JobAssist: Backend API Documentation (Version 1.2.15 - 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	**SECURITY NOTE:** 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

Returns	**Structured JSON Object**: {"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**
Method	GET
URL	/api/next_company
Authentication	Mandatory (Assumed to be protected)
Returns	**JSON Object** 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**
Method	GET
URL	/api/companies/{company_id}
Authentication	Optional (Used primarily for data retrieval)
Returns	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.
annual_revenue	Float (Nullable)	The company's annual revenue value.
revenue_scale	String (Nullable)	The scale of the revenue (e.g., 'M', 'B').
target_interest	Boolean	Flag indicating if this is a high-priority target company.
notes	String (Nullable)	**New:** 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

Method	GET
URL	/api/companies/{companyId}/raw_names
Authentication	Mandatory (Requires Bearer Token - MOCK_USER_ID is currently used)
Path Parameter	**companyId** (Integer): The unique ID of the standardized company profile.
Query Parameters	None
Returns	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/ra
w_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**
400 Bad Request	{"status": "error", "message": "Invalid company ID format."}	Occurs if the path parameter is not a positive integer.
404 Not Found	{"status": "error", "message": "Company profile with ID 9999 not found."}	The requested companyId does not exist in the companies table.
500 Server Error	{"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
400 Bad Request	{"status": "error", "message": "raw_name_id and company_id are required."}	Missing required fields in the payload.
404 Not Found	{"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.
500 Server Error	{"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 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 Endpoint 2.0 or 4.0.
company_name_clean	String	The finalized, clean, official name to be used for the new company profile.	**Yes**	This value populates companies.company_name_clean.

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**
400 Bad Request	{"status": "error", "message": "raw_name_id and raw_name are required."}	Missing required fields in the payload.
404 Not Found	{"status": "error", "message": "Mapping not updated. Raw Name ID not found or already mapped. New company was not created."}	The transaction was rolled back because the raw_name_id was already mapped or didn't exist.
500 Server Error	{"status": "error", "message": "Database error during self map."}	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
GET	Retrieves a single, full application record, including nested company_info and job_title_info.	None	Full Application JSON Object.
PUT	Updates all fields for the specified application (e.g., status, date applied).	**JSON**: Requires the full payload (same structure as 10.0 POST) with updated values.	Status JSON: {"status": "success", "message": "..."}
DELETE	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	

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

Request Body Fields (Form Data) - FINAL CORRECTION

Field Name	**Type**	**Required**	**Description**
document	Binary File	Yes	**FIXED:** The actual file content to be stored (must use this name for request.files).
document_type_code	String (ENUM)	Yes	**FIXED:** 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
company_id	Integer	The unique ID of the standardized company profile this application is for (e.g., from the companies table).	**Yes**	Must be a valid, existing company_id.
job_title_name	String (Max 255)	The full, human-readable job title (e.g., "Senior Data Analyst").	**Yes**	The backend is responsible for creating a new entry in job_titles or finding the existing job_title_id based on this name.
date_applied	String	The date the user submitted the application.	**Yes**	Must be in the standard YYYY-MM-DD format.
current_status	String (Max 50)	The initial status of the application.	**Yes**	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>	**Mandatory.** 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	**Nested object** containing standardized company data.	
job_title_info	Object	**Nested object** containing standardized job title data.	
documents	Array	**Nested array** 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	**The standardized, human-readable name.**	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	**The human-readable job title name.**	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 **UUID** returned by Endpoint

		9.0 or retrieved via 11.0.
Authentication	Mandatory	Server performs an **ownership check** against the document ID.
Response	**Binary File Stream**	The response is the raw file content (application/octet-stream), *not* a JSON object.
Header	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**
Base URL	https://api.jobassist.com/
Document ID	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."
}
```

API 13.0: GET

/api/companies/int:company_id/contacts

Purpose

Retrieves a list of all contacts associated with the specified standardized company profile. This includes contacts mapped via any of the company's raw (unstandardized) names.

Endpoint Specification

Detail	**Specification**
Method	GET
URL	/api/companies/<int:company_id>/contacts
Authentication	Mandatory (Authorization: Bearer [TOKEN])
Purpose	Get all contacts linked to a specific company ID.
Required Parameters	Path Parameter: <company_id> (Integer)

Success Response (HTTP 200 OK)

A successful request returns a JSON object containing the company ID and an array of contact records.

Field	**Type**	**Description**
status	string	"success"
company_id	integer	The ID of the company profile used for the query.
contacts	array	A list of contact objects associated with the company.

Contact Object Fields (All in Snake_Case)

Field	Type	Description
contact_id	UUID	The unique identifier for the contact record.
first_name	string	Contact's first name.
last_name	string	Contact's last name.
email_address	string	Contact's primary email address.
position	string	Contact's job title or position.
connected_on	date	The date the user recorded connecting with this person (YYYY-MM-DD).
linkedin_url	**string**	**The LinkedIn profile URL for the contact (standardized to snake_case).**
associated_raw_name	string	The unstandardized name used to link the contact to the company.

Example Success Response Body

