**汇总相关资料**

Libcurl ===========》数据传输调用库

SMTP协议 & MIME协议 ===========》邮件发送协议

POP3协议 & IMAP协议 ===========》邮件接收协议

Telnet ===========》测试邮箱服务器工具

smtp://mail.dev.appeon.net:587 ===========》内部邮箱服务器地址&端口

smtp://smtp.gmail.com:587 ===========》Google邮箱服务器地址&端口

smtp://smtp-mail.outlook.com:587 ===========》outlook邮箱服务器地址&端口

==========================================================================================================================

**邮箱服务器搭建：**（windows 10系统搭建服务器比较麻烦，因此使用公司邮箱服务器进行测试）

通常，Windows 10、Windows 7 等客户端操作系统不包含 SMTP 服务器功能。您可以使用 IIS 中的SMTP 电子邮件功能连接到 Windows Server 上的现有 SMTP 服务器。SMTP 电子邮件功能不是 SQL Server 数据库邮件所必需的 SMTP 服务器。因此，本主题不包括在客户端操作系统上安装和配置 SMTP 服务器的步骤。

MSDN：

<https://docs.microsoft.com/en-us/biztalk/install-and-config-guides/appendix-d-create-the-smtp-server>

==========================================================================================================================

**测试邮箱Exchange服务器方式：**(先安装Telnet)

通过Telnet进行邮箱服务器测试。

MSDN：

<https://docs.microsoft.com/en-us/exchange/mail-flow/test-smtp-with-telnet?view=exchserver-2019>

相关数据：

目标SMTP邮箱服务器：mail.dev.appeon.net:25

域名domain：dev.appeon.net

发件人：[zenzhaofa@dev.appeon.net](mailto:zenzhaofa@dev.appeon.net)

收件人：@dev.appeon.net

步骤：

1. cmd---------》telnet
2. set localecho
3. set logfile SMTPlog
4. open mail.dev.appeon.net:25
5. EHLO dev.appeon.net
6. mail from:zengzhaofa@dev.appeon.net
7. rcpt to:zengzhaofa@dev.appeon.net 🡪enter
8. data ==🡺enter
9. subject: test=🡺enter =🡺enter
10. use telnet test SMTP from cmd =🡺enter
11. .=🡺enter
12. quit

==========================================================================================================================

**发送邮件的三种方式：**

1. **加密发送邮件（SMTP+SSL/TLS）**
2. **不加密发送邮件（SMTP）**
3. **发送邮件带附件（MIME+SSL/TLS）**

==========================================================================================================================

**发送邮件的流程：**

1. **建立连接**
2. **发送邮件**
3. **断开连接**
4. 建立连接

当Client编辑好邮件后，点击发送的过程中，Client会和邮箱服务器建立连接。在进行连接过程中可以进行（SSL/TLS）协议加密，也可以进行邮箱服务器的证书认证。证书认证可以使用本地证书/自带证书文件。

1. 发送邮件

发送邮件过程中，不是一次性将数据传输到服务器，而是响应应答的方式。（具体看案例分析）。

1. 断开连接

断开连接过程需要释放libcurl资源。

**案例分析（Client和Server 应答流程）**

== Info: Trying 74.125.192.109:587...

== Info: Connected to smtp.gmail.com (74.125.192.109) port 587 (#0)

<= Recv header, 0000000054 bytes (0x00000036)

0000: 220 smtp.gmail.com ESMTP x16sm18277003qko.17 - gsmtp

=> Send header, 0000000022 bytes (0x00000016)

0000: EHLO developertestse

<= Recv header, 0000000052 bytes (0x00000034)

0000: 250-smtp.gmail.com at your service, [20.84.124.50]

<= Recv header, 0000000019 bytes (0x00000013)

0000: 250-SIZE 35882577

<= Recv header, 0000000014 bytes (0x0000000e)

0000: 250-8BITMIME

<= Recv header, 0000000014 bytes (0x0000000e)

0000: 250-STARTTLS

<= Recv header, 0000000025 bytes (0x00000019)

0000: 250-ENHANCEDSTATUSCODES

<= Recv header, 0000000016 bytes (0x00000010)

0000: 250-PIPELINING

<= Recv header, 0000000014 bytes (0x0000000e)

0000: 250-CHUNKING

<= Recv header, 0000000014 bytes (0x0000000e)

0000: 250 SMTPUTF8

=> Send header, 0000000010 bytes (0x0000000a)

0000: STARTTLS

<= Recv header, 0000000030 bytes (0x0000001e)

0000: 220 2.0.0 Ready to start TLS

== Info: TLSv1.0 (OUT), TLS header, Certificate Status (22):

=> Send SSL data, 0000000005 bytes (0x00000005)

0000: .....

