

24.4.0 Komponente gb.net.smtp

Die Komponente ermöglicht es Ihnen, E-Mails mit Hilfe des SMTP-Protokolls zu senden. Unterstützt werden u.a. E-Mail-Anhänge und optional Verbindungssicherheit über SSL oder TLS, sofern das Programm 'openssl' auf Ihrem System installiert ist. Diese Komponente gb.net.smtp ist seit der Version 3.6 in allen Gambas3-Versionen vorhanden und wurde komplett in Gambas geschrieben.

24.4.0.1 Eigenschaften

Die Klasse *SmtplibClient* hat folgende Eigenschaften:

Eigenschaft	Datentyp	Beschreibung
Count	Integer	Gibt die Anzahl der Teile der aktuellen E-Mail zurück.
MessageID	String	Setzt die Message-ID in das Feld 'Message-Id' im Header oder gibt die ID zurück.
InReplyTo	String	Setzt die E-Mail-Adresse in das Feld 'In-Reply-To' im Header oder gibt die E-Mail-Adresse zurück.
Debug	Boolean	Gibt an, ob der Debugging-Modus für den SMTP-Client aktiviert wurde oder schaltet den Modus zu. Wenn der Debugging-Modus aktiviert ist, wird die gesamte Client-Aktivität in der Konsole der Gambas-IDE ausgegeben.
Alternative	Boolean	Ermittelt oder legt fest, ob Mail-Anhänge gleiche Daten in unterschiedlichen Formaten sind – kodiert in HTML, UTF-8 oder in ASCII. Standardmäßig ist diese Eigenschaft FALSE. Eine Änderung müssen Sie vor der Add-Methode definieren.
From	String	Ermittelt oder legt die Return-Adresse der E-Mail fest (Absender).
To	String[]	Ermittelt oder legt die Liste der Empfänger der E-Mail (E-Mail-Adressen) fest. Der Typ der Eigenschaft ist ein String-Array, so dass Sie mehrere Empfänger für die gleiche E-Mail angeben können.
Cc	String[]	Ermittelt oder legt die Liste der zusätzlichen (Cc-)Empfänger (carbon copy) der E-Mail (E-Mail-Adressen) fest. Der Typ der Eigenschaft ist ein String-Array, so dass Sie mehrere zusätzliche Empfänger für die gleiche E-Mail angeben können.
Bcc	String[]	Ermittelt oder legt die Liste der zusätzlichen (Bcc-)Empfänger (blind carbon copy) der E-Mail (E-Mail-Adressen) fest. Der Typ der Eigenschaft ist ein String-Array, so dass Sie mehrere zusätzliche, versteckte Empfänger für die gleiche E-Mail angeben können.
Host	String	Ermittelt oder legt den SMTP-Server fest, an den die E-Mail übergeben wird (MTA).
Port	Integer	Ermittelt oder legt den SMTP-Port fest, der vom SMTP-Server verwendet wird. Der Standard-Port ist 25.
User	String	Ermittelt den Benutzer-Namen oder legt diesen für die Identifizierung fest (Login).
Password	String	Ermittelt oder legt das Passwort fest, das zur Authentifizierung erforderlich ist (Login).
Subject	String	Ermittelt oder legt den Betreff der E-Mail fest.
Body	String	Ermittelt oder legt den E-Mail-Text fest. Der Mail-Text wird immer als erster Teil gesendet. Es wird der Mime-Typ 'text/plain; charset=utf-8' verwendet.
Encrypt	Integer	Ermittelt oder legt die Methode der SMTP-Verbindungssicherheit fest.

