Contents

[Post Method: 2](#_Toc81734705)

[POSTCLOB : Calling program 2](#_Toc81734706)

[POSTCLOB\_C – Copybook 2](#_Toc81734707)

[POSTCLOBV – Called program 3](#_Toc81734708)

[Unit test Post Method 4](#_Toc81734709)

[Put Method: 6](#_Toc81734710)

[PUTCLOB : Calling program 6](#_Toc81734711)

[PUTCLOB\_C : Copybook 6](#_Toc81734712)

[PUTCLOBV – Called program 7](#_Toc81734713)

[Unit test Put Method: 9](#_Toc81734714)

[Get Method: 10](#_Toc81734715)

[GETCLOB : program 10](#_Toc81734716)

[Delete Method: 12](#_Toc81734717)

[DELCLOB : Calling program 12](#_Toc81734718)

[Unit test delete Method: 14](#_Toc81734719)

# Post Method:

## POSTCLOB : Calling program

ctl-opt option(\*srcstmt:\*nodebugio);

ctl-opt dftActgrp(\*no) actGrp(\*caller);

copy QRPGLESRC,POSTCLOB\_C

dcl-pi postClob\_t;

email char(30);

name char(25);

gender char(10);

status char(10);

end-pi;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// MainLine

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

POSTCLOBV(email: name: gender: status);

\*inlr = \*on;

## POSTCLOB\_C – Copybook

dcl-pr postClobV extPgm('POSTCLOBV');

email char(30);

name char(25);

gender char(10);

status char(10);

end-pr;

dcl-pr postClob\_t extpgm('POSTCLOB\_T');

email char(30);

name char(25);

gender char(10);

status char(10);

end-pr;

dcl-pr system zoned(10:0) extproc('system');

szcmd pointer Value Options(\*String);

end-pr;

dcl-s body Char(200);

dcl-s body Char(200);

dcl-s command Char(200);

dcl-s messageText varchar(32740);

dcl-s messageLength int(5);

dcl-s responseMsg Varchar(9999);

dcl-s responseHeader Varchar(1000);

dcl-s responsePos Packed(4);

dcl-s returnedSQLCode char(5);

dcl-s tokenheader Char(500);

dcl-s header Char(500);

dcl-c fail const('FAIL');

dcl-c pass const('PASS');

dcl-c rcd const('Record Not Found');

dcl-c tokenKey const('Bearer 1e24167a1c6b84d104eab2d0bfa19f9a467a+

deab9b4132bd1d1d065a40c5f334');

dcl-c url const('https://gorest.co.in/public-api/users');

## POSTCLOBV – Called program

Ctl-opt option(\*srcstmt:\*nodebugio);

Ctl-opt dftactgrp(\*no) actgrp(\*caller);

/copy QRPGLESRC,postclob\_c

Dcl-pi POSTCLOBV;

email char(30);

name char(25);

gender char(10);

status char(10);

End-pi;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// MainLine

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SetEnvironment( );

ProcessApi( );

\*inlr = \*on;

Exec sql set option commit = \*none;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// SetRequest - Set the web service variables

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Dcl-proc SetEnvironment;

Exec sql CALL QSYS2.QCMDEXC('CHGJOB CCSID(37)');

Tokenheader = '<httpHeader>' +

'<header name="Authorization" value="' +

%trim(TokenKey) + '">' +

'</header>' +

'<header name="Content-Type" value="application/json">' +

'</header>' +

'</httpHeader>';

Body = '{' + '"email":' + '"' + %trim(email) + '",' +

'"name":' + '"' + %trim(name) + '",' +

'"gender":' + '"' + %trim(gender) + '",' +

'"status":'+ '"' + %trim(status) + '"' + '}';

End-proc;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// ProcessApi - Send Post Request And Fetch Data

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Dcl-proc ProcessApi;

Exec sql

Select Coalesce(Varchar(ResponseMsg,9999),' '),

Varchar(ResponseHttpHeader,1000) into :ResponseMsg,

:ResponseHeader

From Table(Systools.HttpPostClobVerbose(Trim(:Url),

Trim(:Tokenheader),

Trim(:body)))

as JsonResponse;

Diagnostics();

if ReturnedSqlCode = \*Blanks;

exec sql drop table if exists qtemp/WkTestFile;

exec sql create table qtemp/WkTestFile(data char(9999));

exec sql insert into qtemp/WkTestFile(data)

values(:ResponseMsg);

dsply 'Check WkTestFile file in Qtemp for data';

endIf;