== Info: TLSv1.3 (OUT), TLS handshake, Client hello (1):

=> Send SSL data, 0000000512 bytes (0x00000200)

0000: .......$r.F:l....Y..]..gDL..OyB.<.'.5. .\_..C^.S.... .=....0}/...

0040: ..,|.a..>.......,.0.........+./...$.(.k.#.'.g.....9.....3.....=.

0080: <.5./.....u.........smtp.gmail.com..............................

00c0: .............1.....\*.(.........................................+

0100: ............-.....3.&.$... }s.........\_.........K%.l..p..#W.....

0140: ................................................................

0180: ................................................................

01c0: ................................................................

== Info: TLSv1.2 (IN), TLS header, Certificate Status (22):

<= Recv SSL data, 0000000005 bytes (0x00000005)

0000: ....z

== Info: TLSv1.3 (IN), TLS handshake, Server hello (2):

<= Recv SSL data, 0000000122 bytes (0x0000007a)

0000: ...v...nA....`...v..oH...KD.y..[D..qY. .\_..C^.S.... .=....0}/...

0040: ..,|.a.......3.$... G$..z..b/..q........F..:n\o..x.^.+....

== Info: TLSv1.2 (IN), TLS header, Finished (20):

<= Recv SSL data, 0000000005 bytes (0x00000005)

0000: .....

== Info: TLSv1.2 (IN), TLS header, Supplemental data (23):

<= Recv SSL data, 0000000005 bytes (0x00000005)

0000: ....;

<= Recv SSL data, 0000000001 bytes (0x00000001)

0000: .

== Info: TLSv1.3 (IN), TLS handshake, Encrypted Extensions (8):

<= Recv SSL data, 0000000006 bytes (0x00000006)

0000: ......

== Info: TLSv1.3 (IN), TLS handshake, Certificate (11):

<= Recv SSL data, 0000004002 bytes (0x00000fa2)

0000: ...........0...0..o.......j...f+.3...../.|0...\*.H........0F1.0..

0040: .U....US1"0 ..U....Google Trust Services LLC1.0...U....GTS CA 1C

0080: 30...220117030736Z..220411030735Z0.1.0...U....smtp.gmail.com0Y0.

00c0: ..\*.H.=....\*.H.=....B...T.$.]zA..].3O7\_"b1....!...x.ai.:....-I.>

0100: ....?]...h\_a..n..+...S....g0..c0...U...........0...U.%..0...+...

0140: ....0...U.......0.0...U......0....C\...b@....9...0...U.#..0....t

0180: ......=...F..q5.'0j..+........^0\0'..+.....0...http://ocsp.pki.

01c0: goog/gts1c301..+.....0..%http://pki.goog/repo/certs/gts1c3.der0.

0200: ..U....0...smtp.gmail.com0!..U. ..0.0...g.....0...+.....y...0<..

0240: U...50301./.-.+http://crls.pki.goog/gts1c3/moVDfISia2k.crl0.....

0280: +.....y............v.Q.....y.Vm.7x...z..'....B..........~f8.@...

02c0: ..G0E.!..I|7..J.rk.n..^.......M...>...... H|.Y......".....V....

0300: I...y....].v.F.U.u.. 0...i..}.,At..I.....p.mG...~f8.h.....G0E. .

0340: X....|. .....c.......p....u..|..!..$L.5.........q.......i.I.w.@1

0380: Y.0...\*.H.............$....}.I{S....9...4j.Sv...@.P.k..}.M..{1L.

03c0: .J..ZMZ..[..~|H.....UB.x..g.D"..F...{Wd.0..7.n.(c9.#Y...Re.8.].D

0400: ..m..]\.-,...Z......V...Fl)g..@..!.Otl.U...R...sv....M..UJ@....Q

0440: .G.v..Gp.()..Yx..w..Z...7O.;,..../f....)-...T1y..Nlx(.j...0.}+V.

0480: .3-(.i.......B.$....C......0...0..~..........SYk4....Pf0...\*.H..

