

Email System Documentation

Overview

The ASP.NET application includes an integrated email system capable of sending asynchronous email messages using SMTP (Simple Mail Transfer Protocol). The email system utilizes a dedicated method:

`SendEmail(string recipientEmail, string subject, string body)`

This method facilitates sending emails with custom subjects and message bodies to specified recipients.

Method Signature

`public static async Task SendEmail(string recipientEmail, string subject, string body)`

- *recipientEmail*: A string containing the email address of the recipient.
- *subject*: A string representing the subject line of the email.
- *body*: A string containing the body of the email. This parameter accepts HTML content.























Implementation Details

1. SMTP Server Configuration:

- The method uses an SMTP server tied to a Gmail account which is used as the sender.
- Server credentials are not included in the source control for security reasons.
 1. NOTE: the password in the credentials is a google app password generated in the Gmail account, NOT THE GMAIL ACCOUNT PASSWORD
- External users or future student groups requiring access to the login credentials should contact SCUTTW@ETSU.EDU or grantscutt2@gmail.com, using a school account for authentication.

2. Password Retrieval:

- The password for the SMTP server is stored in a separate file (gmail.txt) and is read from the file at runtime.
 1. NOTE: in order for the file to be read correctly, gmail.txt needs to be in the ScavengeRUs\ScavengeRUs directory (the same directory the database files are in, the root directory for the project) ((picture below for clarity))

 .config	11/1/2023 12:26 PM	File folder	
 Areas	11/1/2023 12:26 PM	File folder	
 Attributes	11/20/2023 12:16 PM	File folder	
 bin	11/15/2023 12:03 PM	File folder	
 Controllers	12/6/2023 10:59 AM	File folder	
 Data	11/1/2023 12:26 PM	File folder	
 Models	11/1/2023 12:26 PM	File folder	
 obj	11/25/2023 11:17 PM	File folder	
 Properties	11/25/2023 3:57 PM	File folder	
 Services	12/2/2023 7:43 PM	File folder	
 Views	11/1/2023 12:26 PM	File folder	
 wwwroot	11/1/2023 12:26 PM	File folder	
 appsettings.Development.json	11/1/2023 12:26 PM	JSON File	1 KB
 appsettings.json	11/1/2023 12:26 PM	JSON File	1 KB
 gmail.txt	11/25/2023 6:22 PM	Text Document	1 KB
 Program.cs	11/27/2023 12:28 PM	C# Source File	2 KB
 Readme.txt	11/1/2023 12:26 PM	Text Document	1 KB
 ScavengeRUS.csproj	11/25/2023 11:17 PM	C# Project File	2 KB
 ScavengeRUS.db	11/29/2023 11:53 AM	Data Base File	168 KB
 ScavengeRUS.db-shm	11/29/2023 12:04 PM	DB-SHM File	32 KB
 ScavengeRUS.db-wal	11/29/2023 12:06 PM	DB-WAL File	338 KB
 scrum-bums-042e94718ef6.json	11/1/2023 12:26 PM	JSON File	3 KB

- The secretsFileName variable contains the filename for the password file.

3. SMTP Client Setup:

- SmtplibClient: Configured with the server URL and port number (variables smtpURL and smtpPort need to be defined).
- Credentials are set using an application-specific email address and the password retrieved from the file.
- SSL encryption is enabled for security.

4. Email Message Construction:

- A MailMessage object is created with the application's email, the recipient's email, the subject, and the body.
- The message body is treated as HTML content.

5. Sending the Email:

- The email is sent using the configured SmtplibClient instance.
- Potential exceptions such as SmtplibException are not explicitly handled within this method. Callers should be prepared to handle such exceptions as suited to the application context.

Important Notes

- *Security Reminder:* Storing email passwords or sensitive information in code or unsecured files can expose the system to significant security risks. Ensure the secrets file is properly protected and access-restricted. **THERE IS A RISK THAT IF SOMEONE MALICIOUS GAINS ACCESS TO THE EMAIL THEY CAN SEND / SPAM EMAILS AND TEXTS WITH ARBITRARY STRINGS OR HTML CODE. THIS SHOULD BE MAKING ANY CSMN PEOPLE VERY UNCOMFORTABLE!**
- *Exception Handling:* The method currently does not handle exceptions that might be thrown by the client.Send method. It is recommended that exception handling is implemented by the caller or within the method itself, depending on the application's error handling strategy.
- *Asynchrony:* Although the method signature suggests asynchrony with the Task return type, the method currently operates synchronously. To make it truly asynchronous, consider using SendMailAsync instead of Send and awaiting the call.