IBM z/OS Connect EE V3.0

Developing CICS API Requester Applications

IBM Z
Wildfire Team –
Washington System Center

Lab Version Date: November 22, 2020

Table of Contents

Overview **4**
CICS RESTful API Requester **5**
Generate the API requester artifacts from a Swagger document *5*
Review the application programs *14*
Compile and link-edit the application programs *17*
Test the API requester application programs *18*
Summary **22**

Mitch Johnson (mitchj@us.ibm.com)

General Exercise Information and Guidelines

- ✓ The Windows artifacts for this exercise are in directory *c:\z\apiRequester*. Open a DOS command prompt Window and go to this directory using a change directory command, e.g. ***cd \z\apiRequester***
- ✓ Viewing or changing the contents of a file can easily be done by opening the Windows Explorer and going to directory *c:\z\apiRequester*. One this folder is display you can select a file and right mouse button click and select the *Open with EditPad Lite* option to open a file for viewing.
- ✓ This exercise requires using TSO user *USER1* and the TSO password for this user will be provided by the lab instructor.
- ✓ This exercise primarily uses data sets *USER1.ZCEE.CNTL* and *USER.ZCEE.SOURCE.*
- ✓ Please note that there may be minor differences between the screen shots in this exercise versus what you see when performing this exercise. These differences should not impact the completion of this exercise.
- ✓ Any time you have any questions about the use of IBM z/OS Explorer, 3270 screens, features or tools do not hesitate to ask the instructor for assistance.

Important: The REST API provider applications used in this exercise access a CICS applications and MQ. These APIs were developed using the instructions provided in other exercises in this workshop. The REST API providers could have just as easily have been anywhere in the cloud on any remote system as long as the Swagger document for the REST API was available.

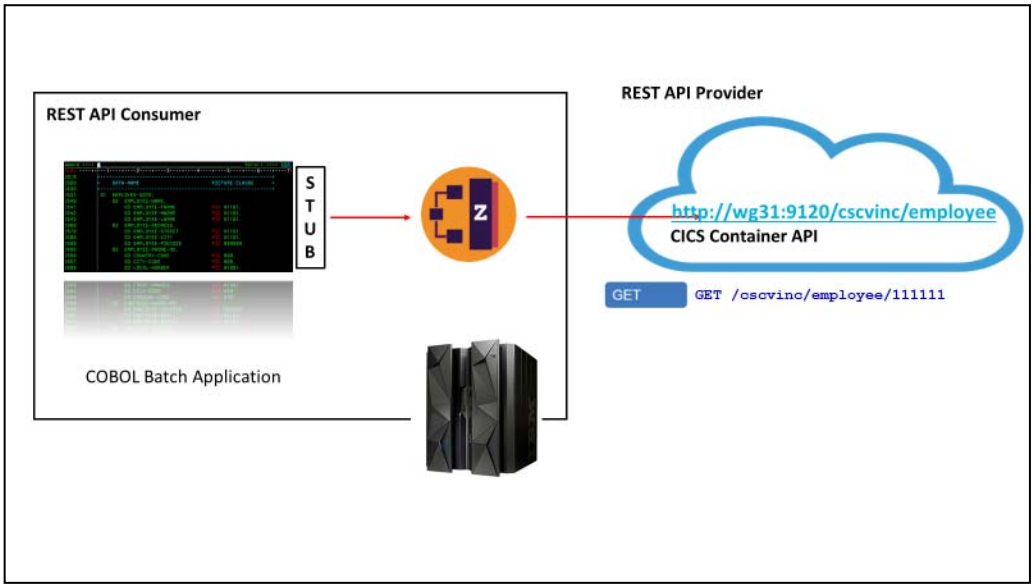
Mitch Johnson (mitchj@us.ibm.com)

Overview

These exercises demonstrate the steps required to enable COBOL programs to invoke RESTful APIs. There are two samples provided to do this. One sample uses RESTful APIs to access a CICS program and the other uses RESTful APIs to interact with an MQ queue. Note that even though these APIs are hosted in z/OS Connect EE server these same steps could be followed to access APIs hosted anywhere in the cloud.

The process starts with the Swagger document that describes the API which the COBOL program will invoke. The Swagger document is used by the z/OS Connect EE build toolkit to (1) generate the COBOL copy books included in the COBOL program to invoke the API (2) as well as an API requester archive file (ARA) which is deployed to a z/OS Connect EE server.

The diagram shows this process at a high level. The copy books generated by the z/OS Connect EE build toolkit will be integrated into a COBOL program to provide (1) the COBOL representation of the request and response messages and to identity (2) the HTTP method to be used in the RESTful request. The API requester archive file will be deployed to a z/OS Connect EE server and used to convert (1) the COBOL request and response messages to JSON messages and to invoke (2) the outbound RESTful request and wait for the response.



In this exercise the COBOL program will be compiled and executed in CICS. The only differences between batch and IMS versus CICS IMS is the configuration required for connecting to the z/OS Connect EE server. Consult the z/OS Connect EE Knowledge Center for information how this is done for MVS Batch, CICS and IMS.

Mitch Johnson (mitchj@us.ibm.com)

CICS RESTful API Requester

In this section a CICS based API requester will be developed and tested. The target API accessed from the CICS application is used to approve loan requests. The API uses the contents of JSON request to determine if a loan can be approved. The results are returned in a response message including the explanation if the loan is denied.

The rules for rejecting a loan can be for any one of the following:

- If the credit score of the borrower is less than 300.
- If the yearly repayment amount is more than 30% of the borrower's income.
- If the income of the borrower is less than $24,000.
- If the age of the borrower is more than 65 years.
- The loan amount is more than $1,000,000.

.

Generate the API requester artifacts from a Swagger document

The first step is to use to use the API's Swagger document to generate the artifacts required for the COBOL API client application and the API requester archive file. The Swagger document was obtained from the server where the API is hosted. All the files mention in this section are in directory *c:/z/apiRequester/miniloan* on your workstation.

The z/OS Connect EE build toolkit (ZCONBT) is a Java application shipped with z/OS Connect EE as a zip file and can be installed on a workstation or in OMVS. Anywhere a Java runtime is available. In this exercise the build toolkit will on Windows.

Mitch Johnson (mitchj@us.ibm.com)

___1. Begin by reviewing the API's Swagger document for the API. This document identifies the types of HTTP methods supported as well as the path of each HTTP method. It also the contents each method's request JSON message and the response JSON message field contents and field types. This is the blue print for invoking the API.


```
{

   "swagger": "2.0",
   "info": {
      "description": "",
      "version": "1.0.0",
      "title": "miniloancics"
   },
   "basePath": "/miniloancics",
   "schemes": [
      "https",
      "http"
   ],
   "consumes": [
      "application/json"
   ],
   "produces": [
      "application/json"
   ],
   "paths": {
      "/loan": {
         "post": {
            "tags": [
               "miniloancics"
```


Tech-Tip: Instructions for installing the z/OS Connect EE build toolkit can be found at URL https://www.ibm.com/support/knowledgecenter/en/SS4SVW_3.0.0/com.ibm.zosconnect.doc/installing/bt_install.html

Mitch Johnson (mitchj@us.ibm.com)

___2. Next review the input parameters that will be passed to the z/OS Connect EE build toolkit in a properties file. This file is *miniloan.properties*, see its contents below:

```
apiDescriptionFile=miniloan.json [1]
dataStructuresLocation=./syslib [2]
apiInfoFileLocation=./syslib [3]
logFileDirectory=./logs [4]
language=COBOL [5]
connectionRef=miniloancicsAPI [6]
requesterPrefix=min [7]
```

1. Identifies the Swagger document that describes the API

2. Identifies the location where the generated COBOL copybooks representing the request and response messages will be written.