04c0: ......0G1.0...U....US1"0 ..U....Google Trust Services LLC1.0...U

0500: ....GTS Root R10...200813000042Z..270930000042Z0F1.0...U....US1"

0540: 0 ..U....Google Trust Services LLC1.0...U....GTS CA 1C30.."0...\*

0580: .H.............0............b..7.7B.l...e.%...k..m.Z#.........|

05c0: ....B.^V$.z3....i..t.WLfh.w7US.9.M.4.\_%w7;...<......C...G..D.c..

0600: A..A0H......E.!..B...+eV4.&....}....H|7M?.....u..yW\.Wn.........

0640: %..,...\*....c.<PI....\_.+Y.....Q..w....O.pI.\m .......w.-...k...

0680: .+........'....Q.................0..|0...U...........0...U.%..0.

06c0: ..+.........+.......0...U.......0.......0...U.......t......=...

0700: F..q5.'0...U.#..0.....+&q.+H'./Rf,....q>0h..+........\0Z0&..+...

0740: ..0...http://ocsp.pki.goog/gtsr100..+.....0..$http://pki.goog/re

0780: po/certs/gtsr1.der04..U...-0+0).'.%.#http://crl.pki.goog/gtsr1/g

07c0: tsr1.crl0W..U. .P0N08..+.....y...0\*0(..+.........https://pki.goo

0800: g/repository/0...g.....0...g.....0...\*.H..............}. \.<...W

0840: .......rq.6...@..L.F...$..Pq"...n..jo......\_.l.......b....[.f..

0880: .......i>z.FI\_F.A...Me4...?O.lI..SA..!.....D[\*P..M.S6.B..T..wS.

08c0: d8'...X..|9-[..........S$....y.&.a.SR.B..f+?...........q.5($....

0900: .-.H.=Y.Q.t..|...[..4..........."....q....s$.7S...?..\.6..;.)...

0940: :b;lc...Yq.c'.L....s..\*....l2.3...Qq.4...].QX......Y.q..M(..m...

0980: ...F...k.w.....#.........D..u#.4.. ..^...RF.....!pQ.....U.+.3w.K

09c0: B..w..s.....7?..\*f.s.2.2l2....#.[}Mep.+.=...m.2.....c...]...q^\*.

0a00: .."..e:...e.....[.Y.G.-.$:...&....7..o....Q......Q......f0..b0.

0a40: .J.......w..l.6...!...X..0...\*.H........0W1.0...U....BE1.0...U..

0a80: ..GlobalSign nv-sa1.0...U....Root CA1.0...U....GlobalSign Root C

0ac0: A0...200619000042Z..280128000042Z0G1.0...U....US1"0 ..U....Googl

0b00: e Trust Services LLC1.0...U....GTS Root R10.."0...\*.H...........

0b40: ..0...............w.;...>...@<....}2..q.......j.....K.+........

0b80: ...........^..R..#'....c...~..^.h...ZG.M.3.N.....lK......d)%#...

0bc0: .=.`.......H.M..z.....Y........1.......ml....~&.E.=.y..(..&....

0c00: ..<h.S..:.+.....z..u....Vd..Oh.=......@..\....5l..P..L... .3.R.

0c40: .2.).%\*.H.r..d...........8f..c...x.{\w.v......y.W..&............

0c80: ..U.....K)...2%N\*.eD.....I...|..@{.C..l..}...L......K.....E.v..@

0cc0: +.S....;......1..w.o{>...".....2..c.Qr.]....)h3.:f...&...Wex'.^I

0d00: ......!............lH<@.~.Z.V<.....K.9K..?.U.n$..q..........A..

0d40: .=:..z.7...........80..40...U...........0...U.......0....0...U..

0d80: ......+&q.+H'./Rf,....q>0...U.#..0...`{f.E....P/}..4....K0`..+..

0dc0: ......T0R0%..+.....0...http://ocsp.pki.goog/gsr10)..+.....0...ht

0e00: tp://pki.goog/gsr1/gsr1.crt02..U...+0)0'.%.#.!http://crl.pki.goo

0e40: g/gsr1/gsr1.crl0;..U. .4020...g.....0...g.....0...+.....y....0..

0e80: .+.....y....0...\*.H.............4...(...v..1z!..R>..t.A..=5.....

