve All Companies)

Retrieves all standardized company profiles.

Detail	Specification
Method	GET
URL	/api/companies

<b>Returns</b>	<b>Structured JSON Object:</b> {"status": "success", "companies": [...]}
----------------	--

## 2.0 GET /api/next\_company (Get Next Raw Name to Standardize)

Retrieves the next unstandardized company name that needs review.

Detail	Specification
<b>Method</b>	GET
<b>URL</b>	/api/next_company
<b>Authentication</b>	Mandatory (Assumed to be protected)
<b>Returns</b>	<b>JSON Object</b> containing the raw company name.

## 3.0 GET /api/companies/{company\_id}

**Detail:** Retrieves a single, standardized company profile using its unique integer ID. This endpoint is used by the frontend to display detailed company information and pre-populate the standardization form.

Detail	Specification
<b>Method</b>	GET
<b>URL</b>	/api/companies/{company_id}
<b>Authentication</b>	Optional (Used primarily for data retrieval)
<b>Returns</b>	Structured JSON object containing the company profile under the company key.

Success Response Body (200 OK):

The response is wrapped in a consistent structure. The company object contains all standardized company fields:

Field Name	Type	Description
company_id	Integer	Unique identifier for the company profile.
company_name_clean	String	Standardized company name.
headquarters	String (Nullable)	Location of company headquarters.
size_employees	Integer (Nullable)	Total employee count.
annual_revenue	Float (Nullable)	Company's annual revenue value.
revenue_scale	String (Nullable)	Revenue scale (e.g., 'M', 'B').
target_interest	Boolean	High-priority flag.
notes	String (Nullable)	Detailed notes or commentary.

Example Successful Response (200 OK):

```
{
  "status": "success",
  "company": {
    "company_id": 197,
    "company_name_clean": "Alphabet Inc.",
    "headquarters": "Mountain View, CA",
    "size_employees": 156000,
    "annual_revenue": 282.8,
    "revenue_scale": "B",
    "target_interest": true,
    "notes": "Verified against 2024 Q3 earnings report."
  }
}
```

Verification Test Sequence (cURL)

Use this command to retrieve the detailed profile for test company ID 197:

```
curl -X GET http://localhost:8000/api/companies/197
```

Expected Response (Example - 404 Not Found):

If the ID does not exist, the endpoint returns the standard error wrapper:

```
{
  "status": "error",
  "message": "Company ID 999 not found."
}
```

## 4.0 PUT /api/companies/{company\_id}

**Detail:** Updates an existing standardized company profile record in the companies table. This is the main endpoint used by the standardization tool to persist clean, structured data.

Detail	Specification
Method	PUT
URL	/api/companies/{company_id}
Authentication	Mandatory (Verifies user is logged in/authorized for standardization)
Returns	JSON object with status, message, and company_id.

**Request Body** (JSON - Includes all updatable fields):

Field Name	Type	Description
company_name_clean	String	The primary, standardized name for the company.
headquarters	String (Nullable)	The standardized location of the company headquarters.
size_employees	Integer (Nullable)	The total number of

		employees.
<b>annual_revenue</b>	Float (Nullable)	The company's annual revenue value.
<b>revenue_scale</b>	String (Nullable)	The scale of the revenue (e.g., 'M', 'B').
<b>target_interest</b>	Boolean	Flag indicating if this is a high-priority target company.
<b>notes</b>	String (Nullable)	<b>New:</b> Detailed notes or commentary on the company.

## Verification Test Sequence (cURL)

Use the following sequence to test that all fields, especially **notes**, are successfully updated and retrieved for a test ID (e.g., 197):

### Step 1: Execute the PUT Request

This command updates company 197 and includes the test value for notes.

```
curl -X PUT http://localhost:8000/api/companies/197 \
-H "Content-Type: application/json" \
-d '{
  "company_id": 197,
  "company_name_clean": "Alphabet Inc.",
  "headquarters": "Mountain View, CA",
  "size_employees": 156000,
  "annual_revenue": 282.8,
  "revenue_scale": "B",
  "target_interest": true,
  "notes": "Updated via cURL to test the full PUT payload, including notes field persistence."
}'
```

### Expected Response (200 OK):

```
{
  "company_id": 197,
```

```
"message": "Company ID 197 updated successfully.",
"status": "success"
}
```

## Step 2: Verify with a GET Request

This command retrieves the record to confirm the changes were saved, specifically looking for the new notes content.

```
curl -X GET http://localhost:8000/api/companies/197
```

### Expected Response (Example - 200 OK):

```
{
  "company_id": 197,
  "company_name_clean": "Alphabet Inc.",
  "headquarters": "Mountain View, CA",
  "size_employees": 156000,
  "annual_revenue": 282.8,
  "revenue_scale": "B",
  "target_interest": true,
  "notes": "Updated via cURL to test the full PUT payload, including notes field persistence."
}
```

# 5.0 GET /api/companies/{companyId}/raw\_names (Raw Names for Standardization Review)

This endpoint retrieves a list of all user-entered "**raw**" **company names** that have been successfully mapped (standardized) to the specified clean Company Profile ID. This is a critical feature for the Cleanup and Management tools, allowing users to see which raw inputs are linked to a single, clean company entry.

