Email is the most common and necessary requirement for the majority of applications. Java Mail API is a platform and protocol independent framework for developing mail and messaging applications. The JavaMail reference implementation is licensed under the GNU General Public License (GPL) v2 with Classpath Exception and the Common Development and Distribution License (CDDL) v1.1.

For the purpose of alert generation, an automated email mechanism is incorporated for sending email as soon as an intruder is detected by the surveillance system. The email mechanism is available through an API. The HTTP post method sends an API request to email server with request body containing intended receiver email and content.

The send an email following modules are used:

* **simpleMailMessage:**

is used to models a simple mail message, it including data such as the from, to, cc, subject, and text fields.

* **JavaMailSender:**

Is a JavaMail Extended MailSender interface that also supports MIME messages as direct arguments and via preparation callbacks. Typically used in conjunction with the MimeMessageHelper class to simplify the creation of JavaMail MimeMessages, including attachments and so on.

* **SMTP**

Simple Mail Transfer Protocol (SMTP) is an application used by mail servers to send, receive, and/or relay outgoing email between email senders and receivers.

* **Port**

An SMTP port is a communication endpoint that defines the routing of email transactions from server to server across the network. An “SMTP port” refers to the specific part of the Internet address that’s used to transfer email. Port 465 used for implicit SSL/TLS, whereas port 587 or 2525 used for explicit SSL/TLS connections.

* **Host**

When you use SMTP host Gmail or AOL to send an email, the SMTP server processes your email, determines which server to send the message to, and relays the message to that server. The message is then downloaded and delivered to the recipient's inbox service provider, such as Gmail or AOL.

* **Email**

It is use to specify the sender’s email according to the SMTP host

* **Password**

It is use to specify the sender’s password according to the SMTP host. For security purpose smtp host donot allow non secure apps to send email. For this purpose app passwords generated for passwords.

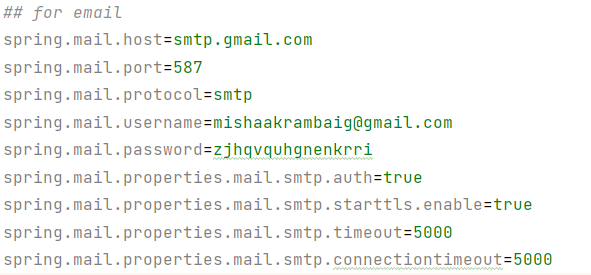
* **TLS**

Transport Layer Security (TLS) is a security protocol that encrypts email for privacy. TLS prevents unauthorized access of your email when it's in transit over internet connections.

* **Starttls**

StartTLS is a protocol command that informs the email server that the email client wishes to upgrade from an insecure to a secure connection via TLS or SSL.

The java API is created using Spring Boot. Spring Boot is a Java framework, built on top of the Spring, used for developing web applications. It allows you to create REST APIs with minimal configurations. The API when invoked calls java’s built in simpleMailMessage function and sets parameters like sender, receiver, title and body text and then sends on the mail using the JavaMailSender function. The email server uses SMTP (Simple Mail Transfer protocol) for sending email. Here **smtp.gmail.com** is used as host with port **587** configured. By default, Gmail always tries to use a secure TLS connection when sending email. However, a secure TLS connection requires that both the sender and recipient use TLS. If the receiving server doesn't use TLS, Gmail still delivers messages, but the connection isn't secure. Transport Layer Security (TLS) is a security protocol that encrypts email for privacy. TLS prevents unauthorized access of your email when it's in transit over internet connections. For secure connections **Secure transport (TLS) compliance**setting to always use TLS for email sent to and from domains and addresses specified.

This is initial properties set for email configuration:

The below code is using SImpleJavaMail to generate basic format and JavaMAilSender to send mail.



Finally the email API is available with the endpoint “/sendmail”