0ec0: .\\_...|......W.&o[..Fh.7okz...7.%Q..h...I.Z...#...+.....Ij.u....

0f00: ...XHW.5.....o..o.......\*..Ni..-.h..+s....".7..f.I..U.g.2..&.p.=

0f40: .gm=|.4..2..n.jo.....K.;..7..D.~.l..F.....!.f...Ul.)...f[.wIH(..

0f80: ..3rS..5.b..$...9..~\*A.R.......?..

== Info: TLSv1.3 (IN), TLS handshake, CERT verify (15):

<= Recv SSL data, 0000000078 bytes (0x0000004e)

0000: ...J...F0D. !f...R...H.-.S..W.L.....=.z/T..i. td..e(..=....}.:`e

0040: Te...v9n.xe.V.

== Info: TLSv1.3 (IN), TLS handshake, Finished (20):

<= Recv SSL data, 0000000052 bytes (0x00000034)

0000: ...0>.......D...B..|R....x:...^.Q....e..j....\*;....~

== Info: TLSv1.2 (OUT), TLS header, Finished (20):

=> Send SSL data, 0000000005 bytes (0x00000005)

0000: .....

== Info: TLSv1.3 (OUT), TLS change cipher, Change cipher spec (1):

=> Send SSL data, 0000000001 bytes (0x00000001)

0000: .

== Info: TLSv1.2 (OUT), TLS header, Supplemental data (23):

=> Send SSL data, 0000000005 bytes (0x00000005)

0000: ....E

=> Send SSL data, 0000000001 bytes (0x00000001)

0000: .

== Info: TLSv1.3 (OUT), TLS handshake, Finished (20):

=> Send SSL data, 0000000052 bytes (0x00000034)

0000: ...0.<.b`2..x;.....F....hdR^.b.......bT.)..w...`.q."

== Info: SSL connection using TLSv1.3 / TLS\_AES\_256\_GCM\_SHA384

== Info: Server certificate:

== Info: subject: CN=smtp.gmail.com

== Info: start date: Jan 17 03:07:36 2022 GMT

== Info: expire date: Apr 11 03:07:35 2022 GMT

== Info: issuer: C=US; O=Google Trust Services LLC; CN=GTS CA 1C3

== Info: SSL certificate verify result: unable to get local issuer certificate (20), continuing anyway.

== Info: TLSv1.2 (OUT), TLS header, Supplemental data (23):

=> Send SSL data, 0000000005 bytes (0x00000005)

0000: ....'

=> Send SSL data, 0000000001 bytes (0x00000001)

0000: .

=> Send header, 0000000022 bytes (0x00000016)

0000: EHLO developertestse

== Info: TLSv1.2 (IN), TLS header, Supplemental data (23):

<= Recv SSL data, 0000000005 bytes (0x00000005)

0000: .....

<= Recv SSL data, 0000000001 bytes (0x00000001)

0000: .

== Info: TLSv1.3 (IN), TLS handshake, Newsession Ticket (4):

<= Recv SSL data, 0000000258 bytes (0x00000102)

0000: ........$R.}.....%..T~..9..9...9D.G.........+[......6..q..m0.t42

0040: .J4W..{A...eS.o.-..!Z7.....b@.9..;aD.`F..f...f......4H.....{z...

0080: .....n.N0....&./...6RA... ...U9x..V...F......|.......4Q[.q...R4.

00c0: ... .g7\...B...m.B..0k..3m.@..@..i".Q.....S ...;.%.....@.@....JJ

0100: ..

== Info: TLSv1.3 (IN), TLS handshake, Newsession Ticket (4):

<= Recv SSL data, 0000000258 bytes (0x00000102)

0000: ........n..2.....%..T~..9..9...9....L[.....x.......(...-...K..{$

0040: b./....K.....V.."+@X"..j$K..U...eu.m..Rg........\....f#.X9@F.4.

0080: g.S1..K..............%=p1.W..,..B}.ZF....I../.k.%.\_~.......J.}.

00c0: ..i.....B......X.2./<#.#.........eQ]!..}#....}.z`k=J...X`.....JJ

0100: ..

== Info: old SSL session ID is stale, removing

== Info: TLSv1.2 (IN), TLS header, Supplemental data (23):

<= Recv SSL data, 0000000005 bytes (0x00000005)

0000: .....

<= Recv SSL data, 0000000001 bytes (0x00000001)

0000: .