## Specification

Detail	Specification
--------	---------------

<b>Method</b>	GET
<b>URL</b>	/api/companies/{companyId}/raw_names
<b>Authentication</b>	Mandatory (Requires Bearer Token - MOCK_USER_ID is currently used)
<b>Path Parameter</b>	<b>companyId</b> (Integer): The unique ID of the standardized company profile.
<b>Query Parameters</b>	None
<b>Returns</b>	JSON object containing the target company_id and a list of raw_names objects.

## Request Example (cURL)

Retrieve all raw names mapped to Company ID 310 (e.g., "Google").

```
curl -X GET
"http://192.168.56.4/api/companies/310/raw_names"(http://192.168.56.4/api/companies/310/raw_names) \
-H "Authorization: Bearer MOCK_TOKEN" \
-H "Content-Type: application/json"
```

## Success Response (Status 200 OK)

The response provides the clean company ID, a message indicating the number of names found, and an array (raw\_names) of the raw company names that have been linked to the clean profile. If no raw names are mapped, the raw\_names array will be empty.

```
{
  "status": "success",
  "company_id": 310,
  "message": "Retrieved 4 raw names mapped to company 310.",
  "raw_names": [
    {
      "mapping_id": 1004,
      "raw_name": "Google",
      "source_filename": "linkedin_imports_q3_2025.csv",
      "date_imported": "2025-09-01T12:00:00"
    },
  ],
}
```

```

{
  "mapping_id": 1005,
  "raw_name": "Alphabet Inc.",
  "source_filename": "linkedin_imports_q3_2025.csv",
  "date_imported": "2025-09-01T12:00:00"
},
{
  "mapping_id": 1006,
  "raw_name": "Google Careers Site",
  "source_filename": "monster_resume_data.csv",
  "date_imported": "2025-10-15T09:30:00"
},
{
  "mapping_id": 1007,
  "raw_name": "googel",
  "source_filename": "monster_resume_data.csv",
  "date_imported": "2025-10-15T09:30:00"
}
]
}

```

## Error Responses

Status Code	Example Error Message	Description
<b>400 Bad Request</b>	{"status": "error", "message": "Invalid company ID format."}	Occurs if the path parameter is not a positive integer.
<b>404 Not Found</b>	{"status": "error", "message": "Company profile with ID 9999 not found."}	The requested companyId does not exist in the companies table.
<b>500 Server Error</b>	{"status": "error", "message": "Database error during raw name retrieval."}	A server-side or database connection failure occurred.



## 6.0 POST /api/map/existing (Map Raw Name to Existing Company)

This endpoint is used when a user reviews an unstandardized raw company name (e.g., "Google LLC") and determines that it should be mapped to an existing standardized company profile (e.g., "Google").

This is a simple, single-step **UPDATE** transaction. It is the primary way standardization is performed after a clean profile has been created (using Endpoint 7.0 or a previously executed standardization action).

### Specification

Detail	Specification
Method	POST
URL	/api/map/existing
Authentication	Mandatory (Requires Bearer Token)
Returns	JSON object confirming the mapping.

### Request Body Fields (JSON Payload)

Field Name	Type	Description	Required	Notes
raw_name_id	Integer	The unique ID of the unstandardized name record (e.g., 1004).	Yes	This is the record being updated.
company_id	Integer	The unique ID of the existing standardized profile (e.g., 310 for "Google").	Yes	This value is populated in the company_name_mapping table.

## cURL Request Example

This example maps the raw name with **raw\_name\_id: 1004** (e.g., the text was "goog corp") to the existing standardized company profile with **company\_id: 310** ("Google").

```
curl -X POST "[http://192.168.56.4/api/map/existing](http://192.168.56.4/api/map/existing)" \
-H "Authorization: Bearer MOCK_TOKEN" \
-H "Content-Type: application/json" \
-d '{
  "raw_name_id": 1004,
  "company_id": 310
}'
```

## Success Response (Status 200 OK)

A successful request confirms the update has been applied.

```
{
  "status": "success",
  "message": "Raw name ID 1004 successfully mapped to existing company ID 310."
}
```

## Failure Responses

Status Code	Example Error Message	Description
<b>400 Bad Request</b>	{"status": "error", "message": "raw_name_id and company_id are required."}	Missing required fields in the payload.
<b>404 Not Found</b>	{"status": "error", "message": "Mapping not updated. Raw Name ID not found or already mapped."}	The raw_name_id provided either does not exist or already has a company_id assigned. The update was not executed.
<b>500 Server Error</b>	{"status": "error", "message": "Database error during existing map."}	A general database or server-side failure occurred.

## 7.0 POST /api/map/new (Create New Company and Map Raw Name)

This endpoint is the most robust way to standardize a raw company name when no existing clean company profile matches it. It performs a single, **atomic database transaction** involving two critical steps:

1. **Creation:** Inserts a new standardized company profile into the **companies** table using the provided `company_name_clean`.
2. **Mapping:** Updates the specified **raw\_name\_id** in the `company_name_mapping` table to link it to the newly generated `company_id`.

### Specification

Detail	Specification
Method	POST
URL	/api/map/new
Authentication	Mandatory (Requires Bearer Token)
Returns	JSON object with the new <code>company_id</code> .

### Request Body Fields (JSON Payload)

Field Name	Type	Description	Required	Notes
<code>raw_name_id</code>	Integer	The unique ID of the unstandardized name record to be mapped.	Yes	Retrieved from Endpoint 2.0 or 4.0.
<code>company_name_clean</code>	String	The finalized, clean, official name to be used for the new company profile.	Yes	This value populates <code>companies.company_name_clean</code> .