```
{
  "status": "success",
  "company_id": 101,
  "contacts": [
    {
      "contact_id": "84d7a8d5-1b4d-4e9c-b8f1-8c4d2e8b23a1",
      "first_name": "Jane",
      "last_name": "Doe",
      "email_address": "jane.doe@example.com",
      "position": "Product Manager",
      "connected_on": "2024-05-15",
      "linkedin_url":
"[https://www.linkedin.com/in/janedoe](https://www.linkedin.com/in/janedoe)",
      "associated_raw_name": "Example Corp"
    },
    {
      "contact_id": "77f3e1b0-2c7a-4f5d-9b0d-1e0f3c5b8a6f",
      "first_name": "Robert",
      "last_name": "Smith",
      "email_address": "robert.smith@acme.com",
      "position": "Recruiter",
      "connected_on": "2024-06-01",
      "linkedin_url":
"[https://www.linkedin.com/in/robertsmithrec](https://www.linkedin.com/in/robertsmithrec)",
      "associated_raw_name": "Acme, Inc."
    }
  ]
}
```

Error Responses

HTTP Status Code	Description	Response Body (JSON Example)
400 Bad Request	Invalid company ID format (e.g., non-integer).	{"status": "error", "message": "Invalid company ID format."}
401 Unauthorized	Missing or invalid authentication token.	{"status": "error", "message": "Authentication required."}
500 Internal Server Error	A general server or database error occurred.	{"status": "error", "message": "Database error retrieving contacts: N/A"}

Example cURL Command

```
# Example cURL command to retrieve all contacts for company profile ID 101

curl -X GET http://localhost:8000/api/companies/101/contacts \
     -H "Authorization: Bearer MOCK_TOKEN"
```

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
Method	GET
URL	/api/sidebar
Authentication	Mandatory (Requires Bearer Token)
Query Parameters	None
Returns	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
URL	/api/unmapped_list
Authentication	Mandatory (Requires Bearer Token)
Path/Query Parameters	None
Returns	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**
status	String	Always "success".
raw_names	Array of Objects	The list of companies pending standardization.
raw_name_id	Integer	The unique primary key used to identify this unmapped name in the company_name_mapping table. **Crucial for all standardization API calls.**
raw_name	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	**String** (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	/api/job_titles
Returns	**Structured JSON Object**: {"status": "success", "job_titles": [...]}

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	**Mandatory** (Requires Authorization: Bearer [TOKEN] header)

Search Target	The company_name_clean column in the companies table.
Result Limit	Maximum of **10** 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 **2 characters** 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
401 Unauthorized	{"status": "error", "message": "Authentication required or invalid token."}	Missing or invalid authentication token.
400 Bad Request	{"status": "error", "message": "Query must be at least 2 characters long."}	The query parameter was missing or shorter than 2 characters.
500 Internal Server Error	{"status": "error", "message": "..."}	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	headquarters

			company.	
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). **Backend maps this field to the target_interest column.**	target_interest
website_url	string	No	**Note:** 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 **nullifies** 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**
HTTP Status Code	204 (No Content)
Response Body	None
Description	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)**
401 Unauthorized	Missing or invalid authentication token.	{"status": "error", "message": "Authentication required."}

404 Not Found	The specified company_id does not exist in the database.	{"status": "error", "message": "Company profile 1015 not found or already deleted."}
500 Internal Server Error	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)
Purpose	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)

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

API 21.0: GET /api/application/{application_id} (Retrieve Single Application)

Purpose

Retrieves all stored data for a single job application identified by its UUID. This comprehensive dataset is required for populating and loading the form fields in the frontend's Edit Mode or for displaying a detailed view of a specific application.

Endpoint Specification

Detail	**Specification**
Method	GET

URL	/api/application/<application_id>
Authentication	Mandatory (Authorization: Bearer [TOKEN]). The authenticated user must be the owner of the application.
Purpose	Get all aggregated data (Application, Job Title, Company, Documents) for a specific application_id.
Required Parameters	Path Parameter: <application_id> (UUID)

Database Query Logic

The implementation will use a multi-JOIN query to aggregate the application, its linked company, its job title, and any associated documents.