<= Recv header, 0000000052 bytes (0x00000034)

0000: 250-smtp.gmail.com at your service, [20.84.124.50]

<= Recv header, 0000000019 bytes (0x00000013)

0000: 250-SIZE 35882577

<= Recv header, 0000000014 bytes (0x0000000e)

0000: 250-8BITMIME

<= Recv header, 0000000067 bytes (0x00000043)

0000: 250-AUTH LOGIN PLAIN XOAUTH2 PLAIN-CLIENTTOKEN OAUTHBEARER XOAUT

0040: H

<= Recv header, 0000000025 bytes (0x00000019)

0000: 250-ENHANCEDSTATUSCODES

<= Recv header, 0000000016 bytes (0x00000010)

0000: 250-PIPELINING

<= Recv header, 0000000014 bytes (0x0000000e)

0000: 250-CHUNKING

<= Recv header, 0000000014 bytes (0x0000000e)

0000: 250 SMTPUTF8

== Info: TLSv1.2 (OUT), TLS header, Supplemental data (23):

=> Send SSL data, 0000000005 bytes (0x00000005)

0000: .....

=> Send SSL data, 0000000001 bytes (0x00000001)

0000: .

=> Send header, 0000000012 bytes (0x0000000c)

0000: AUTH PLAIN

== Info: TLSv1.2 (IN), TLS header, Supplemental data (23):

<= Recv SSL data, 0000000005 bytes (0x00000005)

0000: .....

<= Recv SSL data, 0000000001 bytes (0x00000001)

0000: .

<= Recv header, 0000000006 bytes (0x00000006)

0000: 334

== Info: TLSv1.2 (OUT), TLS header, Supplemental data (23):

=> Send SSL data, 0000000005 bytes (0x00000005)

0000: ....C

=> Send SSL data, 0000000001 bytes (0x00000001)

0000: .

=> Send header, 0000000050 bytes (0x00000032)

0000: AGlkYW9iYS56ZW5nQGdtYWlsLmNvbQAxMjM0NTZxYXouLkA=

== Info: TLSv1.2 (IN), TLS header, Supplemental data (23):

<= Recv SSL data, 0000000005 bytes (0x00000005)

0000: ....%

<= Recv SSL data, 0000000001 bytes (0x00000001)

0000: .

<= Recv header, 0000000020 bytes (0x00000014)

0000: 235 2.7.0 Accepted

== Info: TLSv1.2 (OUT), TLS header, Supplemental data (23):

=> Send SSL data, 0000000005 bytes (0x00000005)

0000: ....4

=> Send SSL data, 0000000001 bytes (0x00000001)

0000: .

=> Send header, 0000000035 bytes (0x00000023)

0000: MAIL FROM:<idaoba.zeng@gmail.com>

== Info: TLSv1.2 (IN), TLS header, Supplemental data (23):

<= Recv SSL data, 0000000005 bytes (0x00000005)

0000: ....;

<= Recv SSL data, 0000000001 bytes (0x00000001)

0000: .

<= Recv header, 0000000042 bytes (0x0000002a)

0000: 250 2.1.0 OK x16sm18277003qko.17 - gsmtp

== Info: TLSv1.2 (OUT), TLS header, Supplemental data (23):

=> Send SSL data, 0000000005 bytes (0x00000005)

0000: ..../

=> Send SSL data, 0000000001 bytes (0x00000001)

0000: .

=> Send header, 0000000030 bytes (0x0000001e)

0000: RCPT TO:<idaoba@outlook.com>

== Info: TLSv1.2 (IN), TLS header, Supplemental data (23):

<= Recv SSL data, 0000000005 bytes (0x00000005)

0000: ....;

<= Recv SSL data, 0000000001 bytes (0x00000001)

0000: .

<= Recv header, 0000000042 bytes (0x0000002a)

0000: 250 2.1.5 OK x16sm18277003qko.17 - gsmtp

== Info: TLSv1.2 (OUT), TLS header, Supplemental data (23):

=> Send SSL data, 0000000005 bytes (0x00000005)

0000: ....2

=> Send SSL data, 0000000001 bytes (0x00000001)

0000: .

