JobAssist: Backend API Documentation

This document describes the current Flask-based RESTful API, implemented in app.py. It manages all data and documents for job applications using a PostgreSQL database (contact_db). All endpoints are now numbered sequentially from 1.0 to 12.0 for easier maintenance.

1. Configuration & Environment

Parameter	Value (Example/Default)	Notes
dbname	contact_db	The name of the database.
user	jobert	The PostgreSQL user.
password	linkedin	**SECURITY NOTE:** This password should be changed.
host	localhost	The database server host.

2. Company Management Endpoints (1.0 - 3.0)

These endpoints manage the standardized company profiles in the companies table.

1.0 GET /api/companies

Retrieves all standardized company profiles. Used as the primary data source for the sortable dashboard view (index.html).

Detail	Specification
Method	GET
URL	/api/companies
Returns	JSON array of all company profiles.

2.0 GET /api/next_company (Get Next Raw Name to Standardize)

Retrieves the next unstandardized company name that needs action. This is the main endpoint used by the data standardization front-end.

Detail	Specification
Method	GET
URL	/api/next_company
Authentication	Mandatory (Assumed to be protected)
Returns	JSON object with raw_name_id and raw_name, or a message if standardization is complete.

3.0 GET /api/companies/int:company_id (Get Single Company Profile)

Retrieves a single standardized company profile by its integer ID, including all core metadata.

Detail	Specification
Method	GET
URL	/api/companies/<int:company_id>
Authentication	Mandatory (User ID required)
Returns	JSON object containing the company profile details.

4.0 PUT /api/companies/int:company_id (Update Company Profile)

Updates a specific company profile's metadata (e.g., target_interest, size_employees). The request allows for partial updates.

Detail	Specification
Method	PUT
URL	/api/companies/<int:company_id>

Authentication	Mandatory (User ID required)
Request Body	JSON (Partial update, e.g., {"target_interest": "HIGH"})
Notes	URL uses integer company_id and plural form for consistency.

5.0 GET /api/companies/int:company_id/raw_names (Get Mapped Raw Names)

Retrieves all raw, unstandardized company names that are currently mapped to the given clean company_id. This is used to display the raw inputs that created the clean profile.

Detail	**Specification**
Method	GET
URL	/api/companies/<int:company_id>/raw_names
Authentication	Mandatory (User ID required)
Returns	JSON array of raw name objects (mapping_id, raw_name, source_filename, etc.).

6.0 POST /api/map/existing

Maps the specified unstandardized raw name (raw_name_id) to an **existing** clean company profile (company_id).

LOGIC: This action performs the mapping by strictly updating the company_name_mapping table where raw_name_id matches and the mapping is currently unassigned (company_id IS NULL). **The underlying SQL update logic does not filter by user_id**.

Detail	**Specification**
Method	POST
URL	/api/map/existing

Request Body	JSON: {"raw_name_id": 456, "company_id": 12}
Returns	Success message (200). Returns 404 if the raw name is not found or is already mapped.

7.0 POST /api/map/new

Creates a **new** clean company profile and maps the specified unstandardized raw name (raw_name_id) to it.

LOGIC: This endpoint executes a database transaction that first **inserts** a new company record into the companies table, retrieves the new company_id, and then **updates** the company_name_mapping table with the new ID where company_id IS NULL.

Detail	**Specification**
Method	POST
URL	/api/map/new
Request Body	JSON: {"raw_name_id": 123, "company_name_clean": "Alphabet Inc."}
Returns	Success message (201) and the new company_id. Returns 404 if the raw name is already mapped.

8.0 POST /api/map/self

Creates a **new** clean company profile using the raw name as the clean name and maps the specified unstandardized raw name (raw_name_id) to itself. This is a transactional endpoint.

Detail	**Specification**
Method	POST
URL	/api/map/self
Request Body	JSON: {"raw_name_id": 789, "raw_name":

	"Acme Corporation"}
Returns	Success message (201) and the new company_id. Returns 404 if the raw name is already mapped.

4. Application Management Endpoints (5.0 - 8.0)

These are the core endpoints for creating and managing job application records.

5. Document Management Endpoints (9.0 - 12.0)

These endpoints handle the secure upload, storage, retrieval, and management of user-uploaded files (resumes, cover letters, etc.).

9.0 POST /api/application/<id>/documents

Uploads a document (Resume, Cover Letter, etc.) associated with a specific job application. The file is saved to the disk storage (/filestore), and metadata is recorded in the job_documents table.

Detail	**Specification**
Method	POST
URL	/api/application/<uuid:application_id>/documents
URL Parameter	The **<id>** in the URL path is the

	Application ID (UUID) for the job application this document belongs to.
Authentication	Mandatory (Verifies user_id ownership of the application)
Request Type	multipart/form-data
Returns	JSON object with the new document_id UUID.

Request Body (multipart/form-data fields)

Field Name	Type	Required	Notes
document	File	Yes	The file binary data (e.g., .docx, .pdf).
document_type_code	string	Yes	The type of document. **Must be one of the enumerated values listed below.**

document_type_code Enumerated Values

The document_type_code value must match one of the following exact strings, which are directly mapped to the PostgreSQL document_type_enum (case-insensitive mapping is handled by the backend for robustness, but front-ends should use these official codes):

Code	Description	Database ENUM Value
RESUME	Candidate's main resume.	RESUME
COVER_LETTER	Candidate's cover letter.	COVER_LETTER
JOB_DESCRIPTION	Copy of the original job posting.	JOB_DESCRIPTION
ASSESSMENT_FORM	Any pre-interview	ASSESSMENT_FORM

	assessment or questionnaire.	
OTHER	Catch-all for miscellaneous documents.	OTHER

Example Request (curl)

```
curl -X POST
http://localhost:8000/api/application/4179f2ee-6082-47ae-bb8c-71a336eeaabb/documents \
  -F "document_type_code=RESUME" \
  -F "document=@/path/to/my_resume.pdf"
```

10.0 POST /api/applications (Inferred: Application Creation API)

Creates a new job application record. This requires looking up or creating the company_id and job_title_id if they don't exist.

