

## Requirement

- An email service is an Apex class that implements the Messaging.InboundEmailHandler interface which allows you to process the contents, headers and attachments of an email sent to the service's associated email address.
- When the service receives an email, it will use the subject of the email (e.g., Mike Smith) and try and find a contact in Salesforce by that name (if there are multiple contacts, simply use the first one found).
- If the record is found, it adds the attachment to the Chatter feed for the contact record and any text in the body of the email as the body for the Chatter post feed.
- Use the email address of the sender, find a user with that email address and set that user as the creator of the Chatter post.
- Must have adequate test coverage (75%+).

## Assumptions

- Complete email subject will be searched to match Contact Name field (ie Subject = Name) i.e., no subject trimming is done, no like operators used in SOQL , as the requirement doesn't demand to do so (It can be done if required).
- Only if a match (exact name match) is found, the code will process further (this is also a requirement)
- Email from address will be searched (across User's Email field) to find the feeditem creator. Here first user match will be selected (this is also a requirement).

## Setup Steps

1. Create Inbound Email Handler Apex Class "Email2Chatter" (Setup -> Develop -> Apex Classes -> New and copy the code from attached [Email2Chatter.cls](#) and Save)
2. Create Email Service (Setup -> Develop -> Email Services -> New Email Service). Fill in necessary details (and make sure you select the apex class you created in step1) and then click "Save and New Email Address"

### Email Service

Save Save and New Email Address Cancel

**Email Service Information**

Email Service Name: Email2Chatter

Apex Class: Email2Chatter

Accept Attachments: All

Advanced Email Security Settings:

Accept Email From:

Convert Text Attachments to Binary Attachments:

Active:

**Failure Response Settings**

Configure how salesforce.com responds when an attempt to access this email service fails for the reasons shown below.

Over Email Rate Limit Action: Discard message

Deactivated Email Address Action: Discard message

Deactivated Email Service Action: Discard message

Unauthenticated Sender Action: Discard message

Unauthorized Sender Action: Discard message

Enable Error Routing:

Route Error Emails to This Email Address: jain.chirag@gmail.com

Save Save and New Email Address Cancel

3. Provide email service name and click save

### Email Service Address

Specify an email address for this email service. The email service processes messages sent to this address. One email service

**Email Service Information**

Email Service Name: Email2Chatter

Accept Email From: All email addresses (subject to security settings)

**Email Address Information**

Email address: email2chatter@3uxtwho9bhng7wjryy5f5h.36hmbeai.3.apex.salesforce.com

Active:

Context User: Chirag Mehta

Accept Email From: jain.chirag@gmail.com

Save Save and New Cancel

- So now you are all set with email service (and you can start testing as mentioned in below Test Cases section)

### Email Service: Email2Chatter

Email Service Name	Email2Chatter
Apex Class	<a href="#">Email2Chatter</a>
Accept Attachments	All
Advanced Email Security Settings	<input type="checkbox"/> <a href="#">i</a>
Accept Email From	All email addresses (subject to security settings)
Convert Text Attachments to Binary Attachments	<input type="checkbox"/> <a href="#">i</a>
Active	<input checked="" type="checkbox"/>

---

**▼ Failure Response Settings**

Over Email Rate Limit Action	Discard message
Deactivated Email Address Action	Discard message
Deactivated Email Service Action	Discard message
Unauthenticated Sender Action	Discard message
Unauthorized Sender Action	Discard message
Enable Error Routing	<input checked="" type="checkbox"/> <a href="#">i</a>
Route Error Emails to This Email Address	<a href="mailto:jain.chirag@gmail.com">jain.chirag@gmail.com</a>

---

**Email Addresses**

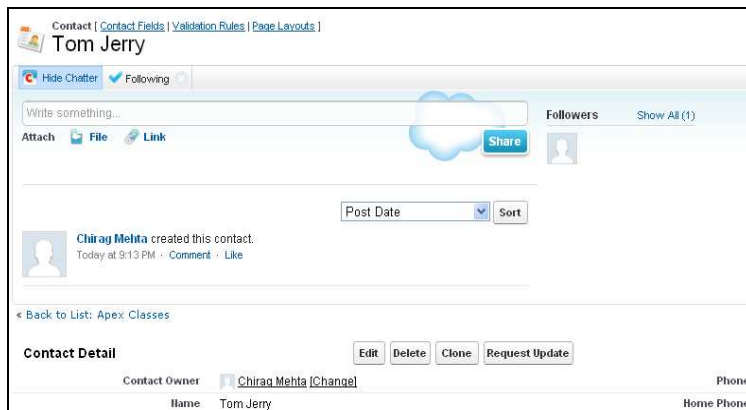
Action	Email Address
<a href="#">View</a>   <a href="#">Edit</a>	<a href="mailto:email2chatter@3uxtwho9bhng7wivr5f5h36hmbeai.3.apex.salesforce.com">email2chatter@3uxtwho9bhng7wivr5f5h36hmbeai.3.apex.salesforce.com</a>

- Create Test Apex Class "Test\_Email2Chatter" (Setup -> Develop -> Apex Classes -> New and copy the code from attached [Test\\_Email2Chatter.cls](#) and Save) (Code Coverage is 100% -☺)

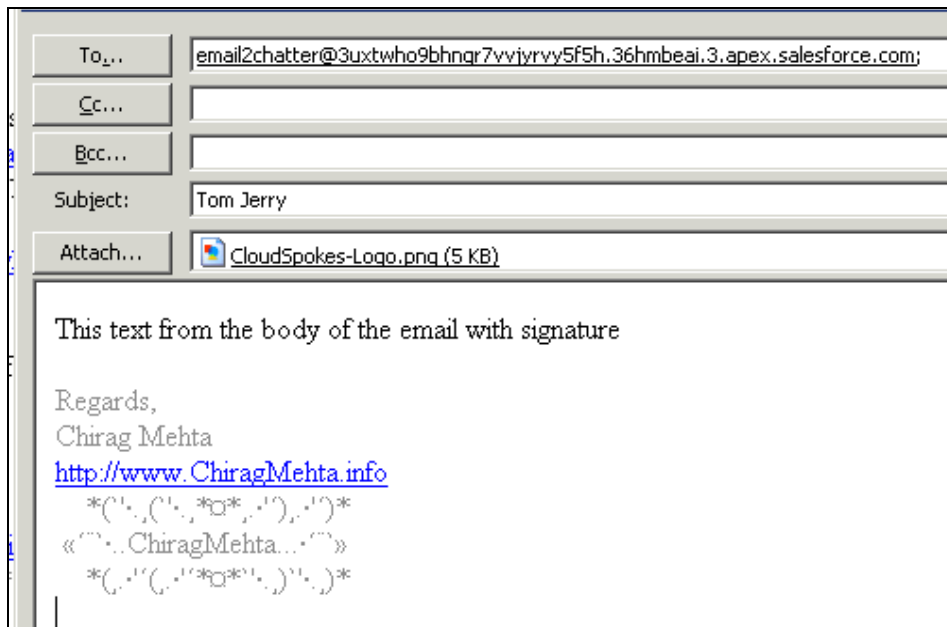
## Test Cases (with few Pie-Points)

1. Created Email Service which accepts incoming email and generates chatter feed with details (as requested in requirement)

User1 “Chirag Mehta” created Contact viz., “Tom Jerry” (as you see there’s no chatter stream currently for this contact)