## cURL Request Example

This example assumes the user has an unmapped raw name with raw\_name\_id: 1003 (e.g., the text was "A\_mazOn co.") and wants to create the new standardized company "Amazon" and map the raw name to it.

```
curl -X POST "[http://192.168.56.4/api/map/new](http://192.168.56.4/api/map/new)" \
-H "Authorization: Bearer MOCK_TOKEN" \
-H "Content-Type: application/json" \
-d '{
  "raw_name_id": 1003,
  "company_name_clean": "Amazon"
}'
```

## Success Response (Status 201 Created)

The endpoint confirms success and returns the new unique company ID.

```
{
  "status": "success",
  "message": "New company ID 312 created and raw name ID 1003 mapped successfully.",
  "company_id": 312
}
```

## 8.0 POST /api/map/self (Map Raw Name to Self / No Standardization Needed)

This endpoint is a standardization action used when the **raw company name** is already in its optimal, clean format (e.g., "Apple Inc."). Instead of requiring the user to manually copy the raw name into the "new company" endpoint, this performs the atomic transaction automatically:

1. **Creation:** Inserts a new standardized company profile into the companies table using the submitted raw\_name as the company\_name\_clean.
2. **Mapping:** Updates the specified raw\_name\_id in the company\_name\_mapping table to link it to the newly generated company\_id.

This action effectively creates a **new, clean profile** from the existing raw data, simultaneously mapping the raw name to it.

## Specification

Detail	Specification
Method	POST
URL	/api/map/self
Authentication	Mandatory (Requires Bearer Token)
Returns	JSON object with the new company_id.

## Request Body Fields (JSON Payload)

Field Name	Type	Description	Required	Notes
raw_name_id	Integer	The unique ID of the unstandardized name record to be mapped.	Yes	Retrieved from the "Next Company" endpoint (2.0).
raw_name	String	The exact raw name string (e.g., "Meta Platforms, Inc."). This is used to create the new company_name_clean profile.	Yes	Must match the raw name text for confirmation.

## cURL Request Example

This example assumes the user has an unmapped raw name with **raw\_name\_id: 1005** where the name itself is "**Coca-Cola**". The goal is to create a new profile called "Coca-Cola" and map the raw name to it.

```
curl -X POST "[http://192.168.56.4/api/map/self](http://192.168.56.4/api/map/self)" \
-H "Authorization: Bearer MOCK_TOKEN" \
-H "Content-Type: application/json" \
-d '{
  "raw_name_id": 1005,
```

```
"raw_name": "Coca-Cola"
},
```

## Success Response (Status 201 Created)

The endpoint confirms the creation of the new company and the successful mapping.

```
{
  "status": "success",
  "message": "New company ID 313 created (using raw name) and raw name ID 1005 mapped successfully.",
  "company_id": 313
}
```

## Failure Responses

Status Code	Example Error Message	Description
<b>400 Bad Request</b>	<pre>{"status": "error", "message": "raw_name_id and raw_name are required."}</pre>	Missing required fields in the payload.
<b>404 Not Found</b>	<pre>{"status": "error", "message": "Mapping not updated. Raw Name ID not found or already mapped. New company was not created."}</pre>	The transaction was rolled back because the raw_name_id was already mapped or didn't exist.
<b>500 Server Error</b>	<pre>{"status": "error", "message": "Database error during self map."}</pre>	A database or internal processing failure occurred, and the transaction was rolled back.

# 4. Application Management Endpoints

## (9.0 - 12.0)

These endpoints manage job application records.

### CRUD /api/applications/uuid:application\\_id (Application Record Management)

This endpoint is used for managing the lifecycle of a single application record using its **UUID**.

Method	Purpose	Request Body	Returns
<b>GET</b>	Retrieves a single, full application record, including nested company_info and job_title_info.	None	Full Application JSON Object.
<b>PUT</b>	Updates all fields for the specified application (e.g., status, date applied).	<b>JSON:</b> Requires the full payload (same structure as 10.0 POST) with updated values.	Status JSON: {"status": "success", "message": "..."} 
<b>DELETE</b>	Permanently deletes the application record and all associated documents.	None	Status JSON: {"status": "success", "message": "..."} 

### 9.0 POST /api/application/uuid:application\\_id/documents (Document Upload)

**CRITICAL FIX APPLIED.** Uploads a new document (file) associated with an existing job application.

Detail	Specification	Notes
Method	POST	

<b>URL</b>	/api/application/{application_id}/documents	{application_id} must be a valid <b>UUID</b> .
<b>Content-Type</b>	<b>multipart/form-data</b>	<b>CRITICAL:</b> The request must be submitted as form data.

### Request Body Fields (Form Data) - FINAL CORRECTION

Field Name	Type	Required	Description
<b>document</b>	Binary File	Yes	<b>FIXED:</b> The actual file content to be stored (must use this name for request.files).
<b>document_type_code</b>	String (ENUM)	Yes	<b>FIXED:</b> The document classification code (must use this name for request.form).

### Example curl Request

```
curl -X POST
"[https://api.jobassist.com/api/application/a1b2c3d4-e5f6-7890-1234-567890abcdef/documents](https://api.jobassist.com/api/application/a1b2c3d4-e5f6-7890-1234-567890abcdef/documents)" \
-H "Authorization: Bearer <YOUR_AUTH_TOKEN>" \
-F "document=@./my_resume.pdf" \
-F "document_type_code=RESUME"
```

## 10.0 POST /api/applications (Application Creation)

This endpoint creates a new application entry for the authenticated user, linking it to a standardized company profile and either reusing an existing job title or creating a new one in



the job\_titles table.