=> Send header, 0000000033 bytes (0x00000021)

0000: RCPT TO:<idaoba.zeng@gmail.com>

== Info: TLSv1.2 (IN), TLS header, Supplemental data (23):

<= Recv SSL data, 0000000005 bytes (0x00000005)

0000: ....;

<= Recv SSL data, 0000000001 bytes (0x00000001)

0000: .

<= Recv header, 0000000042 bytes (0x0000002a)

0000: 250 2.1.5 OK x16sm18277003qko.17 - gsmtp

== Info: TLSv1.2 (OUT), TLS header, Supplemental data (23):

=> Send SSL data, 0000000005 bytes (0x00000005)

0000: .....

=> Send SSL data, 0000000001 bytes (0x00000001)

0000: .

=> Send header, 0000000006 bytes (0x00000006)

0000: DATA

== Info: TLSv1.2 (IN), TLS header, Supplemental data (23):

<= Recv SSL data, 0000000005 bytes (0x00000005)

0000: ....<

<= Recv SSL data, 0000000001 bytes (0x00000001)

0000: .

<= Recv header, 0000000043 bytes (0x0000002b)

0000: 354 Go ahead x16sm18277003qko.17 - gsmtp

== Info: TLSv1.2 (OUT), TLS header, Supplemental data (23):

=> Send SSL data, 0000000005 bytes (0x00000005)

0000: ....K

=> Send SSL data, 0000000001 bytes (0x00000001)

0000: .

=> Send data, 0000000314 bytes (0x0000013a)

0000: Date: Mon, 29 Nov 2010 21:54:29 +1100

0027: To: <idaoba@outlook.com>

0041: From: <idaoba.zeng@gmail.com>

0060: Message-ID: <dcd7cb36-11db-487a-9f3a-e652a9458efd@dev.appeon.net

00a0: >

00a3: Subject: SMTP example message

00c2:

00c4: The body of the message starts here.

00ea:

00ec: It could be a lot of lines, could be MIME encoded, whatever.

012a: Check RFC5322.

== Info: TLSv1.2 (OUT), TLS header, Supplemental data (23):

=> Send SSL data, 0000000005 bytes (0x00000005)

0000: .....

=> Send SSL data, 0000000001 bytes (0x00000001)

0000: .

== Info: TLSv1.2 (IN), TLS header, Supplemental data (23):

<= Recv SSL data, 0000000005 bytes (0x00000005)

0000: ....G

<= Recv SSL data, 0000000001 bytes (0x00000001)

0000: .

<= Recv header, 0000000054 bytes (0x00000036)

0000: 250 2.0.0 OK 1644977065 x16sm18277003qko.17 - gsmtp

== Info: Connection #0 to host smtp.gmail.com left intact

==========================================================================================================================

1. **普通邮件流程**

curl = curl\_easy\_init();

if(curl)

{

curl\_easy\_setopt(curl, CURLOPT\_URL, "smtp://mail.example.com");

curl\_easy\_setopt(curl, CURLOPT\_MAIL\_FROM, FROM\_ADDR);

recipients = curl\_slist\_append(recipients, TO\_ADDR);

recipients = curl\_slist\_append(recipients, CC\_ADDR);

curl\_easy\_setopt(curl, CURLOPT\_MAIL\_RCPT, recipients);

curl\_easy\_setopt(curl, CURLOPT\_READFUNCTION, payload\_source);

curl\_easy\_setopt(curl, CURLOPT\_READDATA, &upload\_ctx);

curl\_easy\_setopt(curl, CURLOPT\_UPLOAD, 1L);

/\* Send the message \*/

res = curl\_easy\_perform(curl);

/\* 释放资源 \*/

curl\_slist\_free\_all(recipients);

curl\_easy\_cleanup(curl);

}

1. **SSL加密邮件流程**

curl = curl\_easy\_init();

if(curl)

{

curl\_easy\_setopt(curl, CURLOPT\_USERNAME, "user"); // 用户名

curl\_easy\_setopt(curl, CURLOPT\_PASSWORD, "secret"); // 密码

// 邮箱服务器地址

curl\_easy\_setopt(curl, CURLOPT\_URL, "smtps://mainserver.example.net");

#ifdef SKIP\_PEER\_VERIFICATION

// 默认为认证服务器证书

curl\_easy\_setopt(curl, CURLOPT\_SSL\_VERIFYPEER, 0L);

#endif

#ifdef SKIP\_HOSTNAME\_VERIFICATION

// 默认为验证主机

curl\_easy\_setopt(curl, CURLOPT\_SSL\_VERIFYHOST, 0L);