3. Identifies the location where the generated COBOL copybook with invocation details of the API will be written.

4. Provides the location where a log file will be written.

5. Identifies the target language.

6. Identifies the corresponding *zosconnect_endpointConnection* element define in the z/OS Connect EE server's server.xml file.

7. Provides a 3-character prefix to be used for all generated copybooks

___3. View file *miniloan.bat* which contains this one line.

```
c:\z\zconbt\bin\zconbt.bat -p=./miniloan.properties -f=./miniloan.ara
```

This is batch command file that when executed invokes the z/OS Connect EE build toolkit *(zconbt*). The -p switch identifies the input properties file and the -f switch identifies the name of the generate API requester archive file. The API requester archive (ARA) file will be deployed and made available to the z/OS Connect EE server in directory *…/resources/zosconnect/apiRequesters.*

___4. On the workstation desktop, locate the *Command Prompt* icon and double click on it to open a DOS session that will allow commands to be entered.

___5. Use the change directory command (cd) to change to directory C:\z*api\Requester\miniloan*, e.g.

cd c:\z\apiRequester\miniloan

Mitch Johnson (mitchj@us.ibm.com)

___6. Enter command ***miniloan*** in the DOS command prompt window and you should see output like the below:

```
c:\z\apiRequester\miniloan>c:\z\software\zconbt\bin\zconbt.bat -p=./miniloan.properties -
f=./miniloan.ara
BAQB0000I: z/OS Connect Enterprise Edition Build Toolkit Version 1.2 (20200408-1120).
BAQB0008I: Creating API requester archive from configuration file ./miniloan.properties.
BAQB0040I: The generated API requester is automatically named miniloan_1.0.0 based on the title
miniloan and version 1.0.0 of the API to be called.
BAQB0015I: Start processing operation (operationId: postMiniloanService, relativePath: /loan, method:
POST).
BAQB0016I: Successfully processed operation (operationId: postMiniloanService, relativePath: /loan,
method: POST).

Total 1 operation(s) (success: 1, ignored: 0) defined in api description file: miniloan.json
------ Successfully processed operation(s) ------
operationId: postMiniloanService, basePath: /miniloan, relativePath: /loan, method: POST
- request data structure  : MIN00Q01
- response data structure : MIN00P01
- api info file           : MIN00I01

BAQB0009I: Successfully created API requester archive file ./miniloan.ara.
```

Note that each HTTP method defined in the Swagger document will cause the generation of up to three copy books. One for the request message(Q01), one for the response message(P01) and one for the API details(I01). The copy book names are based on the *requesterPrefix* property (e.g. *MIN*) and use an ascending sequence number sequence to differentiate between methods. Also note that there may be fewer than three copy books generated. For example, if there is no response message or no request message for a specific method a copy book may not be generated.

Important Note: The COBOL programs used in this exercise include these generated copy books with the names as generated above, e.g. the GET COBOL program includes MIN prefixed copy books. Occasionally (probably after applying service) the sequence of the methods will change causing the corresponding sequence number of the copy book to also change. Review the sequence of the processing of the methods and change the COBOL applications so they include the correct copy book. Otherwise the compilation of the COBOL program may fail with errors or unexpected results may be observed.

Mitch Johnson (mitchj@us.ibm.com)

___7. Now explore a sample snippet of the copy book for the request message of a POST method. Use *EditPad Lite* to open file c:\z\apiRequester\minloan\syslib\MIN00Q01. Scroll down until you see something like the code below.

```
            06 ReqBody.
              09 MINILOAN-COMMAREA2-num        PIC S9(9) COMP-5 SYNC.
               09 MINILOAN-COMMAREA.
                12 name-num                      PIC S9(9) COMP-5 SYNC.
                12 name.
                  15 name2-length                  PIC S9999 COMP-5 SYNC.
                  15 name2                         PIC X(20).
                12 creditScore-num                PIC S9(9) COMP-5 SYNC.
                12 creditScore                    PIC 9(18) DISPLAY.
                12 yearlyIncome-num               PIC S9(9) COMP-5 SYNC.
                12 yearlyIncome                   PIC 9(18) DISPLAY.
                12 age-num                        PIC S9(9) COMP-5 SYNC.
                12 age                           PIC 9(10) DISPLAY.
                12 amount-num                     PIC S9(9) COMP-5 SYNC.
                12 amount                         PIC 9(18) DISPLAY.
                12 effectiveDate-num              PIC S9(9) COMP-5 SYNC.
                12 effectiveDate.
                  15 effectiveDate2-length         PIC S9999 COMP-5 SYNC.
                  15 effectiveDate2                PIC X(8).
                12 yearlyRepayment-num            PIC S9(9) COMP-5 SYNC.
                12 yearlyRepayment                PIC 9(18) DISPLAY.
```

Note that each variable has an associated length (e.g. *name2-length).* The length of a variable is required since there is no delimiter between the variables in storage. The runtime needs to know the size of a variable at execution time, so the request and response messages can be properly converted to and from JSON name/value pairs.

For this API, the Swagger document included an additional variable for the number of occurrences of a variable, (e.g. *name-num*). In this exercise we will set the number of each variable to one.

The corresponding response message (*c:\z\apiRequester\miniloan\syslib\MIN00P01*) will have a similar layout.

Mitch Johnson (mitchj@us.ibm.com)

___8. The corresponding API information copy book (*c:\z\apiRequester\miniloan\syslib\MIN00I01*) has contents providing the API name and the path to be used for the method:

```
           03 BAQ-APINAME                PIC X(255)
              VALUE 'miniloancics_1.0.0'.
           03 BAQ-APINAME-LEN            PIC S9(9) COMP-5 SYNC
              VALUE 18.
           03 BAQ-APIPATH                PIC X(255)
              VALUE '%2Fminiloancics%2Floan'.
           03 BAQ-APIPATH-LEN            PIC S9(9) COMP-5 SYNC
              VALUE 22.
           03 BAQ-APIMETHOD              PIC X(255)
              VALUE 'POST'.
           03 BAQ-APIMETHOD-LEN          PIC S9(9) COMP-5 SYNC
              VALUE 4.
```

Deploy the API Requester Archive file to the z/OS Connect EE server

Next make the API requester archive file available to the z/OS Connect EE server.

The API requester archive (ARA) file needs to be deployed to the API requester directory. In this case the target directory is */var/ats/zosconnect/servers/zceeapir/resources/zosconnect/apiRequesters*.

