

<!-- Filename: Backend\_API\_Documentation.md -->

<!-- Last Edited: 2025-10-24 -->

# JobAssist: Contact Standardization

## Backend API Documentation

This document describes the current Flask-based RESTful API implemented in app.py, which manages the standardization and retrieval of company data using a PostgreSQL database (contact\_db).

### Configuration & Environment

The backend relies on the following PostgreSQL connection parameters, defined in the DB\_CONFIG dictionary. These must be correctly configured for the application to function.

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 should be changed.
host	localhost	The database server host.

### 1. Data Retrieval Endpoints

These endpoints support the main **Dashboard (index.html)** view for displaying standardized data.

#### 1.1 GET /api/companies

Retrieves all standardized company profiles from the companies table. This is the primary data source for the sortable dashboard view.

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

<b>Method</b>	GET
<b>URL</b>	/api/companies
<b>Returns</b>	JSON array of all company profiles.

#### Success Response Body:

```
{
  "status": "success",
  "companies": [
    {
      "company_id": 1,
      "company_name_clean": "Google",
      "target_interest": true,
      "size_employees": 150000,
      "annual_revenue": 282.8,
      "revenue_scale": "B",
      "headquarters": "Mountain View, CA",
      "notes": "High priority target for Q4."
    },
    // ... more companies
  ]
}
```

## 2. Company Mapping and Cleanup Endpoints

These endpoints are used by the **Cleanup Interface (cleanup.html)** to standardize raw company names.

### 2.1 GET /api/unmapped\_list

Retrieves a list of all unique company names from the contacts table that do not yet have a corresponding entry in the company\_name\_mapping table.

Detail	Specification
<b>Method</b>	GET
<b>URL</b>	/api/unmapped_list
<b>Logic</b>	Performs a LEFT JOIN between contacts

	and company_name_mapping where the mapping is NULL.
--	---

#### Success Response Body:

```
{
  "status": "success",
  "raw_names": [
    "Google Inc.",
    "Googlle",
    "Acme Corp LTD",
    // ... more unmapped names
  ]
}
```

## 2.2 GET /api/suggestions

Searches for existing clean company names (companies.company\_name\_clean) using a case-insensitive partial match (ILIKE), used for the autocomplete feature.

Detail	Specification
Method	GET
URL	/api/suggestions
Query Param	q (string): The search query (e.g., ?q=goo).

#### Success Response Body:

```
{
  "suggestions": [
    {
      "company_id": 1,
      "company_name_clean": "Google"
    },
    {
      "company_id": 2,
      "company_name_clean": "Alphabet"
    }
  ]
}
```

}

## 2.3 POST /api/map/existing

Maps a single unmapped raw\_name to an existing standardized company profile identified by company\_id. Corresponds to **Cleanup Action 1**.

Detail	Specification
Method	POST
URL	/api/map/existing

### Request Body Schema:

```
{  
  "raw_name": "Google Inc.",  
  "company_id": 1  
}
```

## 2.4 POST /api/map/new (Create and Map)

Creates a new company profile in the companies table and immediately maps the given raw\_name to the newly generated company\_id. Used for **Cleanup Action 2**.

Detail	Specification
Method	POST
URL	/api/map/new

### Request Body Schema:

```
{  
  "raw_name": "NewCo. LLC",  
  "new_clean_name": "NewCo",  
  "target_interest": false // Optional, defaults to false  
}
```

## 2.5 POST /api/map/self (Skip and Self-Map)

Handles **Cleanup Action 3** (Skip & Self-Map). This is a convenience endpoint that reuses the logic of /api/map/new, where:

1. The raw\_name is used as the clean\_name.
2. target\_interest is set to TRUE by default.

Detail	Specification
Method	POST
URL	/api/map/self

**Request Body Schema (requires only raw\_name):**

```
{
  "raw_name": "Acme Corp"
}
```

## 2.6 POST /api/map/batch

Batch processes all remaining unmapped names. Each unmapped raw\_name is used to create a new company profile (self-map logic), and **all new profiles are marked as target\_interest: TRUE**.

Detail	Specification
Method	POST
URL	/api/map/batch

**Request Body:** An empty JSON object {}.

**Success Response Body:**

```
{
  "status": "success",
  "message": "Batch mapping complete.",
  "processed_count": 42 // Number of names processed
}
```

## 3. Management Placeholders

These endpoints are designed to support the **Management Interface** ([management.html](#)). In the current `app.py` implementation, these are placeholders and return a 501 Not Implemented status.

### 3.1 GET `/api/companies/{company_id}`

Retrieves the full detail of a single standardized company profile.

### 3.2 PUT `/api/companies/{company_id}`

Updates all editable fields for an existing standardized company profile.

## 3. Company Management Endpoints

These endpoints support the **Company Management Tool** ([management.html](#)) for viewing, retrieving, and updating the standardized company profiles and their related contact/raw name data.

### 3.1 GET `/api/companies/{company_id}` (Retrieve Single Company Profile & Related Data)

Retrieves a single standardized company profile, along with all associated raw company names and contact records.

Detail	Specification
Method	GET
URL	<code>/api/companies/&lt;int:company_id&gt;</code>
Returns	JSON object containing the company profile, a list of associated raw names, and a list of contact records.

#### Success Response Body:

```
{  
  "status": "success",
```

```
"company": {
  "company_id": 101,
  "company_name_clean": "Innovatech Solutions",
  "target_interest": true,
  "headquarters": "San Jose, CA",
  "size_employees": 500,
  "annual_revenue": 8.5,
  "revenue_scale": "M",
  "notes": "Key account for Q3 marketing efforts."
},
"raw_names": [
  "Innovatech Solutions Inc.",
  "InnovatechSol",
  "Innovatech-Solutions"
],
"contacts": [
  {
    "contact_id": 501,
    "first_name": "Jane",
    "last_name": "Doe",
    "job_title": "CTO",
    "email": "jane.doe@innovatech.com",
    "linkedin_url": "[https://linkedin.com/in/janedoe](https://linkedin.com/in/janedoe)"
  }
]
```

```
    // ... more contacts
  ]
}
```

### 3.2 PUT /api/companies/{company\_id} (Update Company Profile)

Updates the enrichment data and/or the clean name for a specific standardized company profile.

Detail	Specification
Method	PUT
URL	/api/companies/<int:company_id>
Returns	JSON object confirming success and returning the updated profile data.

#### Request Body Schema:

Only send the fields that need to be updated. All fields are optional, but at least one must be provided.

Field Name	Data Type	Required	Description
company_name_clean	string(100)	No	The standardized company name.
target_interest	boolean	No	Flag for sales/marketing priority.
headquarters	string(100)	No	Primary location.
size_employees	integer	No	Estimated number of employees.



<code>annual_revenue</code>	<code>numeric</code>	No	Numeric part of the annual revenue.
<code>revenue_scale</code>	<code>string(10)</code>	No	Scale of the revenue (K, M, or B).
<code>notes</code>	<code>text</code>	No	Internal notes/history.

#### Example Request Body:

```
{
  "headquarters": "London, UK",
  "target_interest": true
}
```

#### Success Response Body:

```
{
  "status": "success",
  "message": "Company profile updated successfully.",
  "updated_data": {
    "company_id": 101,
    "company_name_clean": "Innovatech Solutions",
    "target_interest": true,
    "headquarters": "London, UK",
    "size_employees": 500,
    "annual_revenue": 8.5,
    "revenue_scale": "M",
  }
}
```

"notes": "Key account for Q3 marketing efforts."

}

}