End-proc;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Diagnostics - get sql details

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Dcl-proc Diagnostics ;

Exec sql GET DIAGNOSTICS CONDITION 1

:ReturnedSqlCode = DB2\_RETURNED\_SQLCODE,

:MessageLength = MESSAGE\_LENGTH,

:MessageText = MESSAGE\_TEXT;

End-proc ;

### Unit test Post Method

Call

A screenshot of a computer

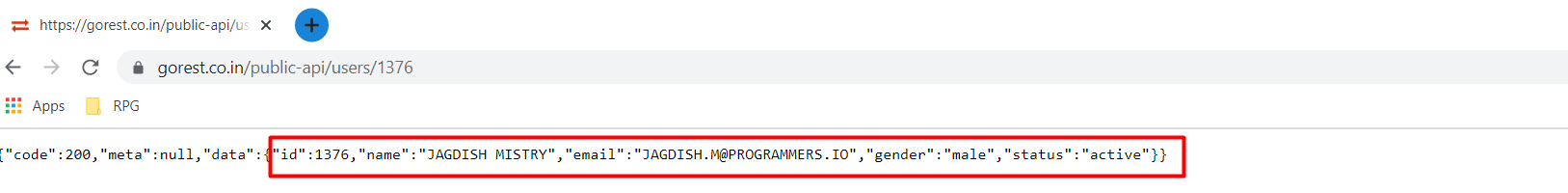
Description automatically generated with medium confidence

Responsemsg

A picture containing text

Description automatically generated

Using url



Completed.

# Put Method:

## PUTCLOB : Calling program

ctl-opt option(\*srcstmt:\*nodebugio);

ctl-opt dftActgrp(\*no) actGrp(\*caller);

/copy QRPGLESRC,PUTCLOB\_C

dcl-pi putClob;

id char(4);

email char(30);

name char(25);

gender char(10);

status char(10);

end-pi;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// MainLine

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

PUTCLOBV(id:email: name: gender: status);

\*inlr = \*on;

## PUTCLOB\_C : Copybook

dcl-pr putClobV extPgm('PUTCLOBV');

id char(4);

email char(30);

name char(25);

gender char(10);

status char(10);

end-pr;

dcl-pr putClob extpgm('PUTCLOB');

id char(4);

email char(30);

name char(25);

gender char(10);

status char(10);

end-pr;

//dcl-pr system zoned(10:0) extproc('system');

// szcmd pointer Value Options(\*String);

//end-pr;

dcl-s body Char(200);

dcl-s command Char(200);

dcl-s messageText varchar(32740);

dcl-s messageLength int(5);

dcl-s responseMsg Varchar(9999);

dcl-s responseHeader Varchar(1000);

dcl-s responsePos Packed(4);

dcl-s returnedSQLCode char(5);

dcl-s tokenheader Char(500);

dcl-s header Char(500);

dcl-s wurl Varchar(1000);

dcl-c fail const('FAIL');

dcl-c pass const('PASS');

dcl-c rcd const('Record Not Found');

dcl-c tokenKey const('Bearer 1e24167a1c6b84d104eab2d0bfa19f9a467a+

deab9b4132bd1d1d065a40c5f334');

dcl-c url const('https://gorest.co.in/public-api/users');

## PUTCLOBV – Called program

Ctl-opt option(\*srcstmt:\*nodebugio);

Ctl-opt dftactgrp(\*no) actgrp(\*caller);

/copy QRPGLESRC,putclob\_c

Dcl-pi PutCLOBV;

id char(4);

email char(30);

name char(25);

gender char(10);

status char(10);

End-pi;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// MainLine

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SetEnvironment( );

wurl = url + '/' + %trim(id) ;

ProcessApi( );

\*inlr = \*on;

Exec sql set option commit = \*none;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// SetRequest - Set the web service variables

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Dcl-proc SetEnvironment;

Exec sql CALL QSYS2.QCMDEXC('CHGJOB CCSID(37)');

Tokenheader = '<httpHeader>' +

'<header name="Authorization" value="' +

%trim(TokenKey) + '">' +

'</header>' +

'<header name="Content-Type" value="application/json">' +

'</header>' +

'</httpHeader>';