Tabelle 24.4.0.1.1 : Eigenschaften der Klasse SmtplibClient

Die Konstanten der Klasse SmtplibClient in Verbindung mit der Encrypt-Eigenschaft sind:

- Net.None (0) Ohne Verbindungssicherheit
- Net.SSL (1) SSL-Verbindungssicherheit
- Net.TLS (2) TLS-Verbindungssicherheit

24.4.0.2 Methoden

Die Klasse *SmtplibClient* verfügt nur über drei Methoden:

Methode	Beschreibung
Add (Data As String [, Mime-Type As String, Name As String])	Fügt einen EMail-Anhang hinzu. <i>Data</i> sind die Daten, die in der Regel direkt aus der Datei kommen, die Sie anhängen möchten. <i>Mime-Type</i> ist der MIME-Typ der angehängten Daten. Standardmäßig ist es 'text/plain'. <i>Name</i> ist der Name des Anhangs. Er wird automatisch generiert, wenn er nicht angegeben wurde. Gegenwärtig werden nicht alle MIME-Typen unterstützt.
AddHeader (Name As String, Value As String)	Fügt ein benutzer-definiertes Header-Feld in den Header ein. <i>Name</i> ist der Feld-Name und <i>Value</i> ist der Feld-Wert.
Send()	Die EMail wird an den SMTP-Server (MTA) übergeben. Wenn die Verbindung verschlüsselt wird, dann wird das Programm 'openssl' im Hintergrund eingesetzt. Der openssl-Prozess wird beendet, wenn die EMail erfolgreich gesendet wurde oder wenn ein Fehler beim Senden auftrat.

Tabelle 24.4.0.2.1 : Methoden der Klasse SmtplibClient

Die statische Methode *FormatDate (Date As Date) As String* formatiert ein Datum nach der Syntax der RFC 822: `SmtplibClient.FormatDate(Now) → Mon, 01 Feb 2016 18:12:36 GMT` .

24.4.0.3 Authentifizierung über CRAM-MD5

Wenn ein SMTP-Server neben den Verfahren AUTH PLAIN und AUTH LOGIN auch die Authentifizierung über das Verfahren CRAM-MD5 (Challenge Response Authentication Mechanism, Message Digest 5) zulässt, dann wird der SMTP-Username zur Identifizierung des Users im Klartext im SSL-Tunnel gesendet, aber das SMTP-Passwort wird verschlüsselt übertragen:

- Der SMTP-Client verbindet sich zum SMTP-Server. Die SSL-Verbindung wird in → Zeile 3 mit dem Status-Code 220 und einer Statusmeldung bestätigt.
- Der Client grüßt mit EHLO und dem lokalen Hostnamen → Zeile 4.
- Der Server antwortet mit einer Übersicht möglicher Kommandos für weitere Transaktionen in den → Zeilen 5 bis 11.
- Der Client entscheidet sich im Beispiel für die Identifizierung und Authentifizierung über CRAM-MD5 und sendet das Kommando *AUTH CRAM-MD5* an den SMTP-Server → Zeile 12.
- Der SMTP-Server antwortet in der → Zeile 13 nach dem Status-Code 334 mit einer base64-kodierten Zeichenkette – der so genannten Challenge – aus Zahlen, Zeitstempel und voll qualifiziertem Hostnamen des Servers.
- Der SMTP-Client antwortet mit einer Zeichenkette (Response) aus User-Name, Leerzeichen und einem 'Digest'. Dieses Digest ist der base64-kodierte MD5-Hash eines Wertes – berechnet aus der Challenge und dem Passwort → Zeile 14.
- Der SMTP-Server prüft das erhaltene Digest, indem er die selbe Berechnung wie der SMTP-Client mit auf dem Server hinterlegtem User-Namen und User-Passwort durchführt und sein Ergebnis mit dem vom Client erhaltenen vergleicht. Stimmen beide Werte überein, ist der identifizierte User auch authentifiziert. Das wird in der → Zeile 15 nach dem Status-Code 235 bestätigt.

```
[1] hans@linux:~$ openssl s_client -quiet -connect mx.freenet.de:465
[2] ...
[3] 220 mx8.freenet.de ESMTP Exim 4.85 Tue, 02 Feb 2016 07:53:59 +0100
[4] EHLO linux
[5] 250-mx8.freenet.de Hello p4fca44f0.dip0.t-ipconnect.de [79.202.68.240]
[6] 250-SIZE 209715200
[7] 250-8BITMIME
[8] 250-ETRN
[9] 250-PIPELINING
[10] 250-AUTH LOGIN PLAIN CRAM-MD5
[11] 250 HELP
[12] AUTH CRAM-MD5
[13] 334 PDE4OTYuNjk3MTcwOTUyQHBvc3RvZmZpY2UucmVzdG9uLm1jaS5uZXQ+
[14] dGltIGI5MTNhNjAyYzdlZGE3YTQ5NWl0ZTZlNzZmZNGQzODkw
[15] 235 Authentication succeeded
[16] QUIT
[17] 221 mx8.freenet.de closing connection
[18] hans@linux:~$
```