Detail	**Specification**
Method	POST
URL	/api/applications
Authentication	Mandatory (Sets user_id ownership)
Request Body	JSON Payload:
	- company_name_clean: String (required).
	- title_name: String (required).
	- date_applied: Date string (required).
	- current_status: String (optional, defaults to a standard status).
Returns	JSON success message, including the new **application_id** (UUID) and the used

	company_id/job_title_id.

11.0 GET /api/applications?company_id=<int>

Retrieves all job applications, including nested job title and document information, for a specific company, filtered by the required company_id query parameter. This is the endpoint feeding the current application management view.

Detail	**Specification**
Method	GET
URL	/api/applications?company_id=<int>
Authentication	Mandatory (Verifies user_id ownership)
Returns (CORRECTED)	**Structured JSON object** containing the root keys: status and applications.

Formal Response Schema

The response is deliberately wrapped in an object for consistent client-side error handling and data extraction.

Field	**Type**	**Description**
status	String	Always "success" for a 200 response.
applications	Array	List of all matching application records.

Nested Application Object Structure

Field	**Type**	**Description**
application_id	UUID	Unique ID for the job application record.

date_applied	String	Date of application (YYYY-MM-DD format).
current_status	String	Current application status (e.g., "APPLIED", "INTERVIEW", "OFFER").
company_info	Object	Nested company details (ID, name).
job_title_info	Object	Nested job title details (ID, name).
documents	Array	**Crucial:** An array of all documents attached to this application.

Nested Document Object Schema (Inside documents array)

This structure is mandatory for the front-end to correctly construct download links and display names.

Field	**Type**	**Description**
document_id	UUID	Unique ID for the document record.
original_filename	String	**MANDATORY FIX:** The user-uploaded name of the file (used for link display and download name).
document_type	String	The internal type code (e.g., "RESUME", "COVER_LETTER").
file_path	String	The secure UUID filename on the disk (used internally by the backend, but available for debugging).

12.0 GET /api/documents/string:file_path

Downloads the physical file associated with a document record. The <string:file_path> parameter must be the **document_id** (the secure UUID) returned during the upload process (Endpoint 9.0).

Detail	Specification
Method	GET
URL	/api/documents/<string:file_path>
Authentication	Mandatory (Verifies file ownership against the user's applications)
Returns	The binary file stream, served with the original filename as the download_name.

NOT IMPLEMENTED

A.0 PUT /api/document/int:document_id (Inferred Standard CRUD)

Updates the metadata for a single document record, such as correcting the document_type or adding a description. Does not change the stored file.

Detail	Specification
Method	PUT
URL	/api/document/<int:document_id>
Authentication	Mandatory (Must own the document record)
Request Body	JSON (Partial update, e.g., { "document_type": "COVER_LETTER" })

B.0 GET /api/documents/string:file_path (Download Document)

Serves the stored document file securely. The file_path must be the secure, relative path returned during the upload process (Endpoint 9.0).

Detail	Specification
Method	GET
URL	/api/documents/<string:file_path>
Authentication	Mandatory (Verifies file ownership)
Returns	The binary file stream.

C.0 PUT /api/application/uuid:application_id (Inferred Standard CRUD)

Updates key fields on a specific application, such as changing the current_status (e.g., from "Applied" to "Interviewing") or updating the job_url.

Detail	Specification
Method	PUT
URL	/api/application/<uuid:application_id>
Authentication	Mandatory (Must own the application record)
Request Body	JSON (Partial update, e.g., { "current_status": "Interviewing" })

D.0 PUT /api/company/uuid:company_id

Updates a specific company profile's metadata (e.g., target_interest, size_employees).

Detail	Specification
Method	PUT
URL	/api/company/<uuid:company_id>
Authentication	Mandatory (User ID required)

Request Body	JSON (Partial update)

E.0 GET /api/job_titles (Inferred Standard CRUD)

Retrieves a list of all unique, standardized job titles currently stored in the system.

Detail	**Specification**
Method	GET
URL	/api/job_titles
Returns	JSON array of job title objects.

F.0 POST /api/application

Creates a new job application record. This endpoint automatically handles company and job title standardization (creating new records if they don't exist) before inserting the application.

Detail	**Specification**
Method	POST
URL	/api/application
Authentication	Mandatory (Assigns new record to current user)
Request Body	JSON (Requires company_name, job_title, date_applied, current_status)

H.0 GET /api/applications

Retrieves a summary of all job applications belonging to the authenticated user. This includes nested information about the associated company and job title.

Detail	**Specification**

Method	GET
URL	/api/applications
Authentication	Mandatory (Verifies user_id ownership)
Returns	JSON array of application objects, each containing company_info and job_title_info.

6. Version Control

Version	Date	Changes
1.0.0	2025-10-25	Initial documentation for Company Profiles (1.0, 2.0) and Application creation (5.0).
1.1.0	2025-10-27	Added Document Upload (9.0) and Download (10.0) endpoints. Implemented secure file serving.
1.2.0	**2025-10-27 (Latest)**	**MAJOR RE-INDEXING.** Added 6 inferred standard CRUD endpoints (3.0, 4.0, 7.0, 8.0, 11.0, 12.0) for completeness, bringing the total to **12**. Renumbered ALL documented endpoints sequentially from 1.0 to

		12.0.