#endif

curl\_easy\_setopt(curl, CURLOPT\_MAIL\_FROM, FROM\_MAIL);

recipients = curl\_slist\_append(recipients, TO\_MAIL);

recipients = curl\_slist\_append(recipients, CC\_MAIL);

curl\_easy\_setopt(curl, CURLOPT\_MAIL\_RCPT, recipients);

curl\_easy\_setopt(curl, CURLOPT\_READFUNCTION, payload\_source);

curl\_easy\_setopt(curl, CURLOPT\_READDATA, &upload\_ctx);

curl\_easy\_setopt(curl, CURLOPT\_UPLOAD, 1L);

curl\_easy\_setopt(curl, CURLOPT\_VERBOSE, 1L);

/\* Send the message \*/

res = curl\_easy\_perform(curl);

/\* Always cleanup \*/

curl\_slist\_free\_all(recipients);

curl\_easy\_cleanup(curl);

}

1. **TLS加密邮件流程**

curl = curl\_easy\_init();

if(curl)

{

/\* Set username and password \*/

curl\_easy\_setopt(curl, CURLOPT\_USERNAME, "user");

curl\_easy\_setopt(curl, CURLOPT\_PASSWORD, "secret");

curl\_easy\_setopt(curl, CURLOPT\_URL, "smtp://mainserver.example.net:587");

curl\_easy\_setopt(curl, CURLOPT\_USE\_SSL, (long)CURLUSESSL\_ALL);

/\* 自带证书 \*/

curl\_easy\_setopt(curl, CURLOPT\_CAINFO, "/path/to/certificate.pem");

curl\_easy\_setopt(curl, CURLOPT\_MAIL\_FROM, FROM\_MAIL);

recipients = curl\_slist\_append(recipients, TO\_MAIL);

recipients = curl\_slist\_append(recipients, CC\_MAIL);

curl\_easy\_setopt(curl, CURLOPT\_MAIL\_RCPT, recipients);

curl\_easy\_setopt(curl, CURLOPT\_READFUNCTION, payload\_source);

curl\_easy\_setopt(curl, CURLOPT\_READDATA, &upload\_ctx);

curl\_easy\_setopt(curl, CURLOPT\_UPLOAD, 1L);

curl\_easy\_setopt(curl, CURLOPT\_VERBOSE, 1L);

/\* Send the message \*/

res = curl\_easy\_perform(curl);

/\* Free the list of recipients \*/

curl\_slist\_free\_all(recipients);

/\* Always cleanup \*/

curl\_easy\_cleanup(curl);

}

1. **异步邮件流程**

curl\_global\_init(CURL\_GLOBAL\_DEFAULT);

curl = curl\_easy\_init();

if(!curl)

return 1;

mcurl = curl\_multi\_init();

if(!mcurl)

return 2;

curl\_easy\_setopt(curl, CURLOPT\_URL, "smtp://mail.example.com");

curl\_easy\_setopt(curl, CURLOPT\_MAIL\_FROM, FROM\_MAIL);

recipients = curl\_slist\_append(recipients, TO\_MAIL);

recipients = curl\_slist\_append(recipients, CC\_MAIL);

curl\_easy\_setopt(curl, CURLOPT\_MAIL\_RCPT, recipients);

curl\_easy\_setopt(curl, CURLOPT\_READFUNCTION, payload\_source);

curl\_easy\_setopt(curl, CURLOPT\_READDATA, &upload\_ctx);

curl\_easy\_setopt(curl, CURLOPT\_UPLOAD, 1L);

/\* Tell the multi stack about our easy handle \*/

curl\_multi\_add\_handle(mcurl, curl);

do

{

CURLMcode mc = curl\_multi\_perform(mcurl, &still\_running);

if(still\_running)

/\* wait for activity, timeout or "nothing" \*/

mc = curl\_multi\_poll(mcurl, NULL, 0, 1000, NULL);

if(mc)

break;

} while(still\_running);

/\* Free the list of recipients \*/

curl\_slist\_free\_all(recipients);

/\* Always cleanup \*/

curl\_multi\_remove\_handle(mcurl, curl);

curl\_multi\_cleanup(mcurl);

curl\_easy\_cleanup(curl);

curl\_global\_cleanup();

==========================================================================================================================