

Email Management System Documentation

Overview

The Email Management System provides a comprehensive interface for administrators to send, manage, track, and resend emails within the Mindful Champion application. This system integrates with Resend for email delivery and maintains a complete history of all emails sent through the platform.

Features

Completed Implementation

1. Send Custom Emails

- Send emails to any recipient
- HTML and plain text support
- Template-based or custom content
- Real-time delivery tracking

2. Email Templates

- Pre-built templates for common use cases
- Sponsor confirmation emails
- Sponsor approval emails
- Admin notifications
- Custom email templates

3. Email History & Logs

- Complete history of all sent emails
- Filter by type, recipient, date range
- View email details and status
- Search functionality

4. Resend Functionality

- Resend any previous email with one click
- Maintains reference to original email
- Tracks resend history

5. Test Email System

- Send test emails to verify system functionality
- Formatted test templates
- Delivery verification

6. Statistics Dashboard

- Total emails sent
- Success/failure rates
- Pending emails
- Real-time metrics

Access

Admin Navigation

Navigate to **Admin Dashboard → Email Notifications** or directly access:

- URL: /admin/emails
- Requires: ADMIN role

API Endpoints

1. Send Custom Email

POST /api/admin/emails/send

Sends a custom email to a specified recipient.

Request Body:

```
{
  "to": "recipient@example.com",
  "subject": "Email Subject",
  "body": "Email body content (HTML or plain text)",
  "template": "custom",
  "isHtml": true
}
```

Response:

```
{
  "success": true,
  "message": "Email sent successfully",
  "emailLog": {
    "id": "email-log-id",
    "recipientEmail": "recipient@example.com",
    "status": "SENT",
    "sentAt": "2024-11-30T...",
    ...
  }
}
```

2. Resend Email

POST /api/admin/emails/resend

Resends a previously sent email.

Request Body:

```
{
  "emailLogId": "previous-email-log-id"
}
```

Response:

```
{
  "success": true,
  "message": "Email resent successfully",
  "emailLog": {
    "id": "new-email-log-id",
    "subject": "[RESEND] Original Subject",
    "status": "SENT",
    ...
  }
}
```

3. Get Email History

GET /api/admin/emails/history

Retrieves email history with optional filters.

Query Parameters:

- type : Filter by email type (optional)
- recipient : Filter by recipient email (optional)
- dateFrom : Start date for filtering (optional)
- dateTo : End date for filtering (optional)
- limit : Number of results per page (default: 50)
- offset : Pagination offset (default: 0)

Response:

```
{
  "emails": [
    {
      "id": "email-id",
      "recipientEmail": "recipient@example.com",
      "subject": "Email Subject",
      "status": "SENT",
      "sentAt": "2024-11-30T...",
      "user": {
        "email": "admin@example.com",
        "firstName": "Admin",
        "lastName": "User"
      }
    }
  ],
  "pagination": {
    "total": 100,
    "limit": 50,
    "offset": 0,
    "hasMore": true
  },
  "stats": {
    "total": 100,
    "sent": 95,
    "failed": 3,
    "pending": 2
  }
}
```

4. Send Test Email

POST /api/admin/emails/test

Sends a formatted test email to verify system functionality.

Request Body:

```
{
  "to": "test@example.com"
}
```

Response:

```
{
  "success": true,
  "message": "Test email sent successfully",
  "emailLog": {
    "id": "test-email-id",
    "type": "ADMIN_TEST",
    "status": "SENT",
    ...
  }
}
```

Email Templates

Available Templates

1. Custom Email

- Fully customizable subject and body
- Use for unique, one-off communications

2. Sponsor Confirmation

- Pre-formatted confirmation for sponsor applications
- Subject: “🎉 Sponsor Application Received - Mindful Champion”

3. Sponsor Approval

- Pre-formatted approval notification
- Subject: “✅ Sponsor Application Approved - Mindful Champion”

4. Admin Notification

- General administrative notifications
- Subject: “🔔 Admin Notification - Mindful Champion”

Using Templates

1. Navigate to the **Templates** tab
2. Click on a template card
3. The form will auto-populate with the template content
4. Customize the recipient and any content as needed
5. Send the email

Email Types (Database Enum)

The system supports the following email types:

- `VIDEO_ANALYSIS_COMPLETE` - Video analysis completion notifications
- `WELCOME` - Welcome emails for new users
- `SUBSCRIPTION_RENEWAL` - Subscription renewal reminders
- `ACHIEVEMENT_UNLOCKED` - Achievement notifications
- `MATCH_Reminder` - Match reminders
- `TRAINING_Reminder` - Training reminders
- `SYSTEM_UPDATE` - System update notifications
- `TRIAL_EXPIRATION` - Trial expiration warnings
- `TRIAL_WARNING_7_DAYS` - 7-day trial warning
- `TRIAL_WARNING_3_DAYS` - 3-day trial warning
- `TRIAL_WARNING_1_DAY` - 1-day trial warning
- `UPGRADE_Reminder` - Upgrade reminders
- `EVENT_Reminder` - Event reminders
- `PODCAST_NEW_EPISODE` - New podcast episode notifications
- `LIVE_STREAM_STARTING` - Live stream notifications
- `ADMIN_CUSTOM` - Custom admin emails
- `ADMIN_TEST` - Test emails
- `SPONSOR_APPLICATION` - Sponsor application notifications
- `SPONSOR_APPROVAL` - Sponsor approval notifications