___1. Open a DOS command session and change to directory c:\z\apiRequester\miniloan, e.g **cd c:\z\apiRequester\miniloan**

___2. Use the z/OS Connect EE RESTful administrative interface to deploy the ARA files by using the cURL command embedded in command file *deploy.cmd*. Invoke this command file to deploy the miniloan API archive file.

```
C:\z\apiRequester\miniloan>curl -X POST --user user1:user1 --data-binary @minilo
an.ara  --header "Content-Type: application/zip" --insecure https://wg31.washing
ton.ibm.com:9483/zosConnect/apiRequesters/miniloan
{"name":"miniloan_1.0.0","version":"1.0.0","description":"","status":"Starte
d","apiRequesterUrl":"https:\/\/wg31.washington.ibm.com:9483\/zosConnect\/apiReq
uesters\/miniloan_1.0.0","connection":"miniloanAPI"}
```

Tech-Tip: If a REST client tool like cURL or Postman was not available then ARA file could have been deployed using FTP to upload the file in binary mode to the *apiRequesters* directory.

Tech-Tip: If an ARA needs to be redeployed, the cURL command with a PUT method can be used to stop the API requester
curl -X PUT --user USER1:USER1 --insecure https://wg31.washington.ibm.com:9483/zosConnect/apiRequesters/miniloan_1.0.0?status=stopped
And the cURL command with a DELETE method can be used to delete the API requester.
curl -X DELETE --user USER1:USER1 --insecure https://wg31.washington.ibm.com:9483/zosConnect/apiRequesters/miniloan_1.0.0
otherwise the redeployment of the ARA file will fail.

___3. The z/OS Connect EE server where the API requester will be installed has the configuration below.

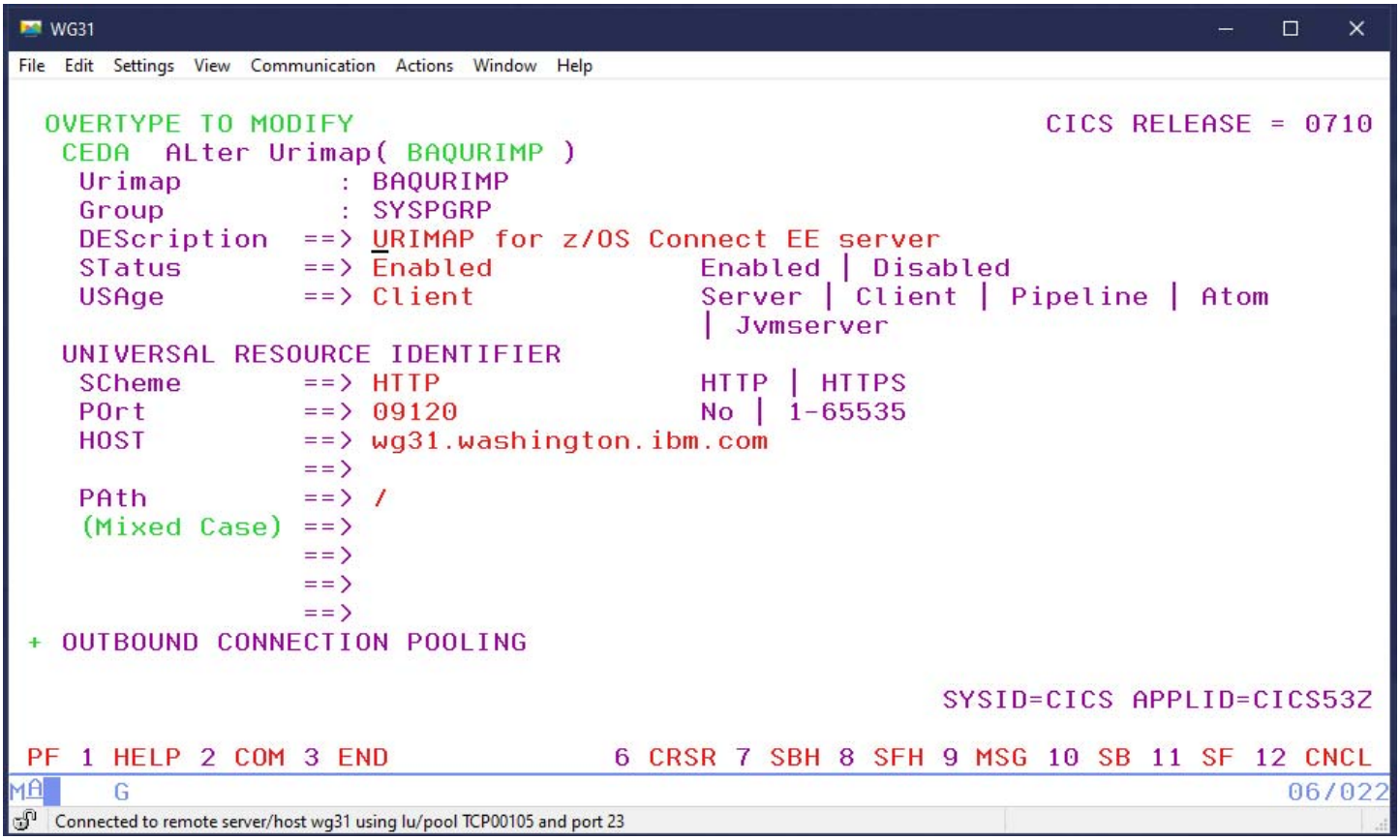
```
<server description="API Requester">
   <!-- Enable features -->
   <featureManager>
       <feature>zosconnect:apiRequester-1.0</feature>
   </featureManager>


<zosconnect_endpointConnection id="miniloancicsAPI"    1
        host="https://wg31.washington.ibm.com"
        port="9120"
        basicAuthRef="myBasicAuth"
        connectionTimeout="10s"
        receiveTimeout="20s" />

 <zosconnect_authData id="myBasicAuth"     2
        user="Fred"
        password="fredpwd" />

</server>
```

1. Identifies the system which hosts the target API as well as any security information and time out parameters. Note that the ID of this element matches the value *connectionRef* property used when the ARA was generated by the z/OS Connect build toolkit.
2. Identifies the system which hosts the target API as well as any security information and time out parameters. Note that the ID of this element matches the value *connectionRef* property used when the ARA was generated by the z/OS Connect build toolkit.

Tech-Tip: Explicitly defining the apiRequesters element for an API Requester prevents use of the RESTful administrative interface to manage the API requester.

Mitch Johnson (mitchj@us.ibm.com)

___4. The API requester stubs uses the BAQURIMP resource defined to CICS to locate the z/OS Connect EE server where the request will be sent.



Mitch Johnson (mitchj@us.ibm.com)

Move the COBOL copy books to an MVS data set

Next make the generated COBOL copy books available for including into the COBOL program.

The generated COBOL copy books need to be available in an MVS data set which is included in the compiler SYSLIB concatenation sequence. In the data set is *USER1.ZCEE. SOURCE*. Use a DOS command prompt session and the commands below to move the copy books file to this data set. Authenticate as user USER1 (password USER1).

