API Contracts: Okada Leasing Agent

This document outlines the API contracts for the Okada Leasing Agent backend service. All endpoints are prefixed with `/api`.

---

1. Chat Endpoint

- **POST** `/api/chat`
- **Description**: This is the primary endpoint for interacting with the chatbot. It takes the user's message and conversation history and returns the AI's response.
- **Content-Type**: `application/json`

Request Body

```
{
  "user_id": "user@example.com",
  "message": "What is the largest property you have?",
  "history": [
    {
      "role": "user",
      "content": "Hi there."
    },
    {
      "role": "assistant",
      "content": "Hello! How can I help you today?"
    }
  ]
}
```

- `user_id` (string, required): The email address of the current user.
- `message` (string, required): The latest message from the user.
- `history` (array of objects, required): The preceding conversation turns.

Response Schema

On Success (200 OK):

```
{
  "answer": "The largest property is 53 Pearl St with a size of 15,289 SF.",
  "schedule_details": null
}
```

- `answer` (string): The chatbot's text response.
- `schedule_details` (object or null): If the bot detects a scheduling intent with enough details, this object will be populated. Otherwise, it is `null`.

If scheduling details are present:

```
{
  "answer": "I can help with that. Please confirm the details below and I'll get it on the calendar.",
  "schedule_details": {
    "address": "53 Pearl St",
    "time": "2023-10-28T14:00:00"
  }
}
```

---

2. User Management Endpoints

Create or Update User

- **POST** `/api/user`
- **Description**: Creates a new user or updates an existing user's details (e.g., adds a company name).
- **Content-Type**: `application/json`

Request Body

```
{
  "full_name": "John Doe",
  "email": "john.doe@example.com",
  "company_name": "Example Corp"
}
```

Response (200 OK)

```
{
    "id": "653d6f7e8f8f8f8f8f8f8f8f",
    "full_name": "John Doe",
    "email": "john.doe@example.com",
    "company": {
        "id": "653d6f7e8f8f8f8f8f8f8f90",
        "name": "Example Corp"
    },
    "preferences": {},
    "scheduled_events": []
}
```

Get User by Email

- **GET** `/api/user?email={email}`
- **Description**: Retrieves a single user's profile by their email address.

Response (200 OK)

Returns the same user object schema as the POST request above.

---

3. Document Management Endpoints

Upload Document

- **POST** `/api/documents/upload`
- **Description**: Uploads a CSV file for a specific user and triggers a background task to index the data.
- **Content-Type**: `multipart/form-data`

Request Body

- A form containing two parts:
    - `user_id`: The email address of the user.
    - `file`: The `.csv` file being uploaded.

Response (200 OK)

```
{
  "message": "File 'your_file.csv' uploaded successfully for user user@example.com. Re-indexing all documents."
}
```

Load User Documents

- **POST** `/api/documents/load`
- **Description**: Called on user login. Checks for existing documents for the user and starts a background indexing task if they are found.
- **Content-Type**: `application/json`

Request Body

```
{
  "user_id": "user@example.com"
}
```

Response (200 OK)

```
{
  "message": "Checking for existing documents. Indexing will start if documents are found."
}
```

Get Indexing Status

- **GET** `/api/documents/status`
- **Description**: Returns the current status of any active document indexing job. The frontend polls this endpoint.

Response (200 OK)

```
{
  "status": "success",
  "message": "Documents for user@example.com loaded successfully."
}
```

- `status` can be `idle`, `in_progress`, `success`, or `error`.

---

4. Scheduling Endpoint

- **POST** `/api/schedule`
- **Description**: Creates a Google Calendar event after the user has confirmed the details in the chat.
- **Content-Type**: `application/json`

Request Body

```
{
  "email": "user@example.com",
  "address": "53 Pearl St",
  "time": "2023-10-28T14:00:00"
}
```

Response (200 OK)

```
{
  "message": "Event scheduled successfully and linked to user.",
  "event_url": "https://calendar.google.com/event?eid=..."
}
```

---

5. Data Reset Endpoint

- **POST** `/api/reset` or `/api/reset?user_id={email}`
- **Description**: A utility endpoint for development. Deletes data. If a `user_id` is provided, it resets only that user's data. If not, it wipes all data in the application (users, companies, history, RAG index).

Response (200 OK)

```
{
  "message": "All application data has been reset."
}
```