Email Status Types

- `PENDING` - Email queued for sending
- `SENDING` - Email currently being sent
- `SENT` - Email successfully sent to email service
- `DELIVERED` - Email delivered to recipient's inbox
- `OPENED` - Recipient opened the email
- `CLICKED` - Recipient clicked a link in the email
- `FAILED` - Email failed to send
- `BOUNCED` - Email bounced back

Database Schema

EmailNotification Model

```

model EmailNotification {
    id          String          @id @default(cuid())
    userId      String
    videoAnalysisId String?
    type        EmailNotificationType @default(VIDEO_ANALYSIS_COMPLETE)
    recipientEmail String
    recipientName String?
    subject      String
    htmlContent  String         // HTML email content
    textContent  String?        // Plain text fallback
    status       EmailStatus     @default(PENDING)
    sentAt       DateTime?
    deliveredAt DateTime?
    openedAt    DateTime?
    clickedAt   DateTime?
    failedAt    DateTime?
    error        String?
    metadata     Json?          // Additional data (template, resend info, etc.
)
    resendEmailId String?
    retryCount   Int            @default(0)
    lastRetryAt DateTime?
    createdAt    DateTime
    updatedAt    DateTime
    user        User            @relation(fields: [userId], references: [id], o
nDelete: Cascade)
    videoAnalysis VideoAnalysis? @relation(fields: [videoAnalysisId],
references: [id], onDelete: Cascade)
}

```

Test Emails Sent

As part of the initial setup and testing, test emails were successfully sent to:

1. lee@onesoulpickleball.com ✓
2. Deansnow59@gmail.com ✓

Both emails included:

- Formatted HTML content
- Responsive design
- System status verification
- Feature highlights

User Interface

Dashboard Tabs

1. **Send Email Tab**
 - Email form with recipient, subject, and body fields
 - HTML/Plain text toggle
 - Send button with loading state

2. Templates Tab

- Grid of available email templates
- Preview and select templates
- One-click template application

3. History Tab

- Searchable and filterable email list
- Status badges for each email
- Resend button for each email
- Date range filters

4. Test Tab

- Send test emails to verify system
- Formatted test email template
- System verification checklist

Statistics Cards

Located at the top of the dashboard:

- **Total Emails** - All emails sent through the system
- **Sent** - Successfully sent emails
- **Failed** - Failed email attempts
- **Pending** - Emails waiting to be sent

Security

Authentication

- All endpoints require admin authentication
- Session validation via NextAuth
- Role-based access control (ADMIN role required)

Authorization

```
const session = await getServerSession(authOptions);
if (!session?.user?.email || session.user.role !== 'ADMIN') {
  return NextResponse.json({ error: 'Unauthorized' }, { status: 401 });
}
```

Email Service Integration

Resend Configuration

The system uses Resend for email delivery:

```
import { getResendClient } from '@/lib/email/resend-client';

const resend = getResendClient();
const fromEmail = 'Mindful Champion <noreply@mindfulchampion.com>';

await resend.emails.send({
  from: fromEmail,
  to: recipientEmail,
  subject: emailSubject,
  html: htmlContent,
  text: textContent,
});

});
```

Environment Variables

Required in `.env.local` :

```
RESEND_API_KEY=rexxxxxxxxxxxxxxxxxxxxx
ADMIN_EMAIL=admin@mindfulchampion.com
```

Future Enhancements

Potential improvements for the email system:

1. Bulk Email Sending

- Send to multiple recipients at once
- CSV import for bulk operations

2. Email Scheduling

- Schedule emails for future delivery
- Recurring email campaigns

3. Advanced Templates

- Visual template editor
- Template variables and personalization
- Template versioning

4. Analytics & Reporting

- Open rate tracking
- Click-through rate analysis
- Email performance reports
- A/B testing capabilities

5. Email Campaigns

- Multi-email campaign management
- Drip campaigns
- User segmentation

6. Attachment Support

- Upload and attach files to emails
- PDF generation and attachment

7. Email Verification

- Verify email addresses before sending

- Bounce management
- Unsubscribe handling

Troubleshooting

Common Issues

1. Emails Not Sending

- Verify RESEND_API_KEY is set in .env.local
- Check Resend dashboard for API status
- Verify from email domain is configured

2. Authentication Errors

- Ensure user has ADMIN role
- Check NextAuth configuration
- Verify session is active

3. Database Errors

- Run `npx prisma generate` after schema changes
- Run `npx prisma db push` to sync schema
- Check database connection

4. Type Errors

- Ensure EmailNotificationType enum includes all used types
- Run TypeScript build to catch errors early

File Structure

```

mindful_champion/nextjs_space/
  -- app/
    -- admin/
      -- emails/
        -- page.tsx # Admin email management page
    -- api/
      -- admin/
        -- emails/
          -- send/
            -- route.ts # Send email endpoint
          -- resend/
            -- route.ts # Resend email endpoint
          -- history/
            -- route.ts # Email history endpoint
          -- test/
            -- route.ts # Test email endpoint
    -- components/
      -- admin/
        -- email/
          -- admin-email-management.tsx # Main email management component
    -- lib/
      -- email/
        -- resend-client.ts # Resend client configuration
    -- prisma/
      -- schema.prisma # Database schema with EmailNotification
  model
    -- scripts/
      -- send-test-emails.ts # Test email script

```

Support

For issues or questions regarding the email management system:

- Contact: lee@mindfulchampion.com
 - System Administrator: ADMIN role users
-

Last Updated: November 30, 2024

Version: 1.0.0

Status:  Production Ready