

Email Service Guide

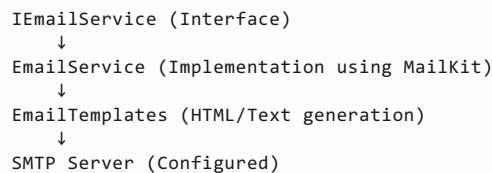
Overview

The IKEA DocuScan application includes a comprehensive email service for sending notifications and documents via email. The service uses **MailKit** for reliable, modern SMTP communication.

Key Features

- **Access Request Notifications** - Admin receives emails when users request access
 - **User Confirmations** - Users receive confirmation of their access requests
 - **Document Links** - Send document links via email
 - **Document Attachments** - Send documents as email attachments
 - **Batch Operations** - Send multiple documents in one email
 - **HTML Templates** - Professional, branded email templates
 - **Audit Trail Integration** - All email sends are logged
 - **Configuration-Driven** - Easy to enable/disable features
 - **Error Handling** - Robust error handling and logging
-

Architecture



Configuration

appsettings.json

```
{  
  "Email": {  
    "SmtpHost": "smtp.company.com",  
    "SmtpPort": 587,  
    "UseSsl": true,  
    "SmtpUsername": "docuscan@company.com",  
    "SmtpPassword": "",  
  
    "FromAddress": "noreply-docuscan@company.com",  
    "FromDisplayName": "IKEA DocuScan System",  
    "AdminEmail": "docuscan-admin@company.com",  
    "AdditionalAdminEmails": ["it-support@company.com"],  
  
    "EnableEmailNotifications": true,  
    "SendAccessRequestNotifications": true,  
    "SendDocumentNotifications": true,  
  
    "AccessRequestSubject": "New Access Request - IKEA DocuScan",  
    "AccessRequestConfirmationSubject": "Your Access Request Has Been Received",  
    "ApplicationUrl": "https://docuscan.company.com",  
    "TimeoutSeconds": 30  
  }  
}
```



Configuration Options

Option	Type	Default	Description
SmtpHost	string	“smtp.company.com”	SMTP server hostname
SmtpPort	int	587	SMTP server port (587=TLS, 465=SSL, 25=unencrypted)
UseSsl	bool	true	Use SSL/TLS for SMTP connection
SmtpUsername	string	“”	SMTP authentication username
SmtpPassword	string	“”	SMTP authentication password (encrypt in production!)
FromAddress	string	Required	Email address to send from
FromDisplayName	string	“IKEA DocuScan System”	Sender display name
AdminEmail	string	Required	Primary administrator email
AdditionalAdminEmails	string[]	[]	Additional admin emails to CC
EnableEmailNotifications	bool	true	Master switch for all email
SendAccessRequestNotifications	bool	true	Enable access request emails
SendDocumentNotifications	bool	true	Enable document emails
AccessRequestSubject	string	Customizable	Subject for access request emails
ApplicationUrl	string	Required	Base URL for links in emails
TimeoutSeconds	int	30	SMTP operation timeout

Secure Configuration

For Production - Use Encrypted Configuration:

```
# Add SMTP password to encrypted config
cd ConfigEncryptionTool
dotnet run
# Enter SMTP password when prompted
```

Or use environment variables:

```
export Email__SmtpPassword="your-password-here"
```

Usage

1. Access Request Notifications

Automatically sent when user requests access:

```
// In CurrentUserService.RequestAccessAsync()
await _emailService.SendAccessRequestNotificationAsync(username, reason);
```

Email content includes: - Username requesting access - Request timestamp - Reason for access (if provided) - Instructions for granting access - Link to DocuScan system

Recipients: - Primary admin (AdminEmail) - Additional admins (AdditionalAdminEmails)

2. Send Document Link

Send a link to a document:

```
// In DocumentService or custom controller
await documentService.SendDocumentLinkAsync(
    barCode: "123456",
    recipientEmail: "user@company.com",
    message: "Please review this contract");
```

Features: - Generates document link automatically - Includes custom message - Logs to audit trail (SendLink action) - Professional email template

3. Send Document Attachment

Send document as email attachment:

```
byte[] fileData = File.ReadAllBytes("document.pdf");

await documentService.SendDocumentAttachmentAsync(
    barCode: "123456",
    recipientEmail: "user@company.com",
    fileData: fileData,
    fileName: "Contract_123456.pdf",
    message: "Contract for your review");
```

Features: - Supports multiple file types (PDF, images, Office docs) - Automatic MIME type detection - Logs to audit trail (SendAttachment action) - File size validation

4. Send Multiple Document Links

Send links to multiple documents in one email:

```
var barCodes = new[] { "123456", "123457", "123458" };

await documentService.SendDocumentLinksAsync(
    barCodes: barCodes,
    recipientEmail: "user@company.com",
    message: "All contracts for Q4 2024");
```