___1. Open a DOS command prompt and use the change directory command to go to directory C:\z\apiRequester\miniloan\syslib, e.g. ***cd \z\apiRequester\miniloan\syslib***

___2. Start a file transfer session with the WG31 host using the *ftp* command, e.g. **ftp wg31**

___3. Logon as USER1 and then use the *cd* command to change to directory to data set *USER1.ZCEE. SOURCE*, *e.g.* ***cd zcee.source***

___4. Toggle prompting off by entering command **prompt**

___5. Perform multiple put requests by using the multiple put command, **mput min***

___6. When the last transfer has completed enter the **quit** command.

```
c:\z\apiRequester\miniloan>cd syslib
c:\z\apiRequester\miniloan\syslib>ftp wg31.washington.ibm.com
Connected to wg31.washington.ibm.com.
220-FTP 16:26:23 on 2018-02-15.
220 Connection will close if idle for more than 200 minutes.
User (wg31.washington.ibm.com:(none)): user1
331 Send password please. user1
Password:
230 USER1 is logged on.  Working directory is "USER1.".
ftp> cd zcee.source
250 The working directory "USER1.ZCEE.SOURCE" is a partitioned data set
ftp> prompt
Interactive mode Off .
ftp> mput MIN*
200 Port request OK.
125 Storing data set USER1.ZCEE.SOURCE(MIN00I01)
250 Transfer completed successfully.
ftp: 544 bytes sent in 0.00Seconds 544000.00Kbytes/sec.
200 Port request OK.
125 Storing data set USER1.ZCEE.SOURCE(MIN00P01)
250 Transfer completed successfully.
ftp: 13081 bytes sent in 0.00Seconds 13081000.00Kbytes/sec.
200 Port request OK.
125 Storing data set USER1.ZCEE.SOURCE(MIN00Q01)
250 Transfer completed successfully.
ftp: 6980 bytes sent in 0.00Seconds 6980000.00Kbytes/sec.
ftp> quit
```

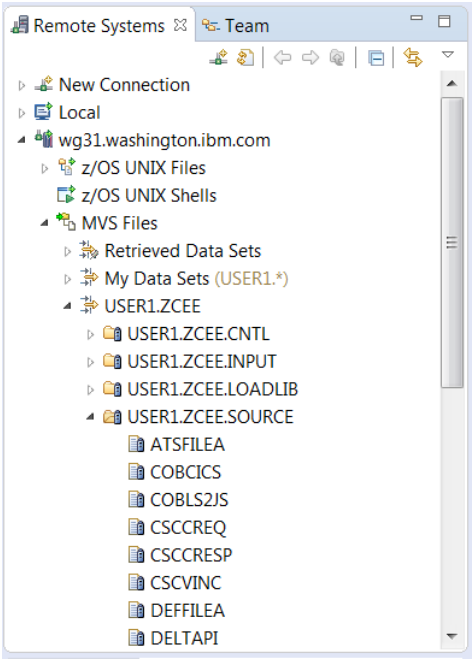

Mitch Johnson (mitchj@us.ibm.com)

Review the application programs

Now that the copy books have been generated and are available in a data set the next step is compile and link edit the application program.

> **Tech-Tip:** Data set USER1.ZCEE.SOURCE can be accessed either by using the IBM z/OS Explorer filter or by using the Personal Communication icon on the desktop and logging on to TSO and use ISPF option 3.4 to access this data set. If you have any questions about either method, ask the instructor.

___1. Back in the IBM z/OS Explorer session switch to the *Remote System Explorer* perspective (see page **Error! Bookmark not defined.**) and expand data set *USER1.ZCEE.SOURCE* in the *Remote System view* to display a list of its members.



___2. Begin by reviewing the application programs LOANAPIR. Open member *LOANAPIR in USER1.ZCEE.SOURCE* and review its contents by selecting the member and right- mouse button clicking and selecting the *Open* option. Scroll down until you see these lines of code.

These lines of code copy into the source the required z/OS Connect EE copy book (BAQRINFO) and the copy books generated by the z/OS Connect EE build tool kit.

```
      * Copy API Requester required copybook
       COPY BAQRINFO.

      * Request and Response
       01 POST-REQUEST.
           COPY MIN01Q01.
       01 POST-RESPONSE.
           COPY MIN01P01.
      * Structure with the API information
       01 POST-INFO-OPER1.
           COPY MIN01I01.
```


Mitch Johnson (mitchj@us.ibm.com)

Scroll down further until you find this code which sets the contents of the request message and invoke the API.

```
            WHEN DFHENTER
              MOVE  NAMEI     to NAME2 in POST-REQUEST
              MOVE  LENGTH of NAME2 in POST-REQUEST to
                    NAME2-length in POST-REQUEST
              MOVE  AGEI      to AGE in POST-REQUEST
              MOVE  CRDSCREI  to CREDITSCORE in POST-REQUEST
              MOVE  INCOMEI   to YEARLYINCOME in POST-REQUEST
              MOVE  AMOUNTI   to AMOUNT       in POST-REQUEST
              MOVE  YRPAYMNTI
                        To YEARLYREPAYMENT in POST-REQUEST
              MOVE 1 to MINILOAN-COMMAREA2-num IN POST-REQUEST
              MOVE 1 to NAME-num               IN POST-REQUEST
              MOVE 1 to CREDITSCORE-num        IN POST-REQUEST
              MOVE 1 to YEARLYINCOME-num       IN POST-REQUEST
              MOVE 1 to AGE-num                IN POST-REQUEST
              MOVE 1 to AMOUNT-num             IN POST-REQUEST
              MOVE 1 to YEARLYREPAYMENT-num    IN POST-REQUEST
              PERFORM INVOKE-API
```

___3. Scroll down further to see the CICS commands to display the results on the screen.

```
            IF BAQ-SUCCESS THEN
              IF APPROVED2 in POST-RESPONSE = 'T'
                MOVE 'Loan approved' to approvedo
              ELSE
                MOVE 'Loan not approved' to approvedo
              END-IF
              MOVE UID2       TO UIDO
              MOVE MESSAGES2(1) TO MSG1O
              MOVE MESSAGES2(2) TO MSG2O
              MOVE MESSAGES2(3) TO MSG3O
              MOVE MESSAGES2(4) TO MSG4O
              MOVE MESSAGES2(5) TO MSG5O
              MOVE MESSAGES2(6) TO MSG6O
              MOVE MESSAGES2(7) TO MSG7O
              MOVE MESSAGES2(8) TO MSG8O
              MOVE MESSAGES2(9) TO MSG9O
            ELSE
….
            END-IF.
       EXEC CICS SEND CONTROL ERASE END-EXEC
       EXEC CICS SEND MAP('MINIMAP')
          MAPSET('MINIMAP')
          FREEKB ERASE END-EXEC
       EXEC CICS RETURN TRANSID ('APIR')
          COMMAREA (COMMAREA-BUFFER) END-EXEC.
```


Mitch Johnson (mitchj@us.ibm.com)