Body = '{' + '"email":' + '"' + %trim(email) + '",' +

'"name":' + '"' + %trim(name) + '",' +

'"gender":' + '"' + %trim(gender) + '",' +

'"status":'+ '"' + %trim(status) + '"' + '}';

End-proc;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// ProcessApi - Send Post Request And Fetch Data

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Dcl-proc ProcessApi;

Exec sql

Select Coalesce(Varchar(ResponseMsg,9999),' '),

Varchar(ResponseHttpHeader,1000) into :ResponseMsg,

:ResponseHeader

From Table(Systools.HttpPutClobVerbose(Trim(:wUrl),

Trim(:Tokenheader),

Trim(:body)))

as JsonResponse;

Diagnostics();

if ReturnedSqlCode = \*Blanks;

exec sql drop table if exists qtemp/WkTestFile;

exec sql create table qtemp/WkTestFile(data char(9999));

exec sql insert into qtemp/WkTestFile(data)

values(:ResponseMsg);

dsply 'Check WkTestFile file in Qtemp for data';

endIf;

End-proc;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Diagnostics - get sql details

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Dcl-proc Diagnostics ;

Exec sql GET DIAGNOSTICS CONDITION 1

:ReturnedSqlCode = DB2\_RETURNED\_SQLCODE,

:MessageLength = MESSAGE\_LENGTH,

:MessageText = MESSAGE\_TEXT;

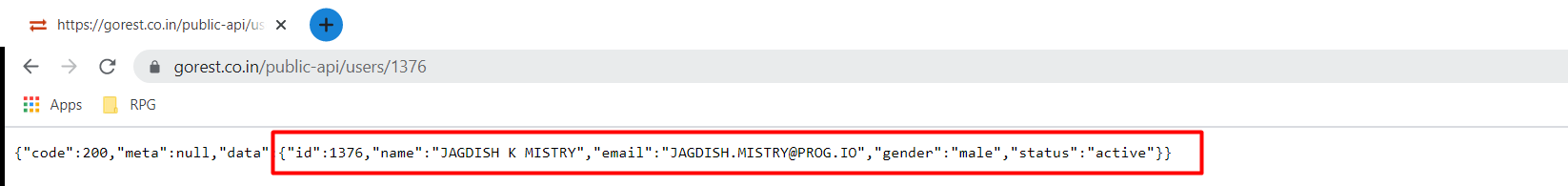
End-proc ;

### Unit test Put Method:

A screenshot of a computer

Description automatically generated with medium confidence

Url



Completed.

# Get Method:

## GETCLOB : program

h option(\*nodebugio:\*srcstmt) DFTACTGRP(\*NO)

Dcl-f JMOUT disk usage(\*Output) Rename(JMOUT:R1) Prefix(A);

Dcl-s MessageText varchar(32740);

Dcl-s MessageLength int(5);

Dcl-s ResponseMsg Varchar(9999);

Dcl-s ResponseHeader Varchar(1000);

Dcl-s ResponsePos Packed(4);

Dcl-s ReturnedSQLCode char(5);

Dcl-s header Char(200);

Dcl-c Url const('https://gorest.co.in/public/v1/users/1376');

//Dcl-pr system zoned(10:0) extproc('system');

// szcmd pointer Value Options(\*String);

//End-pr;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// MainLine

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Exec sql Set Option Commit = \*None;

\*inlr = \*on;

Exec sql set option commit = \*none;

Exec sql CALL QSYS2.QCMDEXC('CHGJOB CCSID(37)');

header =

'<httpHeader>' +

'<header name="Content-Type" value="text/plain">'+

'</header>' +

'</httpHeader>';

Exec sql

Select Coalesce(Varchar(ResponseMsg,9999),' '),

Varchar(ResponseHttpHeader,1000) into :ResponseMsg,

:ResponseHeader

From Table(Systools.HttpGetClobVerbose(Trim(:Url),

Trim(:header)))

as InternalServices;

AJMOUT = ResponseMsg;

Write R1;

Diagnostics();

If ReturnedSqlCode = \*Blanks;

dsply 'OK response with HttpGetClobVerbose';

Else;

dsply 'Error';

EndIf;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Diagnostics - get sql details

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Dcl-proc Diagnostics ;

Exec sql GET DIAGNOSTICS CONDITION 1

:ReturnedSqlCode = DB2\_RETURNED\_SQLCODE,

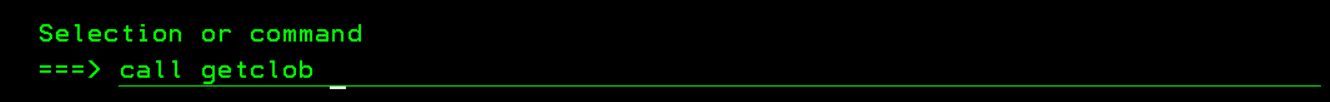
:MessageLength = MESSAGE\_LENGTH,

:MessageText = MESSAGE\_TEXT;

End-proc ;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Unit test Get method:



The response is written in out file



Completed.