### Request Body Fields (JSON Payload)

Field Name	Type	Description	Required	Notes
<b>company_id</b>	Integer	The unique ID of the standardized company profile this application is for (e.g., from the companies table).	<b>Yes</b>	Must be a valid, existing company_id.
<b>job_title_name</b>	String (Max 255)	The full, human-readable job title (e.g., "Senior Data Analyst").	<b>Yes</b>	The backend is responsible for creating a new entry in job_titles or finding the existing job_title_id based on this name.
<b>date_applied</b>	String	The date the user submitted the application.	<b>Yes</b>	Must be in the standard YYYY-MM-DD format.
<b>current_status</b>	String (Max 50)	The initial status of the application.	<b>Yes</b>	Typically "Applied" or a similar starting state defined by the client.

### Request Body Example

```
{  
  "company_id": 420,  
  "job_title_name": "Software Engineering Intern - 2026 Summer",  
}
```

```
"date_applied": "2025-11-06",  
"current_status": "Applied"  
}
```

### Success Response (Status 201 Created)

A successful response returns the unique ID (UUID) of the newly created application record.

```
{  
  "status": "success",  
  "message": "Application created successfully.",  
  "application_id": "8b51d8b0-a3f2-4e4b-9c7f-05f7e7f8e8e9"  
}
```

## 11.0 APPLICATION AGGREGATE API: GET /api/applications

This endpoint retrieves a comprehensive list of job applications for the authenticated user, filtered by a specific company. The response is highly structured, nesting company, job title, and document information within each application record.

### Detail Specification

Detail	Specification	Notes
Method	GET	
URL	/api/applications	
Query Param	company_id=<int>	<b>Mandatory.</b> Must be a valid integer ID for the target company.
Authentication	Mandatory	Requires a valid user session (User ID filter enforced on all returned data).
Response	JSON Object	Encapsulated response

		with root key applications.
--	--	-----------------------------

# cURL Testing Example

This command tests the endpoint using a placeholder Company ID (24) and requires an Authorization header with a valid user token.

```
# Replace <YOUR_AUTH_TOKEN> with a valid user session token.
# Replace 24 with a company_id known to have applications.
curl -X GET \
  -H "Authorization: Bearer <YOUR_AUTH_TOKEN>" \
  "http://localhost:5000/api/applications?company_id=24"
```

# Response Schema (JSON)

The response is a structured object containing a list of application records.

```
{
  "status": "success",
  "applications": [
    {
      "application_id": "e7b0c5d6-...",
      "date_applied": "2025-10-25",
      "current_status": "INTERVIEW",

      "company_info": {
        "company_id": 24,
        "company_name_clean": "3545 Consulting-Global"
        // CRITICAL: Frontend must access this field explicitly.
      },

      "job_title_info": {
        "job_title_id": 150,
        "title_name": "Senior Data Analyst"
        // CRITICAL: Frontend must access this field explicitly.
      },

      "documents": [
        {
          "document_id": "4a1d82f0-...",
          "document_type": "RESUME",
```

```

    "file_path": "4a1d82f0-...",
    "original_filename": "My_Resume_Q4.pdf"
  },
  // ... more document objects
]
},
// ... more application objects
]
}

```

## Application Object Fields

Field	Type	Description	Frontend Access Path (Example)
application_id	UUID (String)	Unique ID for the application record.	app.application_id
date_applied	Date (String)	The date the user applied (YYYY-MM-DD).	app.date_applied
current_status	String	Current status of the application (e.g., APPLIED, INTERVIEW).	app.current_status
company_info	Object	<b>Nested object</b> containing standardized company data.	
job_title_info	Object	<b>Nested object</b> containing standardized job title data.	
documents	Array	<b>Nested array</b> of all documents uploaded for this application.	

## Nested company\_info Object

Field	Type	Description	CRITICAL Access Path
company_id	Integer	Unique ID of the standardized company.	app.company_info.company_id
company_name_clean	String	<b>The standardized, human-readable name.</b>	app.company_info.company_name_clean

## Nested job\_title\_info Object

Field	Type	Description	CRITICAL Access Path
job_title_id	Integer	Unique ID of the standardized job title.	app.job_title_info.job_title_id
title_name	String	<b>The human-readable job title name.</b>	app.job_title_info.title_name

## 12.0 GET /api/documents/{document\_id} (Document Download)

Securely retrieves the binary file content for a specific document.

Detail	Specification	Notes
Method	GET	
URL	/api/documents/{document_id}	{document_id} must be the <b>UUID</b> returned by Endpoint

		9.0 or retrieved via 11.0.
<b>Authentication</b>	Mandatory	Server performs an <b>ownership check</b> against the document ID.
<b>Response</b>	<b>Binary File Stream</b>	The response is the raw file content (application/octet-stream), <i>not</i> a JSON object.
<b>Header</b>	Content-Disposition: attachment	This forces the browser to download the file using the original filename.