Features: - Single email with all links - Skips invalid bar codes - Logs batch operation to audit trail - Efficient for multiple documents

5. Send Multiple Document Attachments

Send multiple documents as attachments:

```

var documents = new[]
{
    (Barcode: "123456", FileData: file1Data, FileName: "doc1.pdf"),
    (Barcode: "123457", FileData: file2Data, FileName: "doc2.pdf")
};

await documentService.SendDocumentAttachmentsAsync(
    documentsToSend: documents,
    recipientEmail: "user@company.com",
    message: "All required documents");

```

Features: - Multiple attachments in one email - Validates all documents exist - Logs batch operation - Size limits apply (check SMTP server limits)

6. Custom Email

Send custom email with IEmailService:

```

@inject IEmailService EmailService

await EmailService.SendEmailAsync(
    toEmail: "user@company.com",
    subject: "Custom Notification",
    htmlBody: "<h1>Hello</h1><p>This is a custom email</p>",
    plainTextBody: "Hello\n\nThis is a custom email",
    attachments: null);

```

Email Templates

All emails use branded HTML templates with IKEA colors (#0051A5).

Template Features

- **Responsive Design** - Works on desktop and mobile
- **Professional Styling** - IKEA brand colors
- **Plain Text Fallback** - For email clients that don't support HTML
- **Structured Layout** - Header, content, footer
- **Accessible** - Good contrast, readable fonts

Template Customization

Templates are defined in: - **File:** /Services/EmailTemplates.cs - **Methods:** BuildAccessRequestNotification(), BuildDocumentLink(), etc.

To customize: 1. Edit the HTML in EmailTemplates.cs 2. Modify colors, fonts, or layout
3. Rebuild and deploy

Common SMTP Configurations

Microsoft 365 / Outlook

```

{
    "Email": {
        "SmtpHost": "smtp.office365.com",
        "SmtpPort": 587,
        "UseSsl": true,
        "SmtpUsername": "docuscan@yourcompany.com",
        "SmtpPassword": "your-password"
    }
}

```

Gmail

```
{  
  "Email": {  
    "SmtpHost": "smtp.gmail.com",  
    "SmtpPort": 587,  
    "UseSsl": true,  
    "SmtpUsername": "your-email@gmail.com",  
    "SmtpPassword": "app-specific-password"  
  }  
}
```

Note: Gmail requires app-specific password, not your regular password.

On-Premise Exchange Server

```
{  
  "Email": {  
    "SmtpHost": "mail.yourcompany.local",  
    "SmtpPort": 25,  
    "UseSsl": false,  
    "SmtpUsername": "",  
    "SmtpPassword": ""  
  }  
}
```

SendGrid

```
{  
  "Email": {  
    "SmtpHost": "smtp.sendgrid.net",  
    "SmtpPort": 587,  
    "UseSsl": true,  
    "SmtpUsername": "apikey",  
    "SmtpPassword": "your-sendgrid-api-key"  
  }  
}
```

Testing

Development Mode

Disable emails during development:

```
{  
  "Email": {  
    "EnableEmailNotifications": false  
  }  
}
```

Emails will be logged but not sent.

Test SMTP Servers

Papercut SMTP (Windows): - Download:

<https://github.com/ChangemakerStudios/Papercut-SMTP> - Captures emails locally without sending - Great for development

MailTrap (Online): - Website: <https://mailtrap.io> - Free tier available - Inspect emails in browser

Configuration for test tools:

```
{  
  "Email": {  
    "SmtpHost": "localhost",  
    "SmtpPort": 25,  
    "UseSsl": false,  
    "SmtpUsername": "",  
    "SmtpPassword": ""  
  }  
}
```

Manual Testing

```
// Test access request notification  
await emailService.SendAccessRequestNotificationAsync("testuser", "Testing  
  
// Test document Link  
await documentService.SendDocumentLinkAsync("123456", "test@company.com");  
| |
```

Troubleshooting

Emails Not Sending

Check: 1. EnableEmailNotifications = true 2. SMTP host, port, and credentials are correct 3. Firewall allows outbound SMTP traffic 4. Check application logs for errors

View logs:

```
2024-01-15 10:30:00 [Information] Email sent successfully: New Access Request - IKEA DocuScan to 1 recipient(s)  
2024-01-15 10:31:00 [Error] Failed to send email: Connection refused
```

Authentication Failures

Symptoms: - “Authentication failed” errors in logs - “535 Authentication credentials invalid”

Solutions: - Verify username/password - Check if account requires app-specific password (Gmail, Office365 with MFA) - Ensure account has SMTP sending permissions

Connection Timeouts

Symptoms: - “Connection timed out” errors - Slow email sending

Solutions: - Check firewall rules - Verify SMTP port is correct - Increase TimeoutSeconds in configuration - Check if SMTP server is accessible from app server

SSL/TLS Errors

Symptoms: - “SSL handshake failed” - “Certificate validation failed”

Solutions: - Try UseSsl: false for internal servers - Verify certificate is valid - For self-signed certs, may need to add cert to trusted store

Emails Go to Spam

Solutions: - Configure SPF record for sending domain - Set up DKIM signing - Use authenticated SMTP - Avoid spam trigger words in subject/body - Send from reputable domain

Security Best Practices

1. Secure Credentials

Never commit SMIP password to source control!

