

JAVAMAIL API - AUTHENTICATION

http://www.tutorialspoint.com/javamail_api/javamail_api_authentication.htm

Copyright © tutorialspoint.com

In the previous chapters [Checking Emails](#) and [Fetching Emails](#), we passed authorization credentials *useradpassword* along with host, when connecting to store of your mailbox. Instead we can configure the *Properties* to have the host, and tell the Session about your custom Authenticator instance. This is shown in the example below:

Create Java Class

We will modify our CheckingMails.java from the chapter [Checking Emails](#). Its contents are as below:

```
package com.tutorialspoint;

import java.util.Properties;

import javax.mail.Authenticator;
import javax.mail.Folder;
import javax.mail.Message;
import javax.mail.MessagingException;
import javax.mail.NoSuchProviderException;
import javax.mail.PasswordAuthentication;
import javax.mail.Session;
import javax.mail.Store;

public class CheckingMails {

    public static void check(String host, String storeType, String user,
        String password)
    {
        try {

            // create properties field
            Properties properties = new Properties();

            properties.put("mail.pop3s.host", host);
            properties.put("mail.pop3s.port", "995");
            properties.put("mail.pop3s.starttls.enable", "true");

            // Setup authentication, get session
            Session emailSession = Session.getInstance(properties,
                new javax.mail.Authenticator() {
                    protected PasswordAuthentication getPasswordAuthentication() {
                        return new PasswordAuthentication(
                            "manisha@gmail.com", "manisha123");
                    }
                });
            // emailSession.setDebug(true);

            // create the POP3 store object and connect with the pop server
            Store store = emailSession.getStore("pop3s");

            store.connect();

            // create the folder object and open it
            Folder emailFolder = store.getFolder("INBOX");
            emailFolder.open(Folder.READ_ONLY);

            // retrieve the messages from the folder in an array and print it
            Message[] messages = emailFolder.getMessages();
            System.out.println("messages.length---" + messages.length);

            for (int i = 0, n = messages.length; i < n; i++) {
                Message message = messages[i];
                System.out.println("-----");
            }
        }
    }
}
```

```

        System.out.println("Email Number " + (i + 1));
        System.out.println("Subject: " + message.getSubject());
        System.out.println("From: " + message.getFrom()[0]);
        System.out.println("Text: " + message.getContent().toString());
    }

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

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

public static void main(String[] args) {

    String host = "pop.gmail.com";// change accordingly
    String mailStoreType = "pop3";
    String username = "abc@gmail.com";// change accordingly
    String password = "*****";// change accordingly

    check(host, mailStoreType, username, password);

}
}

```

You can set the debug on by uncommenting the statement `emailSession.setDebugtrue` ;

Compile and Run

Now that our class is ready, let us compile the above class. I've saved the class `CheckingMails.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
both the jars are placed in /home/manisha/directory from command prompt:

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

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

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

Verify Output

You can see a similar message as below on the command console:

```

messages.length---3
-----
Email Number 1
Subject: Today is a nice day
From: XYZ <xyz@gmail.com>
Text: javax.mail.internet.MimeMultipart@45f676cb
-----
Email Number 2
Subject: hiii....
From: XYZ <xyz@gmail.com>
Text: javax.mail.internet.MimeMultipart@37f12d4f
-----

```

Email Number 3
Subject: helloo
From: XYZ <xyz@gmail.com>
Text: iavax.mail.internet.MimeMultipart@3ad5ba3a

Loading [Mathjax]/jax/output/HTML-CSS/fonts/TeX/fontdata.js