___4. Scroll down further and you will the call to the z/OS Connect API requester stub module.

```
       INVOKE-API SECTION.
     *---------------------------------------------------------------*
     * Initialize API Requester PTRs & LENs                          *
     *---------------------------------------------------------------*
     * Use pointer and length to specify the location of
     *  request and response segment.
     * This procedure is general and necessary.
          SET BAQ-REQUEST-PTR TO ADDRESS OF POST-REQUEST
          MOVE LENGTH OF POST-REQUEST TO BAQ-REQUEST-LEN
          SET BAQ-RESPONSE-PTR TO ADDRESS OF POST-RESPONSE
          MOVE LENGTH OF POST-RESPONSE TO BAQ-RESPONSE-LEN

     *---------------------------------------------------------------*
     * Call the communication stub                                   *
     *---------------------------------------------------------------*
     * Call the subsystem-supplied stub code to send
     * API request to zCEE
          CALL COMM-STUB-PGM-NAME USING
               BY REFERENCE   POST-INFO-OPER1
               BY REFERENCE   BAQ-REQUEST-INFO
               BY REFERENCE   BAQ-REQUEST-PTR
               BY REFERENCE   BAQ-REQUEST-LEN
               BY REFERENCE   BAQ-RESPONSE-INFO
               BY REFERENCE   BAQ-RESPONSE-PTR
               BY REFERENCE   BAQ-RESPONSE-LEN.
* The BAQ-RETURN-CODE field in 'BAQRINFO' indicates whether this
     * API call is successful.
       CHECK-API-ERROR SECTION.
     * Some error happened in API, z/OS Connect EE server
     * or communication stub. 'BAQ-STATUS-CODE' and
     * 'BAQ-STATUS-MESSAGE' contain the detailed information
     *  of this error.
            DISPLAY "Error code: " BAQ-STATUS-CODE
            DISPLAY "Error msg:" BAQ-STATUS-MESSAGE
            MOVE BAQ-STATUS-CODE TO EM-CODE
            MOVE BAQ-STATUS-MESSAGE TO EM-DETAIL
            EVALUATE TRUE
     * When error happens in API, BAQ-RETURN-CODE is BAQ-ERROR-IN-API.
     * BAQ-STATUS-CODE is the HTTP response code of API.
              WHEN BAQ-ERROR-IN-API
                MOVE 'API' TO EM-ORIGIN
     * When error happens in server, BAQ-RETURN-CODE is
     * BAQ-ERROR-IN-ZCEE
     * BAQ-STATUS-CODE is the HTTP response code of
     * z/OS Connect EE server.
              WHEN BAQ-ERROR-IN-ZCEE
                MOVE 'ZCEE' TO EM-ORIGIN
     * When error happens in communication stub, BAQ-RETURN-CODE is
     * BAQ-ERROR-IN-STUB, BAQ-STATUS-CODE is the error code of STUB.
              WHEN BAQ-ERROR-IN-STUB
                MOVE 'STUB' TO EM-ORIGIN
            END-EVALUATE
```

Note that the parameters are being passed by reference. This means that the z/OS Connect API requester stub will have direct access to the program storage for both accessing data and making changes (i.e. the response message).

Compile and link-edit the application programs

The applications programs now are ready to be compiled and link edited.

___1. Submit the job in member **LOANAPIR** in data set *USER1.ZCEE.CNTL*.
Note that the CALL statement in each of this program uses a variable for the program name z/OS Connect EE stub program (BAQCSTUB). This means that this program will by dynamically linked at execution. The load module library containing the CICS version of BAQSCTUB (SBAQLIB1) must be available in the DFHRPL concatenation sequence when the program is executed.

```
//COMPILE EXEC IGYWCL,LNGPRFX=IGY610,
// PARM.COBOL='LIB,NODYNAM,CICS,SIZE(4000K)'
//COBOL.STEPLIB DD
//             DD DISP=SHR,DSN=CICSTS54.CICS.SDFHLOAD
//COBOL.SYSIN  DD DISP=SHR,DSN=USER1.ZCEE.SOURCE(LOANAPIR)
//COBOL.SYSLIB DD DISP=SHR,DSN=USER1.ODM.JCL
//             DD DISP=SHR,DSN=USER1.ZCEE.SOURCE
//             DD DISP=SHR,DSN=ZCEE30.SBAQCOB
//             DD DISP=SHR,DSN=CICSTS54.CICS.SDFHCOB
//LKED.SYSLMOD DD DISP=SHR,DSN=USER1.ZCEE.LOADLIB(LOANAPIR)
//LKED.CICSLIB DD DISP=SHR,DSN=CICSTS54.CICS.SDFHLOAD
//LKED.SYSIN   DD *
 INCLUDE CICSLIB(DFHELII)
```

This job should complete with a zero-return code.

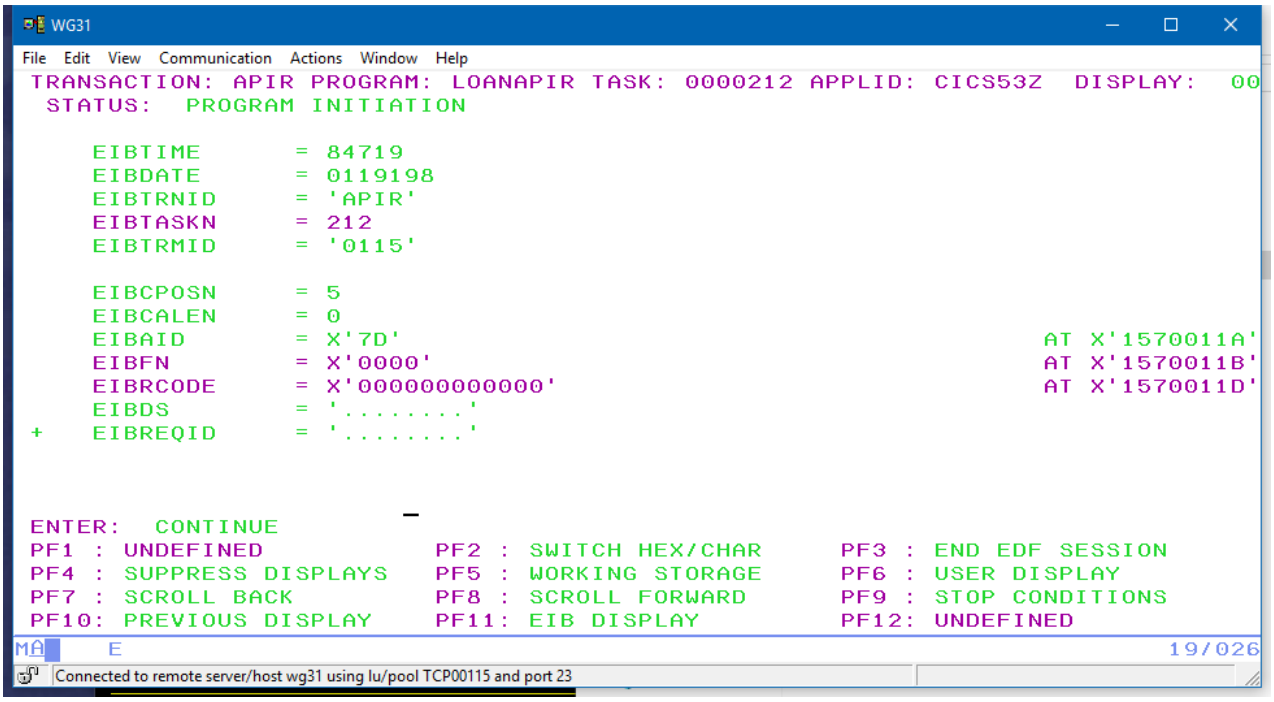
Tech Tip: To submit a job when using the IBM z/OS Explorer select the member and right mouse button click and select the *Submit* option. Or if the member is currently opened simply right mouse button click and select the *Submit* option. Click the **Locate Job** button on the *Job Confirmation* pop-up. This will display the job output in the *Retrieved Job* section under *JES* in the *Remote Systems* view. The job's output can be viewed right mouse button clicking and selecting the *Open* option.

Mitch Johnson (mitchj@us.ibm.com)

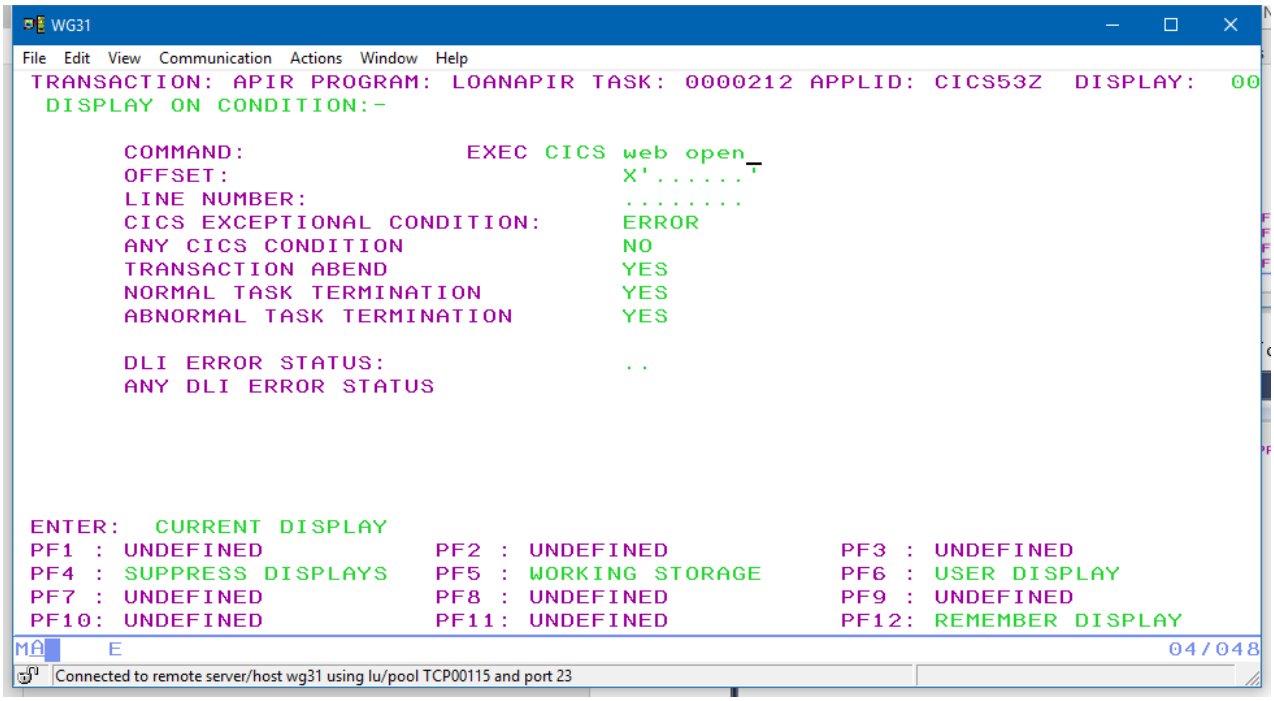
Test the API requester application programs

___1. In a new or existing 3270 session start a session with CICS, clear the screen and enter CICS transaction ***CEDF*** to start a CICS execution diagnostic facility (EDF).

___2. Clear the screen again and enter CICS transaction ***APIR***.

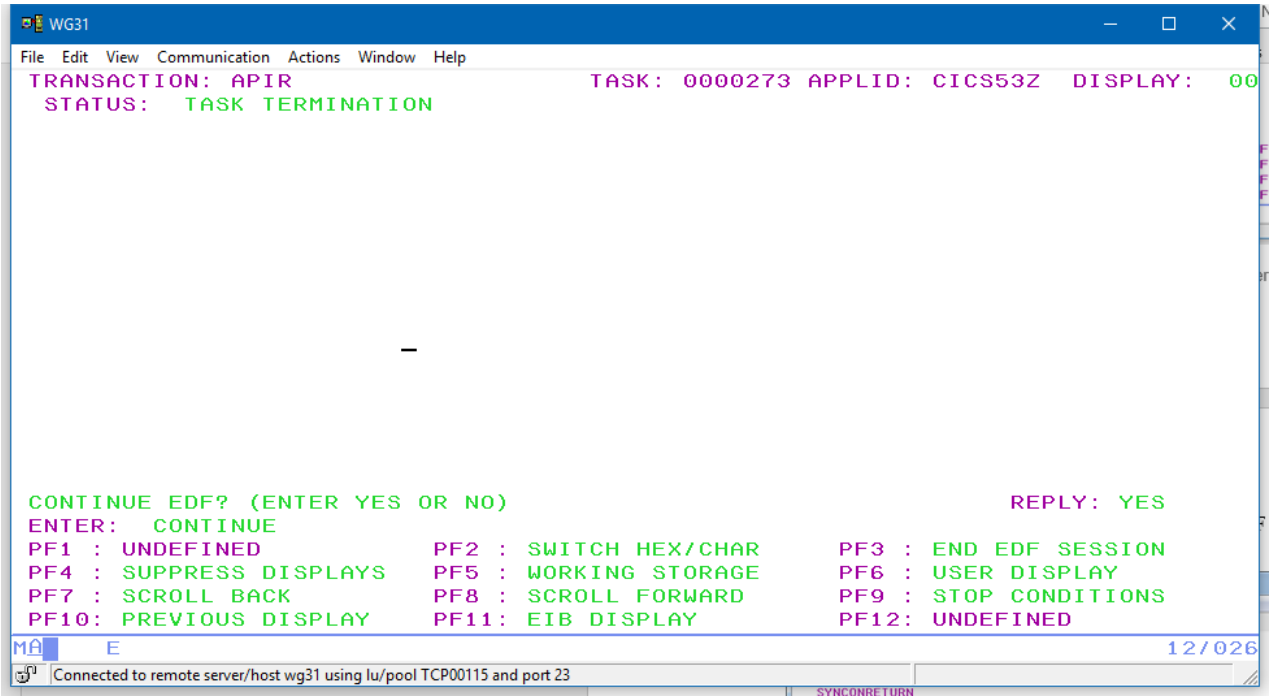


___3. Use the **F9** key to set a stop condition. Set a stop on ***EXEC CICS WEB OPEN*** commands.

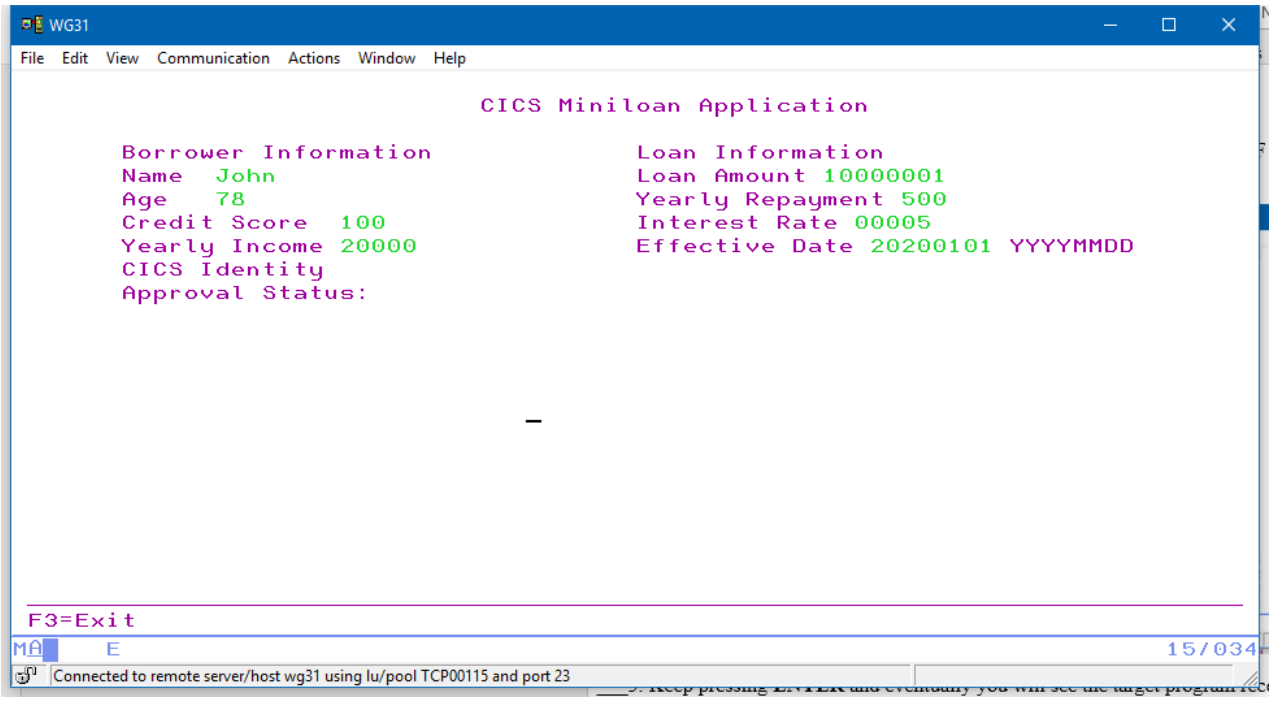


Mitch Johnson (mitchj@us.ibm.com)

___4. Use the **F4** key to suppress the intervening EDF displays. Eventually EDF will stop on a task termination screen. Ensure the *Reply* is set to **Yes** and press **Enter** to continue.

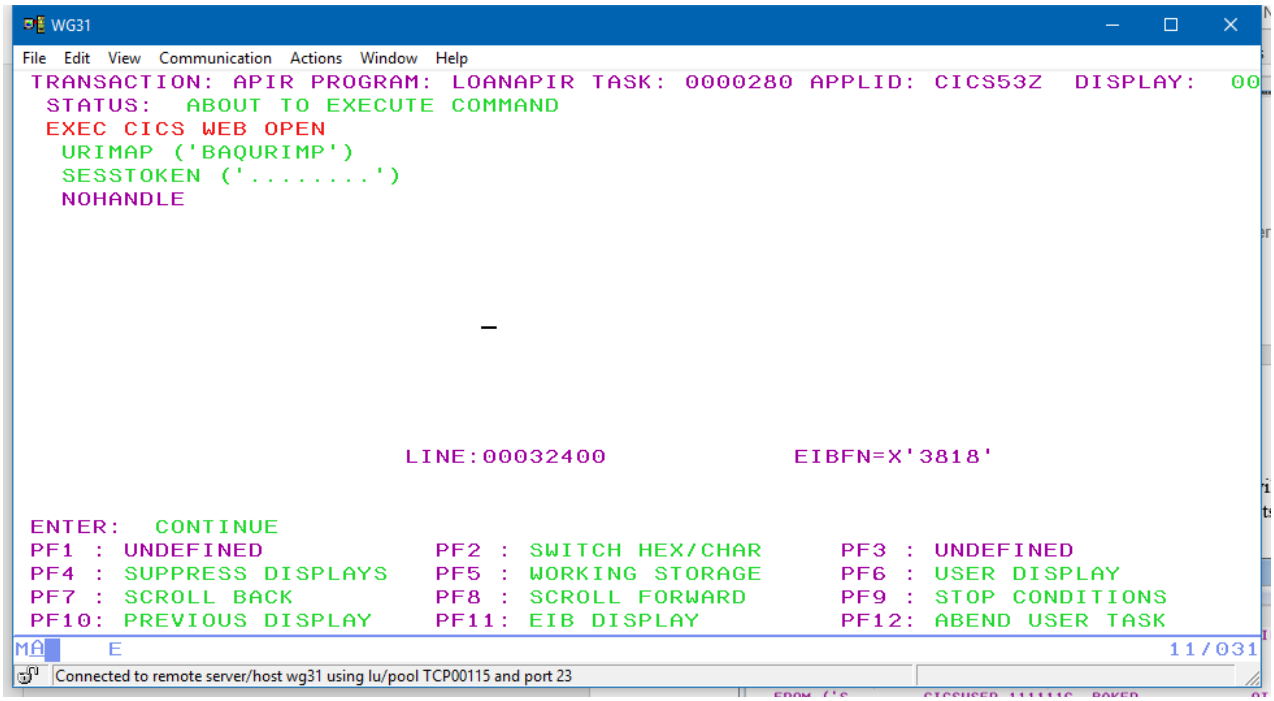


___5. Enter the information shown below and press **Enter.**

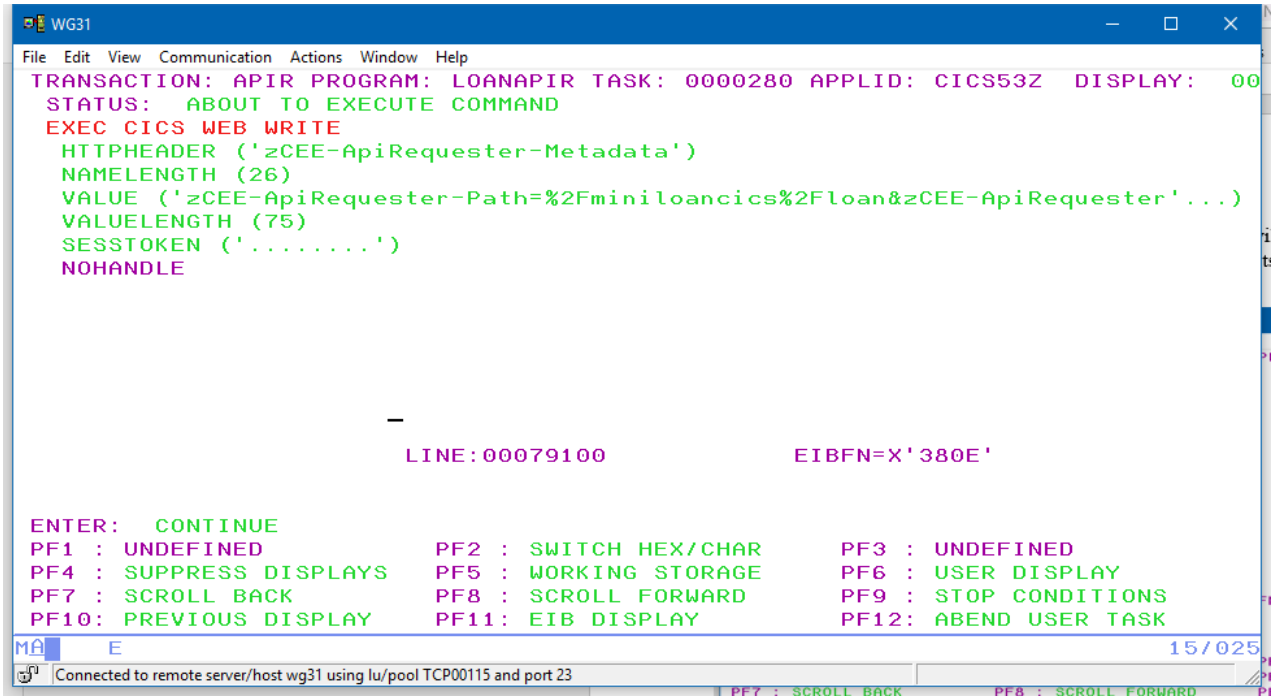


Mitch Johnson (mitchj@us.ibm.com)