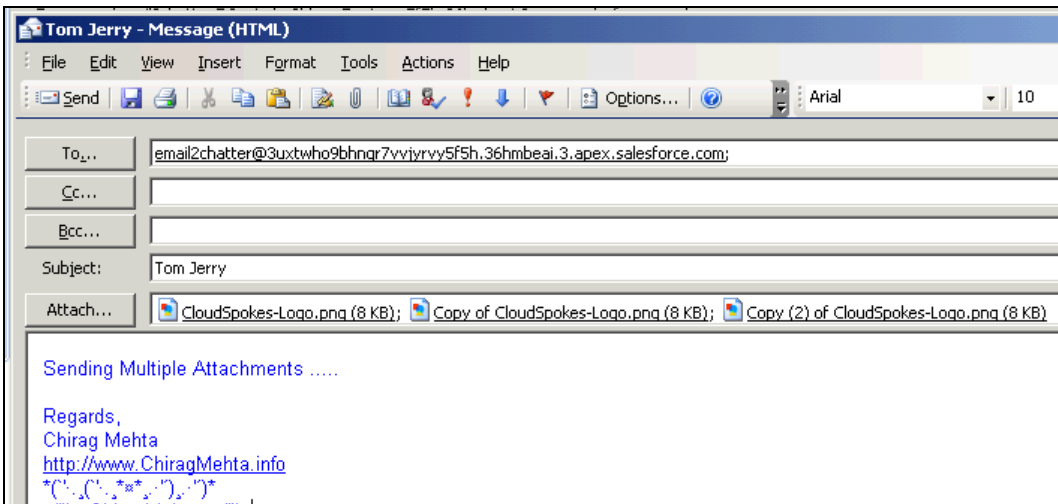
Send email to Email Service (with subject = contact name) Here sender email is matching user User2 “Deepak Mehta” in my developer org



Stream shows Chatter feed with body = email body, content = email attachment and createdbyid = email sender



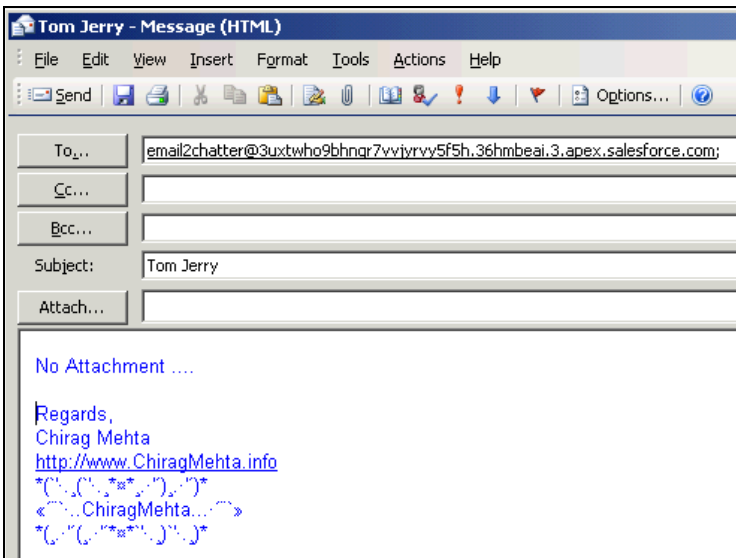
- Added handling of multiple attachments too : If email with multiple attachments, then email service will insert multiple feeds with one attachment each



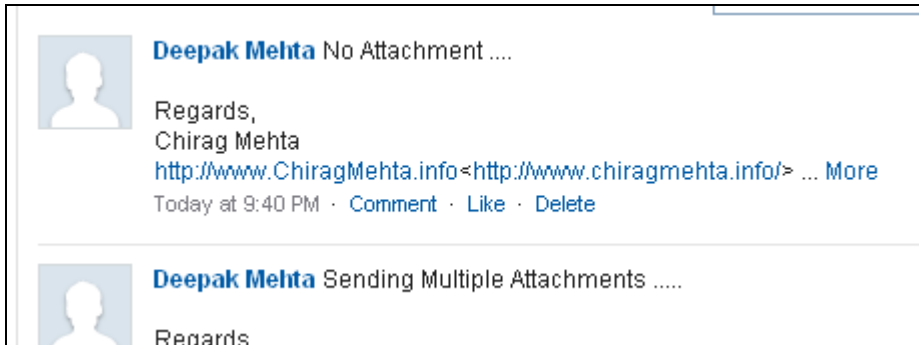
3 attachments, so 3 chatter posts inserted by email service



- Added handling of emails with zero attachments too : if email has no attachments, then email service will insert text feed only



Chatter post with no content inserted



## Attached/Enclosed (Eclipse export of below items)

- ❖ Apex Classes
  - Email2Chatter.cls
  - Test\_Email2Chatter.cls