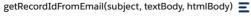


Developers







Apex Reference Guide / System Namespace / EmailMessages Class / EmailMessages

getRecordIdFromEmail(subject, textBody, htmlBody)

Returns the record ID corresponding to the specified email threading token, or $r\varepsilon$ is found.

Signature

public static Id getRecordIdFromEmail(String subject, String textBody, String

Parameters

subject

Type: String

The subject of the email.

textBody

Type: String

The body of the email in text format.

htmlBody

Type: String

The body of the email in HTML format.

Return Value

Type: Id

The record ID that corresponds to the embedded threading token.

Usage

When you send emails with threading tokens embedded in the email subject, the both the subject and body, most email clients quote the email body and maintai subject in a response. This method finds a corresponding record that matches th threading token in a response.

Typically this method is used in Email Services so that you can provide your own I inbound emails using Apex code.

Example

If you implement header-based threading in your Email Services currently, we recuse Lightning threading, which combines token-based threading and header-base header-based threading to continue to work, store emails as EmailMessage recor MessagedIdentifier field set properly. With Lightning threading, you can use threathe primary threading method and rely on header-based threading as a fallback,



V

```
global Messaging.InboundEmailResult handleInboundEmail(Messaging.int
              Messaging.InboundEnvelope env) {
      // Create an InboundEmailResult object for returning the result
      // Apex Email Service.
      Messaging.InboundEmailResult result = new Messaging.InboundEmail
      // Try to find the Case ID using threading tokens in email attri
      Id caseId = EmailMessages.getRecordIdFromEmail(email.subject, en
      // If we haven't found the Case ID, try finding it using headers
      if (caseId == null) {
          caseId = Cases.getCaseIdFromEmailHeaders(email.headers);
      // If a Case isn't found, create a new Case record.
      if (caseId == null) {
          Case c = new Case(Subject = email.subject);
          insert c:
          System.debug('New Case Object: ' + c);
          caseId = c.Id;
      // Process recipients
      String toAddresses;
      if (email.toAddresses != null) {
          toAddresses = String.join(email.toAddresses, '; ');
      \ensuremath{//} To store an EmailMessage for threading, you need at minimum
      // the Status, the MessageIdentifier, and the ParentId fields.
      EmailMessage em = new EmailMessage(
         Status = '0',
          MessageIdentifier = email.messageId,
          ParentId = caseId,
          // Other important fields.
          FromAddress = email.fromAddress.
          FromName = email.fromName,
          ToAddress = toAddresses,
          TextBody = email.plainTextBody,
          HtmlBody = email.htmlBody,
          Subject = email.subject,
          // Parse thread-index header to remain consistent with Email
          ClientThreadIdentifier = getClientThreadIdentifier(email.hea
          // Other fields you wish to add.
      );
      // Insert the new EmailMessage.
      insert em:
      System.debug('New EmailMessage Object: ' + em );
  // Set the result to true. No need to send an email back to the user
  // with an error message.
  result.success = true;
  // Return the result for the Apex Email Service.
  return result;
private String getClientThreadIdentifier(List<Messaging.InboundEmail.F</pre>
 if (headers == null || headers.size() == 0) return null;
  try {
      for (Messaging.InboundEmail.Header header : headers) {
          if (header.name.equalsIgnoreCase('thread-index')) {
              Blob threadIndex = EncodingUtil.base64Decode(header.valu
              return EncodingUtil.convertToHex(threadIndex).substring(
          }
  } catch (Exception e){
     return null;
  }
  return null:
```





DID THIS ARTICLE SOLVE YOUR ISSUE?

Let us know so we can improve!





Podcasts

AppExchange

© Copyright 2025 Salesforce, Inc. <u>All rights reserved.</u> Various trademarks held by their respective owners. S Salesforce Tower, 415 Mission Street, 3rd Floor, San Francisco, CA 94105, United States

Privacy Information Terms of Service Legal Use of Cookies Trust Cookie Preferences

Your Privacy Choices Responsible Disclosure Contact

Einstein

Quip