# Delete Method:

## DELCLOB : Calling program

ctl-opt option(\*srcstmt:\*nodebugio);

ctl-opt dftActgrp(\*no) actGrp(\*caller);

/copy QRPGLESRC,DELCLOB\_C

dcl-pi delClob;

id char(4);

end-pi;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// MainLine

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

DELCLOBV(id);

\*inlr = \*on;

DELCLOB\_C : Copybook

dcl-pr delClobV extPgm('DELCLOBV');

id char(4);

end-pr;

dcl-pr delClob extpgm('DELCLOB');

id char(4);

end-pr;

//dcl-pr system zoned(10:0) extproc('system');

// szcmd pointer Value Options(\*String);

//end-pr;

dcl-s body Char(200);

dcl-s command Char(200);

dcl-s messageText varchar(32740);

dcl-s messageLength int(5);

dcl-s responseMsg Varchar(9999);

dcl-s responseHeader Varchar(1000);

dcl-s responsePos Packed(4);

dcl-s responsePos Packed(4);

dcl-s returnedSQLCode char(5);

dcl-s tokenheader Char(500);

dcl-s header Char(500);

dcl-s wurl Varchar(1000);

dcl-c fail const('FAIL');

dcl-c pass const('PASS');

dcl-c rcd const('Record Not Found');

dcl-c tokenKey const('Bearer 1e24167a1c6b84d104eab2d0bfa19f9a467a+

deab9b4132bd1d1d065a40c5f334');

dcl-c url const('https://gorest.co.in/public-api/users');

DELCLOB : Called program

Ctl-opt option(\*srcstmt:\*nodebugio);

Ctl-opt dftactgrp(\*no) actgrp(\*caller);

/copy QRPGLESRC,delclob\_c

Dcl-pi delCLOBV;

id char(4);

End-pi;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// MainLine

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SetEnvironment( );

wurl = url + '/' + %trim(id) ;

ProcessApi( );

\*inlr = \*on;

Exec sql set option commit = \*none;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// SetRequest - Set the web service variables

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Dcl-proc SetEnvironment;

Exec sql CALL QSYS2.QCMDEXC('CHGJOB CCSID(37)');

Tokenheader = '<httpHeader>' +

'<header name="Authorization" value="' +

%trim(TokenKey) + '">' +

'</header>' +

'<header name="Content-Type" value="application/json">' +

'</header>' +

'</httpHeader>';

//Body = '{' + '}';

Body = \*Blanks;

End-proc;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// ProcessApi - Send Post Request And Fetch Data

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Dcl-proc ProcessApi;

Exec sql

Select Coalesce(Varchar(ResponseMsg,9999),' '),

Varchar(ResponseHttpHeader,1000) into :ResponseMsg,

:ResponseHeader

From Table(Systools.HttpDeleteClobVerbose(Trim(:wUrl),

Trim(:Tokenheader)))

as JsonResponse;

// Trim(:body)))

Diagnostics();

if ReturnedSqlCode = \*Blanks;

exec sql drop table if exists qtemp/WkTestFile;

exec sql create table qtemp/WkTestFile(data char(9999));

exec sql insert into qtemp/WkTestFile(data)

values(:ResponseMsg);

dsply 'Check WkTestFile file in Qtemp for data';

endIf;

End-proc;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Diagnostics - get sql details

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Dcl-proc Diagnostics ;

Exec sql GET DIAGNOSTICS CONDITION 1

:ReturnedSqlCode = DB2\_RETURNED\_SQLCODE,

:MessageLength = MESSAGE\_LENGTH,

:MessageText = MESSAGE\_TEXT;

End-proc ;

### Unit test delete Method:

CALL PGM(DELCLOB) PARM('1376')

Graphical user interface

Description automatically generated with low confidence

URL

Graphical user interface, text, application

Description automatically generated

Completed.