```
-- Conceptual Query (Implementation details will vary slightly to group documents)
SELECT
    a.*, -- All application fields
    c.company_name_clean, -- Company Info
    jt.title_name, -- Job Title Info
    jd.document_id, jd.document_type, jd.file_path, jd.original_filename -- Document Info
(multi-row potential)
FROM applications a
LEFT JOIN companies c ON a.company_id = c.company_id
LEFT JOIN job_titles jt ON a.job_title_id = jt.job_title_id
LEFT JOIN job_documents jd ON a.application_id = jd.application_id
WHERE a.application_id = %s AND a.user_id = %s;
```

Success Response (HTTP 200 OK)

A successful request returns a JSON object containing a single, nested application record. Since the application object includes nested information from other tables (Company, Job Title, Documents), the fields are structured for ease of consumption by the frontend.

Field	**Type**	**Description**
status	string	"success"
application	object	The full application data

		object.

Nested Application Object Fields

Field	**Type**	**Description**
application_id	UUID	The unique ID of the application.
user_id	UUID	The ID of the owner.
date_applied	date	The date of the application (YYYY-MM-DD).
current_status	string	The current status.
company_id	integer	The foreign key ID of the standardized company (can be null).
job_title_id	bigint	The foreign key ID of the standardized job title.
company_info	object	Nested object containing company details.
job_title_info	object	Nested object containing job title details.
documents	array	A list of all documents linked to this application.

Example Success Response Body

```
{
  "status": "success",
  "application": {
    "application_id": "4e83460b-39f0-4c00-a1f4-ac41c4994cb4",
    "user_id": "d06c2b39-3d5f-4e8c-acff-8589a6383ecb",
    "date_applied": "2024-07-20",
    "current_status": "ONSITE_INTERVIEW",
    "company_id": 327,
    "job_title_id": 9845,
    "created_at": "2024-07-20T10:00:00Z",
    "updated_at": "2024-07-25T11:30:00Z",
    "company_info": {
      "company_id": 327,
      "company_name_clean": "Acme Corp"
      // Additional fields like headquarters, size_employees, etc., may be included here
    },
    "job_title_info": {
      "job_title_id": 9845,
      "title_name": "Senior Frontend Developer"
      // Additional standardized_title field may be included here
    },
    "documents": [
      {
        "document_id": "a9d1d1f0-5b7c-4e8a-b8f1-8c4d2e8b23a1",
        "document_type": "RESUME",
        "file_path": "a9d1d1f0-5b7c-4e8a-b8f1-8c4d2e8b23a1",
        "original_filename": "MyResume_v3.pdf"
      },
      {
        "document_id": "b1e2c3d4-6f8e-4a9b-c0d2-1e0f3c5b8a6f",
        "document_type": "COVER_LETTER",
        "file_path": "b1e2c3d4-6f8e-4a9b-c0d2-1e0f3c5b8a6f",
        "original_filename": "Acme_CoverLetter.docx"
      }
    ]
  }
}
```

Error Responses

HTTP Status Code	Description	Response Body (JSON Example)
401 Unauthorized	Missing or invalid authentication token.	{"status": "error", "message": "Authentication required."}
403 Forbidden	The user is authenticated but does not own this	{"status": "error", "message": "Access

	application record.	denied. Application not found for this user."}
404 Not Found	The specified application_id does not exist for the authenticated user.	{"status": "error", "message": "Application 4e83... not found."}
500 Internal Server Error	A general server or database error occurred.	{"status": "error", "message": "Database error retrieving application details."}

Example cURL Command

