**Email Automation Documentation**

**1. Introduction**

Email Automation is a console application designed to provide automation for logging customer support calls. This powerful tool is utilized by the support team to streamline the process of handling customer support requests received via email. The application connects to the Outlook database using the Graph API, reads incoming unread emails, and extracts relevant data based on a predefined format. The extracted data is then used to log support calls automatically, leading to efficient issue resolution and improved customer satisfaction.

**2. Purpose and Usage**

The primary purpose of Email Automation is to automate the process of logging support calls received via email. The application acts as a service to handle customer support needs efficiently, reducing the need for manual intervention and ensuring prompt issue resolution. Whenever a customer sends an email seeking support, the support team runs the Email Automation app to process the incoming emails and initiate the support ticket creation process.

**3. Application Workflow**

The Email Automation application follows a structured workflow to process incoming emails and log support calls:

**3.1. Mail Retrieval**

- The application connects to the Outlook database using the Microsoft Graph API.

- It reads and retrieves all incoming unread emails.

**3.2. Data Extraction**

- The application checks the content of each email to identify a predefined format or pattern.

- Relevant data, such as customer details, product serial number, issue description, and other required information, is extracted from the email content.

**3.3. Support Ticket Creation**

- After data extraction, the application processes the data and saves it in the appropriate folder based on the result:

- If the email follows the predefined format, the data is saved in our Oracle database, and a support ticket is automatically logged.

- If the email does not match the expected format, it is marked as an invalid format.

**4. Technology Stack**

The Email Automation application is built using the following technologies:

- Node.js: The application is developed using Node.js to enable asynchronous processing.

- PM2: PM2 is used to manage the application, ensuring continuous availability and automatic restarts in case of failure.

- Microsoft Graph API: The Graph API is employed to connect to the Outlook database and perform various tasks, including email retrieval and folder management.

**5. Benefits**

Email Automation provides numerous benefits to the support team and the organization as a whole:

- Improved Customer Satisfaction: Automation leads to faster response times and quicker issue resolution, enhancing the overall customer experience.

- Cost Reduction: By automating support ticket creation, manual efforts are minimized, resulting in cost savings.

- Error Reduction: Automation significantly reduces the risk of human errors in the support call logging process.

1. **Challenges**

- Integrating the application with the API to dynamically associate email IDs with their respective parsing folders can be complex.

- Carefully design and implement an API integration mechanism that efficiently maps email IDs to their designated parsing folders. Test thoroughly to ensure accurate association and data extraction.

1. **Troubleshooting and Logs**

- In case of issues or errors during the setup or operation of Email Automation, refer to the application logs generated by PM2. Logs are available in the directory specified by PM2.

1. **Logging and Error Handling**

- Creating and maintaining a robust automated testing suite for various email scenarios can be time-consuming.

- Invest in automated testing tools, simulate diverse email scenarios, and regularly update test cases to reflect changes in the application's logic.

**9. Benefits**

Email Automation serves as a valuable tool for the support team, streamlining the logging of customer support calls received via email. With the capability to connect to the Outlook database through the Microsoft Graph API, the application efficiently extracts data and automates the ticket creation process. This results in enhanced customer satisfaction, reduced costs, and improved accuracy in managing support requests. The asynchronous nature of the application, along with the use of Node.js and PM2, ensures seamless operation and reliability for effective customer support management.