JAVAMAIL API - REPLYING EMAILS

In this chapter we will see how to reply to an email using JavaMail API. Basic steps followed in the program below are:

- Get the Session object with POP and SMPT server details in the properties. We would need POP details to retrieve messages and SMPT details to send messages.
- Create POP3 store object and connect to the store.
- Create Folder object and open the appropriate folder in your mailbox.
- Retrieve messages.
- Iterate through the messages and type "Y" or "y" if you want to reply.
- Get all information *To*, *From*, *Subject*, *Content* of the message.
- Build the reply message, using Message.reply method. This method configures a new Message with the proper recipient and subject. The method takes a boolean parameter indicating whether to reply to only the sender *false* or reply to all *true*.
- Set From,Text and Reply-to in the message and send it through the instance of Transport object.
- Close the Transport, folder and store objects respectively.

> *Here we have used JangoSMPT server via which emails are sent to our destination email address. The setup is explained in the Environment Setup chapter.*

Create Java Class

Create a java class file **ReplyToEmail**, the contents of which are as follows:

```
package com.tutorialspoint;

import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.util.Date;
import java.util.Properties;

import javax.mail.Folder;
import javax.mail.Message;
import javax.mail.Session;
import javax.mail.Store;
import javax.mail.Transport;
import javax.mail.internet.InternetAddress;
import javax.mail.internet.MimeMessage;

public class ReplyToEmail {
   public static void main(String args[])
   {
      Date date = null;

      Properties properties = new Properties();
      properties.put("mail.store.protocol", "pop3");
      properties.put("mail.pop3s.host", "pop.gmail.com");
      properties.put("mail.pop3s.port", "995");
      properties.put("mail.pop3.starttls.enable", "true");
      properties.put("mail.smtp.auth", "true");
      properties.put("mail.smtp.starttls.enable", "true");
      properties.put("mail.smtp.host", "relay.jangosmtp.net");
```

```
      properties.put("mail.smtp.port", "25");
      Session session = Session.getDefaultInstance(properties);

      // session.setDebug(true);
      try
      {
         // Get a Store object and connect to the current host
         Store store = session.getStore("pop3s");
         store.connect("pop.gmail.com", "xyz@gmail.com",
            "******");//change the user and password accordingly

         Folder folder = store.getFolder("inbox");
         if (!folder.exists()) {
            System.out.println("inbox not found");
               System.exit(0);
         }
         folder.open(Folder.READ_ONLY);

         BufferedReader reader = new BufferedReader(new InputStreamReader(
            System.in));

         Message[] messages = folder.getMessages();
         if (messages.length != 0) {

            for (int i = 0, n = messages.length; i < n; i++) {
               Message message = messages[i];
               date = message.getSentDate();
               // Get all the information from the message
               String from = InternetAddress.toString(message.getFrom());
               if (from != null) {
                  System.out.println("From: " + from);
               }
               String replyTo = InternetAddress.toString(message
          .getReplyTo());
               if (replyTo != null) {
                  System.out.println("Reply-to: " + replyTo);
               }
               String to = InternetAddress.toString(message
          .getRecipients(Message.RecipientType.TO));
               if (to != null) {
                  System.out.println("To: " + to);
               }

               String subject = message.getSubject();
               if (subject != null) {
                  System.out.println("Subject: " + subject);
               }
               Date sent = message.getSentDate();
               if (sent != null) {
                  System.out.println("Sent: " + sent);
               }

               System.out.print("Do you want to reply [y/n] : ");
               String ans = reader.readLine();
               if ("Y".equals(ans) || "y".equals(ans)) {

                  Message replyMessage = new MimeMessage(session);
                  replyMessage = (MimeMessage) message.reply(false);
                  replyMessage.setFrom(new InternetAddress(to));
                  replyMessage.setText("Thanks");
                  replyMessage.setReplyTo(message.getReplyTo());

                  // Send the message by authenticating the SMTP server
                  // Create a Transport instance and call the sendMessage
                  Transport t = session.getTransport("smtp");
                  try {
         //connect to the smpt server using transport instance
       //change the user and password accordingly
               t.connect("abc", "*****");
```

```
                t.sendMessage(replyMessage,
                        replyMessage.getAllRecipients());
                    } finally {
                       t.close();
                    }
                    System.out.println("message replied successfully ....");

                    // close the store and folder objects
                    folder.close(false);
                    store.close();

                 } else if ("n".equals(ans)) {
                    break;
                 }
              }//end of for loop

           } else {
              System.out.println("There is no msg....");
           }

        } catch (Exception e) {
           e.printStackTrace();
        }

    }

}
```

> *You can set the debug on by uncommenting the statement session.setDebug*true*;*

Compile and Run

Now that our class is ready, let us compile the above class. I've saved the class ReplyToEmail.java to directory : **/home/manisha/JavaMailAPIExercise**. We would need the jars *javax.mail.jar* and *activation.jar* in the classpath. Execute the command below to compile the class *boththejarsareplacedin/home/manisha/directory* from command prompt:

```
javac -cp /home/manisha/activation.jar:/home/manisha/javax.mail.jar: ReplyToEmail.java
```

Now that the class is compiled, execute the following command to run:

```
java -cp /home/manisha/activation.jar:/home/manisha/javax.mail.jar: ReplyToEmail
```

Verify Output

You should see the following message on the command console:

```
From: ABC <abc@gmail.com>
Reply-to: abc@trioteksolutions.com
To: XYZ <xyz@gmail.com>
Subject: Hi today is a nice day
Sent: Thu Oct 17 15:58:37 IST 2013
Do you want to reply [y/n] : y
message replied successfully ....
```

Check the inbox to which the mail was sent. In our case the message received looks as below: