

Email Marketing System API Documentation

Overview

This API provides a comprehensive email marketing solution with features for template management, contact handling, email scheduling, and bulk email sending. The system supports HTML templates, PDF to HTML conversion, CSV contact imports, and scheduled email campaigns.

Base Configuration

- **Base URL:** `http://localhost:5000`
- **Database:** MongoDB (Atlas cluster)
- **Email Service:** Gmail SMTP
- **File Storage:** Memory-based with temporary disk storage for CSV processing

Authentication

Currently, the API does not implement authentication. All endpoints are publicly accessible.

Global Middleware

- **CORS:** Enabled for all origins
- **JSON Parser:** Express JSON middleware for request body parsing
- **File Upload:** Multer for handling file uploads

Endpoints

1. Health Check

GET `/api/health`

Check if the API service is running.

Response:

```
json
{
  "status": "OK",
  "message": "PDF to HTML converter, HTML email sender, and email extractor service is running"
}
```

2. Contact Management

GET `/api/getIndustries`

Retrieve all contact groups organized by industry/file name.

Response:

```
json
{
  "contacts": [
    {
      "_id": "filename.csv",
      "count": 150,
      "documents": [
        {
          "_id": "contact_id",
          "email": "user@example.com",
          "industry": "Technology",
          "fileName": "filename.csv"
        }
      ]
    }
  ]
}
```

POST `/api/extract-emails`

Upload and process a CSV file to extract email contacts.

Request:

- **Content-Type:** `multipart/form-data`
- **File Field:** `contacts` (CSV file)

CSV Requirements:

- Must contain an "email" column (case insensitive)
- Must contain an "industry" column (case insensitive)
- Maximum file size: 10MB

Response:

```
json
{
  "success": true,
  "emails": [
    {
      "email": "user@example.com",
      "industry": "Technology",
      "fileName": "contacts.csv"
    }
  ],
  "count": 150,
  "totalProcessed": 155,
  "duplicatesSkipped": 5,
  "originalFilename": "contacts.csv",
  "validation": {
    "hasEmailColumn": true,
    "hasIndustryColumn": true,
    "rowsProcessed": 155
  }
}
```

Error Responses:

- `400`: Missing required columns, invalid file format, or no valid emails found
- `500`: Server error during processing

GET `/api/contacts/count`

Get the total number of contacts in the database.

Response:

```
json
{
  "count": 1250
}
```

3. Template Management

GET `/api/get-htmls`

Retrieve all stored HTML templates.

Response:

```
json
{
  "htmls": [
    {
      "_id": "template_id",
      "fileName": "newsletter_template.html"
    }
  ]
}
```

POST `/api/convert-template`

Convert a PDF file to HTML and send emails to all contacts.

Request:

- **Content-Type:** `multipart/form-data`
- **File Field:** `template` (PDF file)
- **File Requirements:** PDF format, maximum 25MB

Response:

```
json
{
  "success": true,
  "message": "Emails sent successfully to 500 contacts",
  "sampleHtml": "<!DOCTYPE html> <html>..."
}
```

Error Responses:

- `400`: No file uploaded, invalid file format
- `500`: Conversion or email sending failure

4. Email Sending

POST `/api/send-html-template`

Send HTML emails immediately to contacts in a specific industry.

Request:

- **Content-Type:** `multipart/form-data`

Form Fields:

- `subject` (string, required): Email subject line
- `industry` (string, required): Target industry/contact group
- `templateOption` (string, required): "new" or "existing"
- `selectedTemplate` (string, required if `templateOption` is "existing"): Template filename
- `htmlTemplate` (file, required if `templateOption` is "new"): HTML file upload

Response:

```
json
{
  "success": true,
  "message": "Sent 147 emails successfully",
  "stats": {
    "total": 150,
    "success": 147,
    "failed": 3,
    "totalContacts": 150,
    "successCount": 147,
    "failedEmails": ["invalid@email.com", "bounced@example.com", "blocked@domain.com"]
  },
  "preview": "<!DOCTYPE html><html><head>..."
}
```

Error Responses:

- `400`: Missing required fields, no contacts found, or template not found
- `500`: Email sending failure

5. Email Scheduling

POST `/api/send-html-scheduler-template`

Schedule HTML emails to be sent at a specific date and time.

Request:

- **Content-Type:** `multipart/form-data`

Form Fields:

- `subject` (string, required): Email subject line
- `industry` (string, required): Target industry/contact group
- `scheduledDate` (string, required): Date in YYYY-MM-DD format
- `scheduledTime` (string, required): Time in HH:MM format (24-hour)
- `templateOption` (string, required): "new" or "existing"
- `selectedTemplate` (string, required if templateOption is "existing"): Template filename
- `htmlTemplate` (file, required if templateOption is "new"): HTML file upload

Response:

```
json
{
  "success": true,
  "message": "Email ready for schedule",
  "totalContacts": 150,
  "successCount": 150,
  "failedCount": 0,
  "sampleHtml": "<!DOCTYPE html> <html>..."
}
```

6. Email Tracking

GET `/getEmailSentInfo`

Retrieve information about sent emails.