```
# Example cURL command to retrieve the application with the specific UUID
APPLICATION_ID="4e83460b-39f0-4c00-a1f4-ac41c4994cb4"

curl -X GET http://localhost:8000/api/application/${APPLICATION_ID} \
    -H "Authorization: Bearer YOUR_AUTH_TOKEN_HERE"
```

API 22.0: GET /api/applications/all (Retrieve All User Applications)

Purpose

Retrieves a complete, aggregated list of all job applications belonging to the authenticated user (user_id). This endpoint efficiently combines data from the applications, job_titles, and companies tables to provide all necessary display information in a single response.

Endpoint Specification

Detail	**Specification**
Method	GET
URL	/api/applications/all

Authentication	Mandatory (Authorization: Bearer [TOKEN])
Purpose	Get all application records for the authenticated user, sorted by date_applied descending.
Required Parameters	None (User is identified by the authentication token)

Database Query Logic

The implementation will use a multi-JOIN query to fetch the application details, job title, and company name:

```
SELECT
    a.application_id,
    a.date_applied,
    a.current_status,
    jt.title_name,
    c.company_name_clean,
    c.company_id
FROM applications a
LEFT JOIN job_titles jt ON a.job_title_id = jt.job_title_id
LEFT JOIN companies c ON a.company_id = c.company_id
WHERE a.user_id = %s
ORDER BY a.date_applied DESC;
```

Success Response (HTTP 200 OK)

A successful request returns a JSON object containing an array of application records.

Field	**Type**	**Description**
status	string	"success"
applications	array	A list of aggregated application objects.

Application Object Fields

Field	Type	Description
application_id	UUID	The unique identifier for the application.
date_applied	date	The date the application was submitted (YYYY-MM-DD).
current_status	string	The current status of the application (e.g., 'SUBMITTED', 'INTERVIEW', 'OFFER').
title_name	string	The name of the job title (e.g., "Software Engineer II").
company_name_clean	string	The standardized company name (e.g., "Google").
company_id	integer	The ID of the standardized company profile. (Can be null if not standardized).

Example Success Response Body

```
{
 "status": "success",
 "applications": [
   {
     "application_id": "84d7a8d5-1b4d-4e9c-b8f1-8c4d2e8b23a1",
     "date_applied": "2024-06-10",
     "current_status": "INTERVIEW",
     "title_name": "Lead Frontend Architect",
     "company_name_clean": "Global Tech Innovations",
     "company_id": 451
   },
   {
     "application_id": "77f3e1b0-2c7a-4f5d-9b0d-1e0f3c5b8a6f",
     "date_applied": "2024-05-15",
     "current_status": "SUBMITTED",
     "title_name": "Data Scientist",
     "company_name_clean": "DataPro Analytics",
     "company_id": 102
   }
 ]
}
```

Error Responses

HTTP Status Code	Description	Response Body (JSON Example)
401 Unauthorized	Missing or invalid authentication token.	{"status": "error", "message": "Authentication required."}
500 Internal Server Error	A general server or database error occurred.	{"status": "error", "message": "Database error retrieving applications: N/A"}

Example cURL Command

```
# Example cURL command to retrieve all applications for the authenticated user

curl -X GET http://localhost:8000/api/applications/all \
    -H "Authorization: Bearer YOUR_AUTH_TOKEN_HERE"
```

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 **8.0** (GET, PUT, DELETE for single application UUID).
1.2.2	**2025-10-30**	**Documentation Fix:** Updated Request Body Fields table for Endpoint **9.0** to correctly list **document** and **document_type_code** for 100% contract compliance.
1.2.7	**2025-11-06**	**Add new endpoint 14**
1.2.8	**2025-11-06**	**Rebuilt endpoint 5 and documented.**
1.2.9	**2025-11-06**	**Enriched documentation for Endpoint 10**
1.2.10	**2025-11-06**	**Enriched Documentation for Endpoint 7, 8, 9**
1.2.11	**2025-11-06**	**Added Documentation for endpoint 15**

1.2.12	2025-11-17	Added Documentation for Endpoint 16
1.2.13	2025-11-18	Added Documentation for Endpoint 17,18
1.2.14	2025-11-18	Added Documentation for Endpoint 19,20
1.2.15	2025-11-25	Corrected documentation for endpoint 13 to include linkedin_url. Added Endpoint 21,22