## CRITICAL: Download URL Construction

The frontend team must construct the full URL by appending the **document\_id** (which is a UUID string) to the base path. **Do not use the application\_id here.**

Parameter	Example Value
<b>Base URL</b>	https://api.jobassist.com/
<b>Document ID</b>	9f142b78-c1a7-4d43-9e45-12a8f89c6d30

Full, Formatted Download URL Example:

[BASE\_URL]/api/documents/9f142b78-c1a7-4d43-9e45-12a8f89c6d30

## Error Response

If the document is not found, or if the authenticated user does not own the application associated with the document, the server will return a standard error object with a **404 Not Found** status to prevent unauthorized probing of existing document IDs.

```
{
  "status": "error",
  "message": "File not found or unauthorized access."
}
```

## 13.0 GET /api/companies/int:company\_id/contacts

Retrieves a list of all contact records that are associated with a **standardized company profile**. The association is made through the company\_name\_mapping table, linking the contacts.company (raw name) field to the standardized companies.company\_id.

## Detail Specification

Detail	Specification
Method	GET
URL	/api/companies/<int:company_id>/contacts
Authentication	Mandatory (A user session is assumed, although the user_id check is currently bypassed in the SQL query due to database schema limitations).
URL Parameter	<b>company_id</b> (Integer, required in URL path)
Purpose	Feeds a detail view or contact list for a specific, standardized company profile.

## CURL Example

Assuming the server is running on localhost:5000 and company\_id is 1234. The inclusion of a header like Authorization is mandatory for authenticated endpoints, even if the backend uses a mock user ID.

```
curl -X GET 'http://localhost:5000/api/companies/1234/contacts' \  
-H 'Authorization: Bearer <Your_Auth-Token>'
```

## Successful Response (HTTP 200 OK)

The endpoint adheres to the universal rule for array returns, wrapping the list in a structured object.

```
{  
  "status": "success",  
  "company_id": 1234,  
  "contacts": [  
    {  
      "contact_id": 1,  

```

```

    "first_name": "Jane",
    "last_name": "Doe",
    "email_address": "jane.doe@example.com",
    "position": "Senior Developer",
    "connected_on": "2024-05-15",
    "associated_raw_name": "Example Corp"
  },
  {
    "contact_id": 2,
    "first_name": "John",
    "last_name": "Smith",
    "email_address": "john.smith@excorp.com",
    "position": "Talent Acquisition",
    "connected_on": "2024-05-10",
    "associated_raw_name": "Ex. Corp."
  }
]
}

```

## Response Fields

Field Name	Type	Description
<b>status</b>	String	Always "success".
<b>company_id</b>	Integer	The standardized company ID requested in the URL.
<b>contacts</b>	Array of Objects	The list of contact records associated with the company.
<b>contact_id</b>	Integer	Unique primary key for the contact.
<b>position</b>	String	The contact's job title at the time of connection.
<b>connected_on</b>	Date	The date the connection was established (YYYY-MM-DD format).



<b>associated_raw_name</b>	String	The original, raw company name from the contacts table that mapped to this company ID.
----------------------------	--------	--

### Error Responses

Status Code	Reason	Description
<b>400 Bad Request</b>	Invalid company_id format (e.g., negative or non-integer).	Returns {"status": "error", "message": "Invalid company ID format."}
<b>500 Internal Error</b>	Database connection failure or PostgreSQL execution error (e.g., table schema issues).	Returns {"status": "error", "message": "Database error retrieving contacts: [details]"}

# API Endpoint 14.0: Sidebar Summary

## 14.0 GET /api/sidebar

This endpoint retrieves a concise, minimal list of all standardized **Company Profiles**. The data provided is intentionally limited to facilitate quick loading of client-side UI elements, such as a company navigation sidebar or a quick-select dropdown.

The list is sorted alphabetically by the clean company name.

### Specification

Detail	Specification
<b>Method</b>	GET
<b>URL</b>	/api/sidebar
<b>Authentication</b>	Mandatory (Requires Bearer Token)
<b>Query Parameters</b>	None

<b>Returns</b>	JSON object containing a list of company summary objects.
----------------	---

## Request Example (cURL)

The request requires the Authorization: Bearer header.

```
curl -X GET "[http://192.168.56.4/api/sidebar](http://192.168.56.4/api/sidebar)" \
-H "Authorization: Bearer MOCK_TOKEN"
```

## Success Response (Status 200 OK)

The response provides only the company\_id, the clean name, and the is\_target flag for filtering/coloring in the UI.

```
{
  "status": "success",
  "companies": [
    {
      "company_id": 101,
      "company_name_clean": "Adobe",
      "is_target": false
    },
    {
      "company_id": 205,
      "company_name_clean": "Amazon",
      "is_target": true
    },
    {
      "company_id": 310,
      "company_name_clean": "Google",
      "is_target": true
    },
    {
      "company_id": 402,
      "company_name_clean": "Meta Platforms, Inc.",
      "is_target": false
    }
  ]
}
```

## Response Fields

Field Name	Type	Description
company_id	Integer	The unique ID of the clean company profile.
company_name_clean	String	The standardized name used for display.
is_target	Boolean	Maps to the target_interest column. true if this company is a high-priority target.

## Error Response

Status Code	Example Error Message
401 Unauthorized	Authentication required.
500 Internal Server Error	Database error retrieving sidebar data.

# API Endpoint 15.0: Unmapped Company Names List

This endpoint is used in the Company Name Standardization workflow. It retrieves a comprehensive list of all company names entered by users that have **not yet** been successfully mapped (standardized) to a canonical Company Profile ID. This list provides the necessary database ID (raw\_name\_id) to perform subsequent standardization actions (Endpoints 6.0, 7.0, and 8.0).