Use encrypted configuration:

```
cd ConfigEncryptionTool  
dotnet run
```

Or environment variables:

```
export Email__SmtpPassword="password"
```

2. Validate Email Addresses

The service validates email addresses, but you should also:
- Validate in UI before submission
- Implement rate limiting
- Log all email sends

3. Attachment Security

- Validate file types before sending
- Limit attachment sizes
- Scan for malware if required
- Never send passwords or sensitive credentials

4. Link Security

- Use HTTPS for all document links
- Implement link expiration if needed
- Require authentication to view documents
- Log all link access

Performance Considerations

Async Operations

All email operations are async and non-blocking:

```
// Email is sent in background  
await _emailService.SendDocumentLinkAsync(...);  
  
// Request continues immediately  
// Email failures don't break the application
```

Email Send Times

- **Local SMIP:** ~50-200ms
- **External SMIP (Gmail, Office365):** ~500-2000ms
- **With large attachments:** ~5-30 seconds

Batch Operations

Use batch methods for multiple documents:

```

// GOOD - Single email with multiple links
await SendDocumentLinksAsync(barCodes, email);

// BAD - Multiple emails
foreach (var barCode in barCodes)
{
    await SendDocumentLinkAsync(barCode, email);
}

```

Background Queue (Future Enhancement)

For high-volume scenarios, consider implementing:

- `IBackgroundTaskQueue` for non-blocking sends
- Retry logic with exponential backoff
- Failed email queue for later retry

Audit Trail Integration

All email operations are automatically logged to the audit trail:

Action	Audit Trail Entry
Send Link	<code>AuditAction.SendLink</code>
Send Attachment	<code>AuditAction.SendAttachment</code>
Send Links	<code>AuditAction.SendLinks</code>
Send Attachments	<code>AuditAction.SendAttachments</code>

Query audit trail:

```

SELECT * FROM AuditTrail
WHERE ActionType = 'SendLink'
    AND BarCode = '123456'
ORDER BY ActionDate DESC

```

File Reference

- **IEmailService Interface:** /IkeaDocuScan.Shared/Interfaces/IEmailService.cs
 - **EmailService Implementation:** /IkeaDocuScan-Web/Services/EmailService.cs
 - **Email Templates:** /IkeaDocuScan-Web/Services/EmailTemplates.cs
 - **EmailOptions Configuration:**
/IkeaDocuScan.Shared/Configuration/EmailOptions.cs
 - **DocumentService Integration:** /IkeaDocuScan-Web/Services/DocumentService.cs:324-478
 - **CurrentUserService Integration:** /IkeaDocuScan-Web/Services/CurrentUserService.cs:227-237
 - **Configuration:** /IkeaDocuScan-Web/appsettings.json:16-33
-

NuGet Packages Required

```

<PackageReference Include="MailKit" Version="4.3.0" />
<PackageReference Include="MimeKit" Version="4.3.0" />

```

Future Enhancements

Potential improvements to consider:

1. **Background Queue** - Non-blocking email sends with retry
2. **Email Templates from Database** - Allow admin to customize templates

3. **Attachments from File Storage** - Auto-attach from document storage
 4. **Email Tracking** - Track opens and clicks
 5. **Unsubscribe Management** - Allow users to opt-out of notifications
 6. **Email Scheduling** - Send emails at specific times
 7. **Bulk Email** - Send to multiple recipients efficiently
 8. **Email Analytics** - Track send success rates, bounces
 9. **Template Variables** - More dynamic content in templates
 10. **Localization** - Multi-language email templates
-

Support

For issues or questions about the email service:

1. Check application logs
 2. Review this documentation
 3. Test with a local SMTP server (Papercut)
 4. Contact your IT department for SMTP server access
-

Summary

The email service provides:

- Easy Integration** - Inject `IEmailService` and call methods
- Professional Templates**
- Branded HTML emails
- Comprehensive Logging** - All sends logged
- Error Handling** - Robust error handling
- Configurable** - Easy to customize per environment
- Secure** - Encrypted credentials, validation
- Audit Trail** - Integrated with existing audit system
- MailKit** - Modern, reliable SMTP library

The service is production-ready and requires only SMTP configuration to start using!

Using smtp4dev on Marks dev box:

run C:\Smtp4dev\Microsoft.Winget.Source_8wekyb3d8bbwe.Smtp4dev.exe

navigte to <http://localhost:5000> to see the web interface.