SENDING EMAIL USING RUBY - SMTP

Simple Mail Transfer Protocol *SMTP* is a protocol, which handles sending e-mail and routing e-mail between mail servers.

Ruby provides Net::SMTP class for Simple Mail Transfer Protocol *SMTP* client-side connection and provides two class methods *new* and *start*.

- The **new** takes two parameters:
  - The *server name* defaulting to localhost
  - The *port number* defaulting to the well-known port 25
- The **start** method takes these parameters:
  - The *server* - IP name of the SMTP server, defaulting to localhost
  - The *port* - Port number, defaulting to 25
  - The *domain* - Domain of the mail sender, defaulting to ENV["HOSTNAME"]
  - The *account* - Username, default is nil
  - The *password* - User password, defaulting to nil
  - The *authtype* - Authorization type, defaulting to *cram_md5*

An SMTP object has an instance method called sendmail, which will typically be used to do the work of mailing a message. It takes three parameters:

- The *source* - A string or array or anything with an each iterator returning one string at a time.
- The *sender* - A string that will appear in the *from* field of the email.
- The *recipients* - A string or an array of strings representing the recipients' addressee*s*.

Example:

Here is a simple way to send one email using Ruby script. Try it once:

```
require 'net/smtp'

message = <<MESSAGE_END
From: Private Person <me@fromdomain.com>
To: A Test User <test@todomain.com>
Subject: SMTP e-mail test

This is a test e-mail message.
MESSAGE_END

Net::SMTP.start('localhost') do |smtp|
  smtp.send_message message, 'me@fromdomain.com',
                             'test@todomain.com'
end
```

Here, you have placed a basic e-mail in message, using a here document, taking care to format the headers correctly. An e-mails requires a **From**, **To**, and **Subject** header, separated from the body of the e-mail with a blank line.

To send the mail you use Net::SMTP to connect to the SMTP server on the local machine and then use the send_message method along with the message, the from address, and the destination address as parameters *eventhoughthefromandtoaddressesarewithinthee − mailitself, thesearen't alwaysusedtoroutemail*.

If you're not running an SMTP server on your machine, you can use Net::SMTP to communicate with a remote SMTP server. Unless you're using a webmail service *suchasHotmailorYahoo!Mail*, your e-mail provider will have provided you with outgoing mail server details that you can supply to Net::SMTP, as follows:

```
Net::SMTP.start('mail.your-domain.com')
```

This line of code connects to the SMTP server on port 25 of mail.your-domain.com without using any username or password. If you need to, though, you can specify port number and other details. For example:

```
Net::SMTP.start('mail.your-domain.com',
                25,
                'localhost',
                'username', 'password' :plain)
```

This example connects to the SMTP server at mail.your-domain.com using a username and password in plain text format. It identifies the client's hostname as localhost.

Sending an HTML e-mail using Ruby:

When you send a text message using Ruby then all the content will be treated as simple text. Even if you will include HTML tags in a text message, it will be displayed as simple text and HTML tags will not be formatted according to HTML syntax. But Ruby Net::SMTP provides option to send an HTML message as actual HTML message.

While sending an email message you can specify a Mime version, content type and character set to send an HTML email.

Example:

Following is the example to send HTML content as an email. Try it once:

```
require 'net/smtp'

message = <<MESSAGE_END
From: Private Person <me@fromdomain.com>
To: A Test User <test@todomain.com>
MIME-Version: 1.0
Content-type: text/html
Subject: SMTP e-mail test

This is an e-mail message to be sent in HTML format

<b>This is HTML message.</b>
<h1>This is headline.</h1>
MESSAGE_END

Net::SMTP.start('localhost') do |smtp|
  smtp.send_message message, 'me@fromdomain.com',
                             'test@todomain.com'
end
```

Sending Attachments as an e-mail:

To send an email with mixed content requires to set **Content-type** header to **multipart/mixed**. Then text and attachment sections can be specified within **boundaries**.

A boundary is started with two hyphens followed by a unique number which can not appear in the message part of the email. A final boundary denoting the email's final section must also end with two hyphens.

Attached files should be encoded with the **pack** " ***m*** " function to have base64 encoding before transmission.

Example:

Following is the example which will send a file **/tmp/test.txt** as an attachment. Try it once:

```
require 'net/smtp'

filename = "/tmp/test.txt"
# Read a file and encode it into base64 format
filecontent = File.read(filename)
encodedcontent = [filecontent].pack("m")   # base64

marker = "AUNIQUEMARKER"

body =<<EOF
This is a test email to send an attachement.
EOF

# Define the main headers.
part1 =<<EOF
From: Private Person <me@fromdomain.net>
To: A Test User <test@todmain.com>
Subject: Sending Attachement
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary=#{marker}
--#{marker}
EOF

# Define the message action
part2 =<<EOF
Content-Type: text/plain
Content-Transfer-Encoding:8bit

#{body}
--#{marker}
EOF

# Define the attachment section
part3 =<<EOF
Content-Type: multipart/mixed; name=\"#{filename}\"
Content-Transfer-Encoding:base64
Content-Disposition: attachment; filename="#{filename}"

#{encodedcontent}
--#{marker}--
EOF

mailtext = part1 + part2 + part3

# Let's put our code in safe area
begin
  Net::SMTP.start('localhost') do |smtp|
     smtp.sendmail(mailtext, 'me@fromdomain.net',
                           ['test@todmain.com'])
  end
rescue Exception => e
  print "Exception occured: " + e
end
```

NOTE: You can specify multiple destinations inside the array but they should be separated by comma