## Specification

Detail	Specification
Method	GET

<b>URL</b>	/api/unmapped_list
<b>Authentication</b>	Mandatory (Requires Bearer Token)
<b>Path/Query Parameters</b>	None
<b>Returns</b>	JSON object containing an array of unmapped company name objects.

## Request Example (cURL)

This request retrieves the full list of raw company names pending standardization.

```
curl -X GET "[http://192.168.56.4/api/unmapped_list](http://192.168.56.4/api/unmapped_list)" \
-H "Authorization: Bearer MOCK_TOKEN"
```

## Success Response (Status 200 OK)

The response body contains the list of unmapped names. Each item in the raw\_names array is an object that includes the raw\_name\_id—the unique identifier needed for all mapping operations.

```
{
  "status": "success",
  "raw_names": [
    {
      "raw_name_id": 1001,
      "raw_name": "googel inc"
    },
    {
      "raw_name_id": 1002,
      "raw_name": "A-C-M-E-CORP"
    },
    {
      "raw_name_id": 1003,
      "raw_name": "The WALtmAn Co"
    }
  ]
}
```

## Response Fields

Field Name	Type	Description
<b>status</b>	String	Always "success".
<b>raw_names</b>	Array of Objects	The list of companies pending standardization.
<b>raw_name_id</b>	Integer	The unique primary key used to identify this unmapped name in the company_name_mapping table. <b>Crucial for all standardization API calls.</b>
<b>raw_name</b>	String	The original, unstandardized company name (e.g., "googel").

## 15.0 GET /api/unmapped\_list (Retrieve Unmapped List)

### Specification Changes

Detail	Old Value (Assumed)	New Value (Actual)
raw_name_id Type	Integer	<b>String</b> (Contains the raw_name string)
Data Source	company_name_mapping table	contacts table (LEFT JOIN to find unmapped names)

### Request Example (cURL)

This command retrieves the list of raw company names currently pending standardization.

```
curl -X GET "http://192.168.56.4/api/unmapped_list" \  
-H "Authorization: Bearer MOCK_TOKEN"
```

## Success Response (Status 200 OK)

The array contains the unmapped raw name twice: once as the user-facing name, and once as the key used for mapping (aliased as `raw_name_id`).

```
{
  "status": "success",
  "raw_names": [
    {
      "raw_name_id": "googel inc",
      "raw_name": "googel inc"
    },
    {
      "raw_name_id": "A-C-M-E-CORP",
      "raw_name": "A-C-M-E-CORP"
    }
  ]
  // ... more unmapped entries
}
```

## A. Job Title Endpoints ()

These endpoints manage standardized job titles.

### A.0 GET `/api/job_titles` (Retrieve All Titles) - NOT IMPLEMENTED

Retrieves all standardized job titles currently stored in the system.

Detail	Specification
Method	GET
URL	<code>/api/job_titles</code>
Returns	<b>Structured JSON Object:</b> <code>{"status": "success", "job_titles": [...]}</code>

### B.0 GET `/api/job_titles/int:title_id` (Retrieve Single Title) - NOT IMPLEMENTED

Retrieves a single job title record by its ID (Inferred Standard).

### C.0 POST/PUT/DELETE `/api/job_titles/int:title_id` (CRUD) - NOT

# IMPLEMENTED

Standard CRUD operations for job titles (Inferred Standard).

## 5. Standard Response Schemas

### Successful List Response (Status 200 OK)

All successful API calls returning a list of resources use this structured wrapper.

```
{
  "status": "success",
  "[resource_name]s": [ /* array of objects */ ]
}
```

### Successful Creation Response (Status 201 Created)

Field Name	Type	Description
status	String	Always "success".
message	String	Human-readable confirmation.
document_id / application_id	UUID/Integer	The unique ID of the newly created resource.

### Error Response Schema (All Failures)

All error responses return a JSON object for reliable client-side error handling.

Status Code	Example Error Message	Client-Side Interpretation
400 Bad Request	Missing required field: document_type_code	Client-side payload issue.
401 Unauthorized	Authentication required.	User session is missing or invalid.
404 Not Found	File not found or unauthorized access.	Resource not found or ownership check failed.

# API Endpoint 16.0: Company Profile Search

This endpoint provides real-time search suggestions for existing, **standardized company profiles** in the companies table. It is primarily used during the "Map to Existing Clean Profile" standardization workflow.

## Endpoint Details

Detail	Specification
Method	GET
URL	/api/search/company
Authentication	<b>Mandatory</b> (Requires Authorization: Bearer [TOKEN] header)
Search Target	The company_name_clean column in the companies table.
Result Limit	Maximum of <b>10</b> suggestions.

## Request Parameters

This endpoint requires one query parameter.

Parameter	Type	Required	Description
query	string	Yes	The search term provided by the user. Must be at least <b>2 characters</b> long.

## Success Response (HTTP 200 OK)

The response returns a list of matching clean company profiles, sorted by relevance: **Exact**



**Match > Prefix Match > Contains Match.**

## Schema

```
{
  "status": "success",
  "companies": [
    {
      "company_id": "101",
      "company_name_clean": "Google"
    },
    {
      "company_id": "102",
      "company_name_clean": "Google Cloud"
    },
    // ... up to 10 results
  ]
}
```

## Error Responses

HTTP Status Code	Response Body Schema	Description
<b>401 Unauthorized</b>	<pre>{"status": "error", "message": "Authentication required or invalid token."}</pre>	Missing or invalid authentication token.
<b>400 Bad Request</b>	<pre>{"status": "error", "message": "Query must be at least 2 characters long."}</pre>	The query parameter was missing or shorter than 2 characters.
<b>500 Internal Server Error</b>	<pre>{"status": "error", "message": "..."} </pre>	General server or database error.

