Title: Email Sender Application Documentation

---

1. Introduction

The **Email Sender Application** is a Django-based automation tool designed to facilitate mass emailing with the flexibility of scheduling, personalization, and monitoring. Leveraging **Celery** for asynchronous task handling, **Redis** for caching, and **WebSockets** for real-time updates, this application provides a complete solution for email campaigns.

---

2. Setup and Configuration Instructions

2.1 Prerequisites

- **Python 3.6+**: Ensure Python is installed on your system. Check with python --version.
- **Redis Server**: Required for task brokering with Celery. Install and start Redis (redis-server) as described below.
- **Virtual Environment**: A virtual environment is recommended to manage dependencies.

2.2 Installation

1. **Clone the Project**
    - Clone the repository to your local machine.
    - Navigate to the project directory.
2. **Create a Virtual Environment**
    - Run the following commands to create and activate a virtual environment:

bash

python3 -m venv venv

source venv/bin/activate

3. **Install Dependencies**
    - With the virtual environment activated, install project dependencies:

bash

pip install -r requirements.txt

    - These dependencies include Django, Celery, Redis, and other required packages.

4. **Start Redis Server**
    - Make sure Redis is installed on your system.
    - Start the Redis server:

bash

redis-server

    - **Note**: Redis is essential for Celery to handle background tasks like sending emails.
5. **Apply Migrations**
    - Set up the Django database by applying migrations:

bash

python manage.py migrate

    - This will create the necessary tables in the SQLite database (default).
6. **Create Superuser (Optional)**
    - Optionally, create an admin account to access the Django admin interface:

bash

python manage.py createsuperuser

7. **Run the Django Development Server**
    - Start the Django development server:

bash

python manage.py runserver

    - Access the application at http://localhost:8000/.

2.3 API Key Configuration

1. **Email Service Provider (ESP) Setup**
    - The application uses **SendGrid** for email delivery.
    - Sign up on the SendGrid website if you haven't already.
    - Generate an API key from the SendGrid dashboard:
        - Go to **Settings > API Keys > Create API Key**.
        - Name your API key and assign the necessary permissions (e.g., full access for sending emails).

        - Copy the API key to use in the following steps.

2. **OpenAI API Setup**
    - The application uses **OpenAI** for generating personalized content within emails.
    - Create an account on OpenAI and navigate to the API section.
    - Generate an API key from the **API Keys** section.
3. **Update API Keys in Django Settings**
    - Open the settings.py file and locate the configuration section for API keys.
    - Replace placeholders with your actual API keys:

python

```
EMAIL_HOST_USER = 'apikey'  # Username for SendGrid

EMAIL_HOST_PASSWORD = 'your_sendgrid_api_key'

OPENAI_API_KEY = 'your_openai_api_key'
```

    - **Note**: These credentials are sensitive information. Avoid hardcoding them directly in production by using environment variables.

---

3. Configuring Email Scheduling and Throttling

The Email Sender application allows for flexible scheduling options and throttling control to optimize email delivery and avoid server overload.

3.1 Email Scheduling Options

1. **Immediate Sending**:
    - Choose this option to send emails to all recipients right away.
    - Go to the **Schedule Emails** page and select the **Send Immediately** option.
    - Emails will be queued and sent without delay.
2. **Scheduled Time**:
    - Schedule emails for a specific date and time.
    - Select the **Schedule for Specific Time** option.
    - Enter the desired date and time in the provided field in the format YYYY-MM-DD HH:MM.
    - This option is useful for timing email campaigns to reach users at optimal times.

3. **Staggered Sending**:
    - Sends emails at staggered intervals to distribute load and avoid server strain.
    - Choose **Staggered Sending** and enter the interval (in seconds) for sending each email.
    - Example: If set to 10 seconds, each email will be sent 10 seconds apart.
    - This is helpful for large campaigns where ESPs limit the number of emails per minute.

3.2 Throttling Setup

- **Throttle Rate**:
    - Controls the number of emails sent per minute.
    - Specify the throttle rate on the **Schedule Emails** page. For instance, setting 5 will send 5 emails per minute.
    - Helps prevent exceeding ESP rate limits and maintains consistent delivery.
    - Adjust based on your ESP's limit (e.g., 100 emails per minute for some plans).

---

4. Usage Instructions

4.1 Upload Contacts

1. **Navigate to the Upload Page**:
    - Go to http://localhost:8000/ and select the **Upload CSV** option from the navigation bar.
2. **Upload a CSV File**:
    - Choose a CSV file containing contact details. Required headers include:
        - Email: The recipient's email address.
        - Name, Company Name, Location, Products (optional): Additional fields for personalization.
    - Click **Upload** to import contacts. A confirmation message will display once contacts are successfully uploaded.

4.2 Create Email Template

1. **Navigate to the Create Template Page**:
    - Go to **Create Template** in the navigation bar.
2. **Enter Subject and Body**:
    - In the subject and body fields, use placeholders for dynamic data, like:
        - {{ Name }}, {{ Location }}, {{ Company Name }}, etc.

- Placeholders pull data from the uploaded CSV file and enable personalization for each recipient.

3. **Save Template**:
    - Once the template is complete, click **Save Template**. This will store the template for future use.

4.3 Schedule Emails

1. **Navigate to the Schedule Emails Page**:
    - Choose **Schedule Emails** from the navigation bar.
2. **Select Scheduling Option**:
    - Choose from **Immediate**, **Specific Time**, or **Staggered** sending.
    - Specify throttle rate (emails per minute) as needed.
3. **Schedule Emails**:
    - Click **Schedule Emails** to begin the process. Emails will be sent based on the selected scheduling option.

4.4 Monitor Email Status

1. **Navigate to the Email Status Page**:
    - Go to **Email Status** to monitor email delivery.
2. **View Real-Time Status**:
    - The page displays the delivery status, open rates, and errors.
    - WebSocket integration provides real-time updates.

4.5 Dashboard Analytics

1. **Access the Dashboard**:
    - Go to the **Dashboard** page for an overview of email performance metrics.
2. **View Analytics**:
    - Key metrics include:
        - Total emails sent
        - Pending emails
        - Failed emails
        - Opened emails
    - This helps assess engagement and campaign performance.

---

5. Troubleshooting

- **Redis Server Not Starting**: Ensure Redis is installed correctly. Run redis-server to start Redis.
- **Email Not Sending**: Check your SendGrid API key and confirm ESP rate limits aren't exceeded.
- **Error Messages**: Use the Django admin and logs to debug any issues with user accounts or server configurations.