Response:

```
json
```

```
{
  "emailSent": [
    {
      "_id": "sent_email_id",
      "contact": {
        "_id": "contact_id",
        "email": "user@example.com",
        "industry": "Technology"
      },
      "createdAt": "2025-08-22T10:30:00.000Z"
    }
  ],
  "lastEmailSent": {
    "_id": "last_sent_id",
    "contact": {
      "_id": "contact_id",
      "email": "recent@example.com",
      "industry": "Finance"
    },
    "createdAt": "2025-08-22T15:45:00.000Z"
  }
}
```

Data Models

Contact Model

```
javascript

{
  email: String (required),
  industry: String,
  fileName: String
}
```

HTML Content Model

```
javascript
```

```
{  
  htmlContent: Buffer (compressed),  
  fileName: String  
}
```

Schedule Model

```
javascript  
  
{  
  email: String,  
  industry: String,  
  subject: String,  
  date: Date,  
  time: String,  
  htmlcontent: Buffer,  
  status: String (default: 'pending')  
}
```

Email Sent Model

```
javascript  
  
{  
  contact: ObjectId (ref: 'Contact'),  
  createdAt: Date (default: Date.now)  
}
```

Email Configuration

The system uses Gmail SMTP with the following configuration:

- **Service:** Gmail
 - **Username:** leads@enrichifydata.com
 - **Password:** Application-specific password
 - **Pool:** Connection pooling enabled
 - **Rate Limit:** 5 messages per second
 - **Max Connections:** 5 concurrent connections
-

Scheduled Email Processing

The system includes an automated scheduler that runs every minute to process pending emails:

- **Schedule:** Every minute (* * * * *)
 - **Function:** Checks for emails scheduled for the current time or earlier
 - **Processing:** Decompresses HTML content and sends emails
 - **Status Updates:** Updates email status to 'sent' or 'failed'
 - **Notifications:** Sends success/failure notifications to the admin email
-

File Upload Specifications

PDF Files (Template Conversion)

- **Maximum Size:** 25MB
- **Allowed Types:** application/pdf
- **Storage:** Memory-based

HTML Files (Templates)

- **Maximum Size:** 10MB
- **Allowed Types:** text/html, .html files
- **Storage:** Memory-based, compressed for database storage

CSV Files (Contacts)

- **Maximum Size:** 10MB
 - **Allowed Types:** text/csv, .csv files
 - **Storage:** Temporary disk storage in /tmp directory
 - **Auto-cleanup:** Files are automatically deleted after processing
-

Error Handling

The API implements comprehensive error handling:

- **400 Bad Request:** Invalid input, missing required fields, or malformed data
- **500 Internal Server Error:** Database errors, email sending failures, or processing errors

All error responses include descriptive error messages to help with debugging.

HTML Template Processing

The system includes advanced HTML template processing for email compatibility:

Features:

- **Email Client Compatibility:** Adds CSS fixes for Outlook, Gmail, and other clients
- **CSS Inlining:** Converts CSS styles to inline styles using Juice
- **Image Optimization:** Adds proper attributes for email images
- **Table Fixes:** Ensures proper table rendering in email clients
- **Script Removal:** Removes JavaScript (not supported in emails)
- **Compression:** Gzip compression for template storage

Template Enhancements:

- DOCTYPE declaration for XHTML compatibility
 - CSS fixes for various email clients
 - Dark mode support
 - Mobile responsiveness
 - Cross-client font rendering
-

Rate Limiting and Performance

- **Email Sending:** 200ms delay between emails to prevent rate limiting
 - **Connection Pooling:** Up to 5 concurrent SMTP connections
 - **Bulk Processing:** Efficient batch processing for large contact lists
 - **Memory Management:** Temporary file cleanup and memory-based storage
-

Notifications

The system sends automated notifications for:

- **Successful email delivery:** Sent to admin email with progress tracking
- **Failed email delivery:** Detailed error notifications with failure reasons

- **Batch completion:** Summary statistics for bulk email campaigns
-

Security Considerations

- **File Type Validation:** Strict file type checking for uploads
 - **Size Limits:** File size restrictions to prevent abuse
 - **Email Validation:** Regex-based email address validation
 - **SQL Injection Prevention:** MongoDB queries use proper escaping
 - **CORS Configuration:** Currently allows all origins (should be restricted in production)
-

Dependencies

Core Dependencies:

- **express:** Web framework
- **mongoose:** MongoDB ODM
- **multer:** File upload handling
- **nodemailer:** Email sending
- **csv-parser:** CSV file processing
- **node-cron:** Scheduled job processing

Email Processing:

- **juice:** CSS inlining for email templates
- **zlib:** Template compression/decompression

External Services:

- **ConvertAPI:** PDF to HTML conversion service
 - **MongoDB Atlas:** Cloud database
 - **Gmail SMTP:** Email delivery service
-

Usage Examples

Upload and Send Immediate Email

bash

```
curl -X POST http://localhost:5000/api/send-html-template \  
-F "subject=Monthly Newsletter" \  
-F "industry=Technology" \  
-F "templateOption=new" \  
-F "htmlTemplate=@newsletter.html"
```

Schedule Email Campaign

bash

```
curl -X POST http://localhost:5000/api/send-html-scheduler-template \  
-F "subject=Product Launch" \  
-F "industry=Marketing" \  
-F "scheduledDate=2025-08-25" \  
-F "scheduledTime=09:00" \  
-F "templateOption=existing" \  
-F "selectedTemplate=launch_template.html"
```

Import Contacts from CSV

bash

```
curl -X POST http://localhost:5000/api/extract-emails \  
-F "contacts=@contacts.csv"
```