## cURL Example

This example demonstrates searching for companies that match "micros". The placeholder [YOUR\_AUTH\_TOKEN] should be replaced with a valid Bearer token.

```
curl -X GET \
  "http://localhost:5000/api/search/company?query=micros" \
```

```
-H "accept: application/json" \  
-H "Authorization: Bearer [YOUR_AUTH_TOKEN]"
```

### Example Output (Expected)

```
{  
  "status": "success",  
  "companies": [  
    {  
      "company_id": "1005",  
      "company_name_clean": "Microsoft"  
    },  
    {  
      "company_id": "1006",  
      "company_name_clean": "Microsoft Azure"  
    },  
    {  
      "company_id": "1010",  
      "company_name_clean": "Micro Solutions Group"  
    }  
  ]  
}
```

## API 17.0 POST /api/companies (Create New Company Profile)

This endpoint is used to create a new standardized company profile record in the companies table. It returns the ID of the newly created company.

### Endpoint Specification

Detail	Specification
Method	POST
URL	/api/companies
Authentication	Mandatory (Authorization: Bearer [TOKEN])
Purpose	Create a new standardized company

	profile.
Database Table	companies

## Request Body

The request body contains the data for the new company profile. All fields except company\_name\_clean are optional.

Field (JSON Key)	Type	Required	Description	Database Column
company_name_clean	string	Yes	The official, standardized name of the company.	company_name_clean
headquarters	string	No	The primary location of the company.	headquarters
size_employees	integer	No	The number of employees.	size_employees
annual_revenue	numeric	No	The annual revenue figure.	annual_revenue
revenue_scale	string (e.g., M, B)	No	The scale of the revenue (e.g., Millions, Billions).	revenue_scale
notes	text	No	Internal notes about the company.	notes
is_target	boolean	No	Flag indicating if this company is a current target (Default: false).	target_interest

			<b>Backend maps this field to the target_interest column.</b>	
website_url	string	No	<b>Note:</b> This field is accepted in the JSON but is ignored as it is not present in the current companies table schema.	N/A

## Success Response (HTTP 201 Created)

```
{
  "status": "success",
  "message": "Company profile created successfully.",
  "company_id": 12345
}
```

## Example cURL Command

This command demonstrates creating a new company profile with several optional fields.

```
curl -X POST http://localhost:8000/api/companies \
--header 'Content-Type: application/json' \
--header 'Authorization: Bearer MOCK_TOKEN' \
--data-raw '{
  "company_name_clean": "Innovatech Solutions Inc.",
  "headquarters": "Seattle, WA",
  "size_employees": 550,
  "annual_revenue": 85.5,
  "revenue_scale": "M",
  "notes": "High-growth prospect in the cloud sector.",
  "is_target": true
}'
```

# API 18.0 DELETE

## /api/companies/{company\_id} (Soft Delete Company Profile)

This endpoint performs a soft deletion of a standardized company profile. It removes the record from the companies table while preserving related historical data (raw names and applications) by nullifying the foreign key references. This ensures that raw data remains available for future re-standardization or analysis.

### Endpoint Specification

Detail	Specification
Method	DELETE
URL	/api/companies/{company_id}
Authentication	Mandatory (Authorization: Bearer [TOKEN])
Purpose	Deletes the standardized company profile and <b>nullifies</b> its foreign key links in other tables.

### Database Actions

This operation must be performed as a single transaction to ensure data integrity:

1. **UPDATE company\_name\_mapping:** Set company\_id to NULL for all rows linked to the deleted company ID.
2. **UPDATE applications:** Set company\_id to NULL for all rows linked to the deleted company ID.
3. **DELETE** row from the **companies** table where company\_id matches the path parameter.

### Success Response (HTTP 204 No Content)

A successful operation returns an **HTTP 204 No Content** status. No response body is provided, as the action's success is defined by the status code.

Detail	Specification
--------	---------------

<b>HTTP Status Code</b>	204 (No Content)
<b>Response Body</b>	None
<b>Description</b>	The company profile was deleted, and linked company_name_mapping and applications records were successfully disassociated (nullified).

## Error Responses

HTTP Status Code	Description	Response Body (JSON Example)
<b>401 Unauthorized</b>	Missing or invalid authentication token.	{"status": "error", "message": "Authentication required."}
<b>404 Not Found</b>	The specified company_id does not exist in the database.	{"status": "error", "message": "Company profile 1015 not found or already deleted."}
<b>500 Internal Server Error</b>	A database or general server error occurred during the soft deletion process.	{"status": "error", "message": "An unexpected error occurred during profile disassociation."}

## Example cURL Command

This command demonstrates soft deleting the company profile with ID **1015**.

```
curl -X DELETE \
http://localhost:8000/api/companies/1015 \
-H "Authorization: Bearer MOCK_TOKEN"
```

## 19.0 PUT /api/applications/{application\_id} (Update

## Application)

This endpoint allows a user to modify the details of an existing job application. The update process ensures that modifications to the **Company Name** and **Job Title** are standardized by referencing or creating records in the companies and job\_titles tables, respectively.

