

Calvin's Coffee Roast - Email Notification Microservice

A simple microservice to process and send coffee order confirmation emails and promotional announcements. Built with Express.js and Nodemailer, designed to be called by the main ordering system via HTTP POST.

⚠️ **Note:** This service is hosted on a free Render instance, which may enter a sleep state after a few minutes of inactivity. If the service is idle, users may experience a short delay while it spins back up. When ready, a request to the base URL will return: "Calvin's Microservice is Running"

Live Production Instance

Hosted with Render at:

- BASE URL: `https://calvins-coffee-roast.onrender.com`

Overview

This microservice receives HTTP POST requests at `/api/notify` containing an order ID or Member ID, then sends a confirmation email to the customer confirming their coffee purchase or a promotional announcement email. It is stateless and designed for quick synchronous processing.

Features

- Accepts POST requests with JSON payload containing `orders/orderId` and `/members/memberId`.
- Sends a hardcoded confirmation email for demonstration purposes.
- CORS enabled for public requests.
- Logs incoming requests with timestamps, origins, and IP addresses.
- Easy to extend with real email provider credentials.
- Returns JSON `{ "status": "sent" }` on success.
- Data validation returning `{ error: "invalid request" }` missing or non-numeric IDs.

API Overview

This service listens for POST requests and sends hardcoded email responses for testing/demo purposes.

Endpoints

Endpoint	Description
POST <code>/api/notify/orders/:orderId</code>	Sends a confirmation email for an order
POST <code>/api/notify/members/:memberId</code>	Sends a promotional email to a member

How to Request Data from Microservice A

User must send an HTTP `POST` request to one of the service's endpoints, passing a valid numeric ID as a **URL parameter**. I have provided different ways to test it. Through a static HTML page and through a raw http file.

Example: Sending Order Confirmation

```
curl -X POST https://calvins-coffee-roast.onrender.com/api/notify/orders/:orderId
```

Response:

```
{
  "status": "Confirmation Email Sent Successfully",
  "preview": "https://ethereal.email/message/EMAIL_PREVIEW_LINK"
}
```

Example: Sending Promotional Announcement

```
curl -X POST https://calvins-coffee-roast.onrender.com/api/notify/members/:memberId
```

Response:

```
{
  "status": "Promotional Announcement Sent Successfully",
  "preview": "https://ethereal.email/message/EMAIL_PREVIEW_LINK"
}
```

How to Receive Data from This Microservice

The microservice responds synchronously with a JSON payload in the body of the response. Clients should inspect the status and optionally the preview field, which provides a link to view the generated email via Ethereum Email.

General Response:

```
{
  "status": "----- Sent Successfully",
  "preview": "https://ethereal.email/message/EMAIL_PREVIEW_LINK"
}
```

Input Validation

- `orderId` and `memberId` must be numeric.
- Missing or non-numeric values result in HTTP `400` :

Example Error Response

```
{
  "error": "invalid orderId"
}
```

```
{
  "error": "invalid memberId"
}
```

UML Sequence Diagram