Für die Berechnung der zu sendenden Antwort in der Zeile 14 können Sie die folgende Funktion verwenden. Als Parameter werden der SMTP-User-Name, das dazu gehörende SMTP-User-Passwort und die vom Server gestellte 'Aufgabe' übergeben:

```
[1] Public Function CRAMMD5(sUsername As String, sPassword As String, sChallenge64 As String) As String
[2]
[3]     Dim sChallenge, sCommand, sDigestHex, sResponse As String
[4]
[5]     sChallenge = UnBase64(sChallenge64)
[6]
[7]     sCommand = "openssl md5 -hmac " & Shell$(sPassword) & " << EOF\n" & sChallenge
[8]     Shell sCommand To sDigestHex
[9]
[10]    sDigestHex = Split(Trim(sDigestHex), "=")[1]
[11]    sResponse = Base64(sUsername & sDigestHex)
[12]
[13]    Return sResponse
[14]
[15] End ' Function CRAMMD5(...)
```

Hinweise:

- Der Funktionswert in der → Zeile 13 ist die Antwort, die der Server erwartet.
- Seit der Gambas-Version 3.8.4 kann auch das CRAM-MD5-Verfahren eingesetzt werden.
- Der SMTP-Client im Kapitel 24.4.1 setzt das CRAM-MD5-Verfahren alternativ und frei wählbar ein.

24.4.0.4 Exkurs SMTP

Der folgende Exkurs fasst wesentliche, ausgewählte Aspekte zum Thema SMTP zusammen:

- SMTP steht für Simple Mail Transfer Protokoll.
- Das EMail-Transport-System enthält als Element u.a. den Dienst MUA (*Mail-User-Agent*), der die Übergabe einer EMail an den Dienst MTA (*Mail-Transfer-Agent*) leistet, wobei ein MTA für die Weiterleitung und Zustellung an einen Nachrichten-Speicher eingesetzt wird. Den Fernzugriff auf den Nachrichten-Speicher regelt vor allem das Post-Office-Protokoll (POP) – gegenwärtig in der Version 3.
- SMTP regelt die Übergabe einer EMail an einen Server als MTA.
- Wie für alle Internet-Protokolle gibt es die verbindlichen Regeln zu SMTP in unterschiedlichen Beschreibungen – den RFC.
- SMTP wird in den RFC 821 und seit 2001 in den RFC 2821 beschrieben, wobei die RFC 2821 Erweiterungen der seit 1982 geltenden RFC 821 enthält.
- SMTP ist ein Client-Server-Protokoll und setzt eine aktive TCP/IP-Verbindung zwischen Client und Server voraus, die der Client aufbaut.
- SMTP ist ein text-basiertes Protokoll.
- SMTP initiiert Transaktionen zwischen Client und Server durch das Senden von Kommandos durch den Client.
- Nach jedem Kommando wartet der Client eine endliche Zeit auf die Server-Antwort (Timeout).
- Eine Server-Antwort beginnt mit einem Status-Code (Zahl aus drei Ziffern), dem eine (nicht normierte) Status-Meldung folgt.
- SMTP setzt auf das Transport-Protokoll auf (Standard). Der Standard-Port ist 25. Bei einer verschlüsselten Verbindung zwischen Client und Server wird bei SSL auf den Port 465 gewechselt und bei TLS auf 587.