### Specification

Detail	Specification
Method	PUT
URL	/api/applications/{application_id}
URL Parameter	application_id (UUID) - The unique ID of the application to update.
Authentication	Mandatory (Requires ownership of the application)
Purpose	Update one or more fields of an existing application record.

### Request Body (JSON)

The request body should contain the fields you wish to update. **All fields are optional** when updating, but at least one valid field must be present.

Field	Type	Required	Description
company_name_clean	string	No	The standardized company name. The system will look up the corresponding company_id and update the application's link.
title_name	string	No	The job title. The system will look up the corresponding

			job_title_id. If the title doesn't exist, a new one will be created.
date_applied	string	No	The date the application was submitted, in YYYY-MM-DD format.
current_status	string	No	The current status (e.g., 'APPLIED', 'INTERVIEW', 'OFFER', 'REJECTED').

#### Example Request Body:

```
{
  "current_status": "INTERVIEW",
  "date_applied": "2025-11-18",
  "title_name": "Senior Software Engineer"
}
```

#### Example cURL Command (19.0)

Replace [APPLICATION\_ID\_TO\_UPDATE] with an actual application UUID from your database.

```
curl -X PUT \
  http://localhost:8000/api/applications/[APPLICATION_ID_TO_UPDATE] \
  -H "Authorization: Bearer MOCK_TOKEN" \
  -H "Content-Type: application/json" \
  -d '{
    "current_status": "INTERVIEW",
    "date_applied": "2025-11-18",
    "title_name": "Senior Software Engineer (Updated Title)"
  }'
```

#### Response



HTTP Status Code	Description	Response Body (JSON Example)
200 OK	Application updated successfully, or update successful with no changes detected.	{"status": "success", "message": "Application [UUID] updated successfully."}
400 Bad Request	Missing required request body, invalid date format, or no valid fields provided for update.	{"status": "error", "message": "Invalid date format for 'date_applied'. Expected YYYY-MM-DD."}
404 Not Found	The specified application_id does not exist.	{"status": "error", "message": "Application [UUID] not found."}

## 20.0 DELETE /api/applications/{application\_id} (Delete Application)

This endpoint **permanently deletes** a job application and all associated data in an atomic, owner-restricted transaction.

### CRITICAL ACTION: File Deletion

When an application is deleted, all document records linked to that application in the job\_documents table are removed, and the corresponding **physical files are permanently deleted from the disk storage system**.

### Specification

Detail	Specification
Method	DELETE
URL	/api/applications/{application_id}
URL Parameter	application_id (UUID) - The unique ID of the application to delete.
Authentication	Mandatory (Requires ownership of the

	application)
<b>Purpose</b>	Permanently delete the application, document metadata, and associated files.

## Example cURL Command (20.0)

Replace [APPLICATION\_ID\_TO\_DELETE] with the UUID of the application you wish to remove.

```
curl -X DELETE \
  http://localhost:8000/api/applications/[APPLICATION_ID_TO_DELETE] \
  -H "Authorization: Bearer MOCK_TOKEN" \
  -H "Content-Type: application/json"
```

## Response

HTTP Status Code	Description	Response Body (JSON Example)
<b>200 OK</b>	Application, document metadata, and associated files were deleted successfully.	{"status": "success", "message": "Application [UUID] deleted successfully. X associated file(s) removed."}
<b>403 Forbidden</b>	The application exists but is not owned by the authenticated user.	{"status": "error", "message": "Unauthorized access. This application does not belong to your account."}
<b>404 Not Found</b>	The specified application_id does not exist.	{"status": "error", "message": "Application [UUID] not found."}
<b>500 Server Error</b>	A failure occurred during file or database deletion (e.g., file lock).	{"status": "error", "message": "An unexpected server error occurred during deletion."}

## 6. Version Control

Version	Date	Changes
1.0.0	2025-10-25	Initial documentation for Company Profiles (1.0, 2.0) and Application creation (10.0).
1.1.0	2025-10-27	Added Document Upload (9.0) and Download (12.0) endpoints.
1.2.0	2025-10-30	MAJOR RE-INDEXING and CRITICAL FIELD FIXES. Confirmed correct fields for 9.0 and applied universal response wrappers for all list endpoints.
1.2.1	2025-10-30	Detail Fix: Restored full CRUD specification for Endpoint <b>8.0</b> (GET, PUT, DELETE for single application UUID).
1.2.2	2025-10-30	<b>Documentation Fix:</b> Updated Request Body Fields table for Endpoint <b>9.0</b> to correctly list <b>document</b> and <b>document_type_code</b> for

		100% contract compliance.
<b>1.2.7</b>	<b>2025-11-06</b>	<b>Add new endpoint 14</b>
<b>1.2.8</b>	<b>2025-11-06</b>	<b>Rebuilt endpoint 5 and documented.</b>
<b>1.2.9</b>	<b>2025-11-06</b>	<b>Enriched documentation for Endpoint 10</b>
<b>1.2.10</b>	<b>2025-11-06</b>	<b>Enriched Documentation for Endpoint 7, 8, 9</b>
<b>1.2.11</b>	<b>2025-11-06</b>	<b>Added Documentation for endpoint 15</b>
<b>1.2.12</b>	<b>2025-11-17</b>	<b>Added Documentation for Endpoint 16</b>
<b>1.2.13</b>	<b>2025-11-18</b>	<b>Added Documentation for Endpoint 17,18</b>
<b>1.2.14</b>	<b>2025-11-18</b>	<b>Added Documentation for Endpoint 19,20</b>