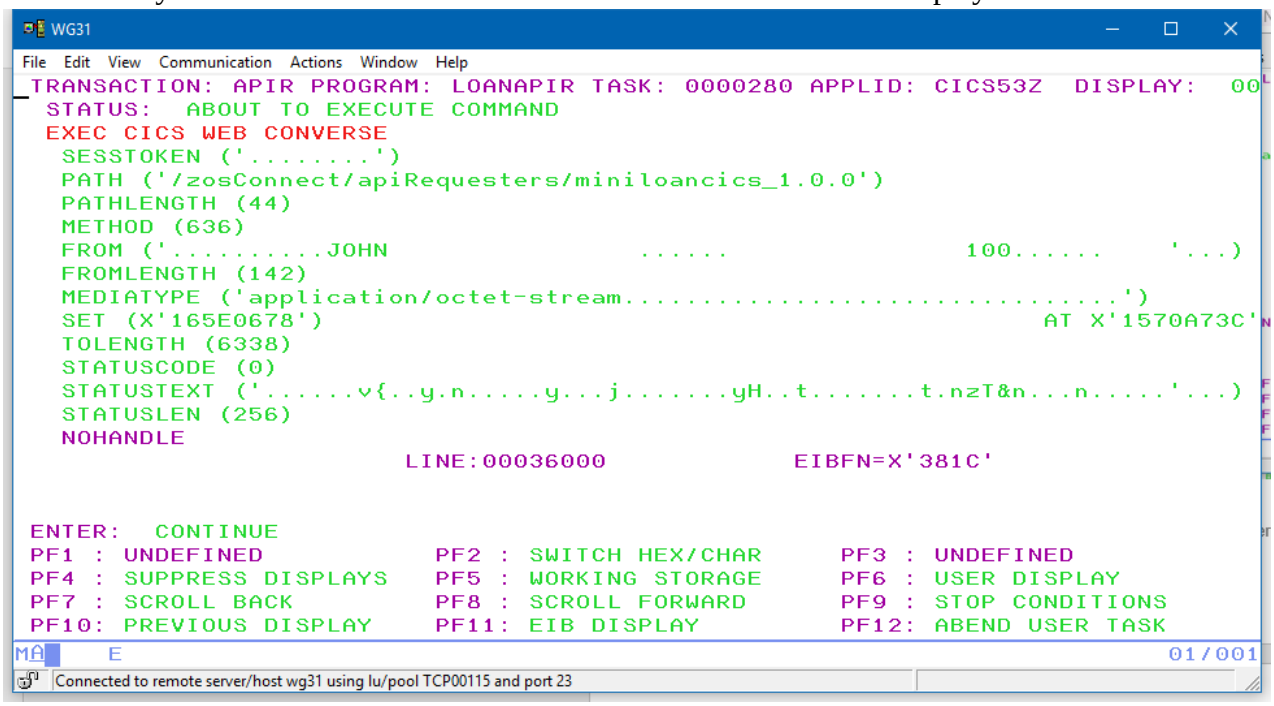
___6. Suppress the EDF displays again using the **F4** key. Eventually there will be *EXEC CICS WEB OPEN* request.



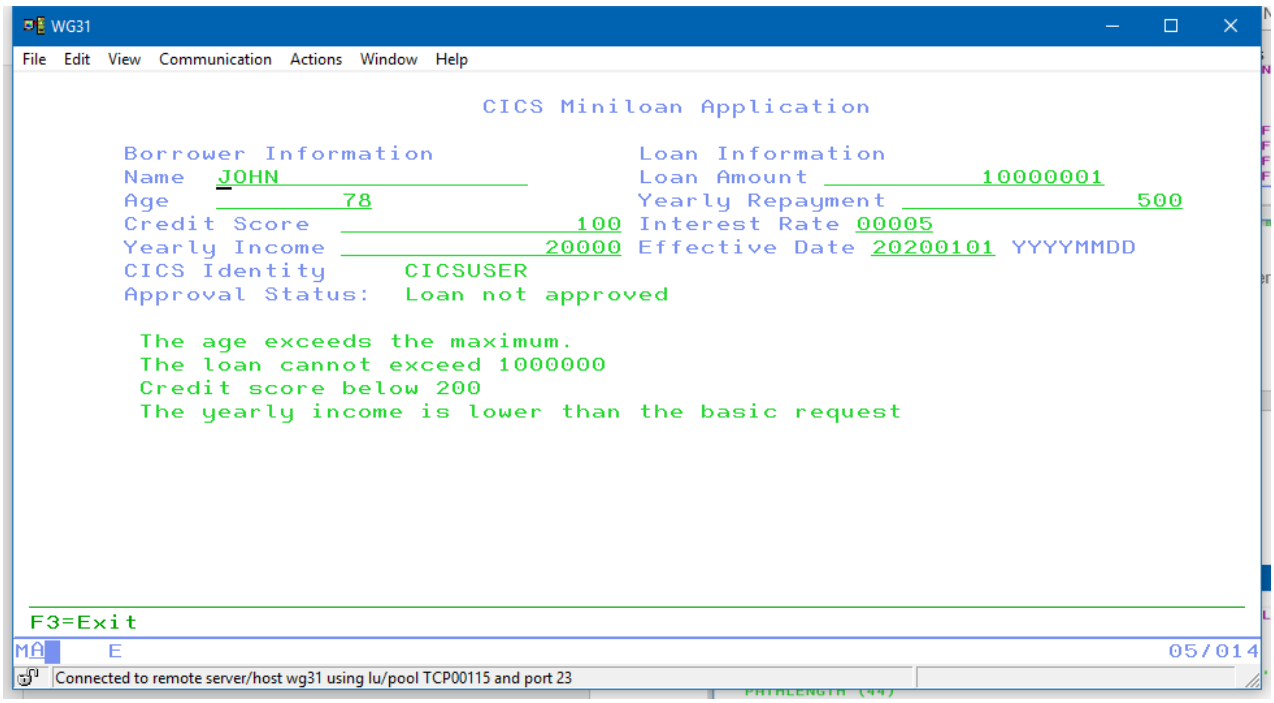
___7. Keep pressing the **Enter** key until the *EXEC CICS WEB WRITE* EDF screen is displayed.



___3. Press the **Enter** key until the *EXEC CICS WEB CONVERSE* EDF screen is displayed.



___4. Keep pressing **Enter** until the results of the loan application is displayed.



___5. Repeat loan applications with and/or without EDF to see the results of various scenarios.

Mitch Johnson (mitchj@us.ibm.com)

Summary

You have use the z/OS Connect build toolkit to generate an API requester archive file and the COBOL copy books that must be included in the COBOL application program when accessing the API requester archive file in a z/OS Connect server. The COBOL applications have been compiled and link edited, and the target API has been tested using various RESTful methods

Mitch Johnson (mitchj@us.ibm.com)