

IBM z/OS Connect EE V3.0

# Customization - Basic Configuration

(1 of 2)



*Lab Version Date: April 19, 2020*

## Table of Contents

<b>General Exercise Information and Guidelines .....</b>	<b>4</b>
<b>Setup and Service Definitions .....</b>	<b>5</b>
Run the jobs to setup RACF framework for server runtime.....	5
Create a z/OS Connect EE 3.0 server .....	11
<b>Summary .....</b>	<b>15</b>
Customize the JCL start procedures and start the server.....	16
Verify the z/OS Connect server configuration.....	26
<b>Deploying Services and APIs .....</b>	<b>31</b>
Update the z/OS Connect EE Server Configuration .....	31
Deploy the Services .....	33
Using Postman .....	33
Using cURL.....	36
Test the Services.....	39
Using Postman .....	39
Using cURL.....	43
Deploy and Test the API.....	47
Optional .....	51

**Important:** On the desktop there is a file named *Basic Configuration CopyPaste.txt*. This file contains commands and other text used in this workshop. Locate that file and open it. Use the copy-and-paste function (**Ctrl-C** and **Ctrl-V**) to enter commands or text. It will save time and help avoid typo errors. As a reminder text that appears in this file will be highlighted in yellow.



## ***General Exercise Information and Guidelines***

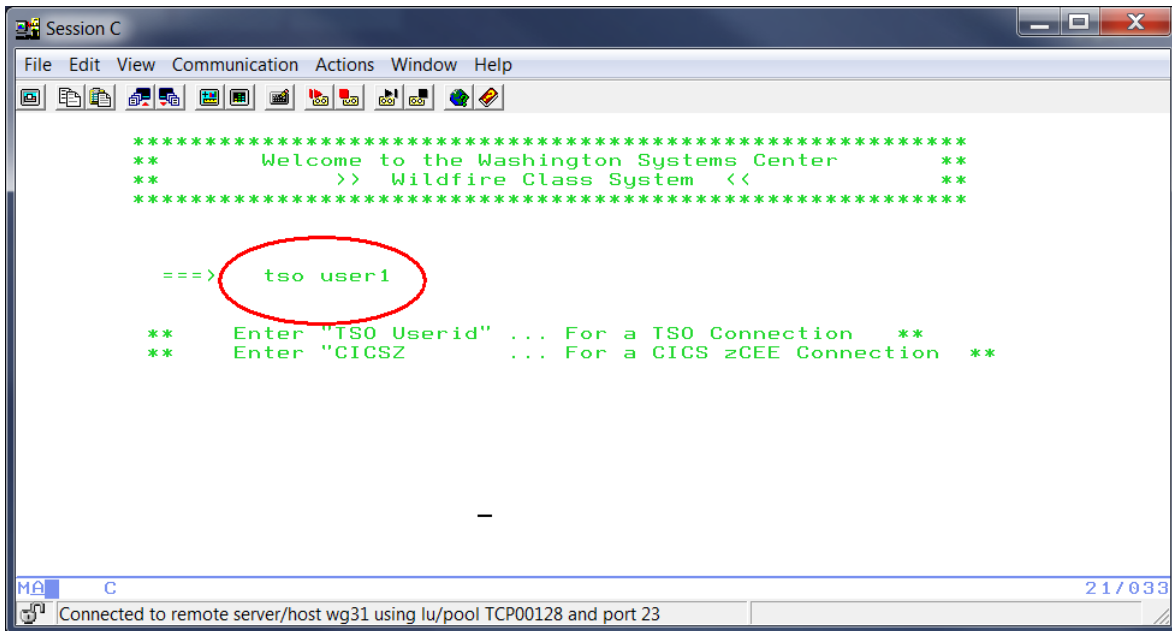
- ✓ This exercise requires using z/OS user identity *USER1*. The password for this user will be provided by the lab instructor.
- ✓ Any time you have any questions about the use of IBM z/OS Explorer, 3270 screens, features or tools, do not hesitate to ask the instructor for assistance.
- ✓ Text in **bold** and highlighted in **yellow** in this document should be available for copying and pasting in a file named *Basic Configuration CopyPaste* file on the desktop.
- ✓ 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.

## Setup and Service Definitions

### Run the jobs to setup RACF framework for server runtime

1. Open the *WG31* icon on the workstation desktop. This will start a 3270-terminal session to your z/OS system.

**Tech-Tip:** Desktop tools can be opened either by double clicking the icon or by selecting the icon and right mouse button clicking and then selecting the *Open* option.



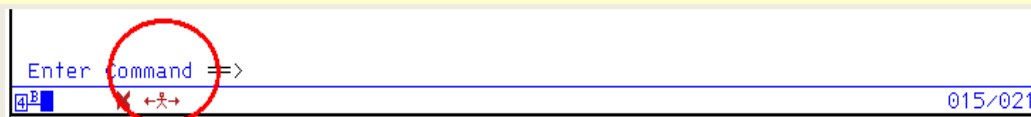
N.B., The 3270-terminal sessions and OMVS screen shots in the remainder of this and subsequent exercises are shown in reverse video simply for printing purposes

2. Enter ***TSO USER1*** (see below) and press the 3270 **Enter** key (the **right-Ctrl** key sequence):  
The 3270-emulator used for this workshop (IBM Personal Communication) maps the 3270 enter key to the right **Ctrl** key (see below). Any references to the *Enter* key in non-3270 windows, OMVS terminal session, etc. refers to the key labeled *Enter* on the keyboard.



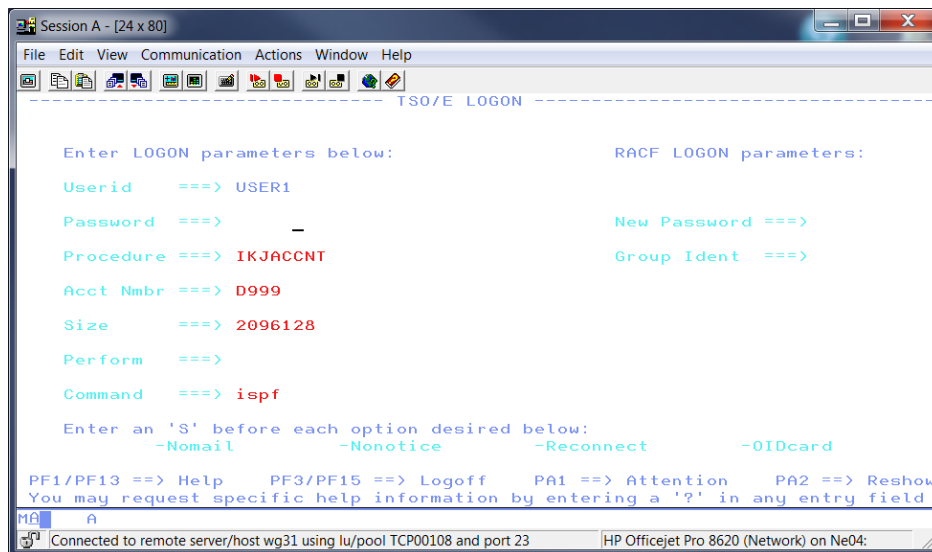
The instructions in this exercise will reference keyboard keys using a **bold** font. In the beginning explicit instructions regarding pressing keyboard keys will be provided. Eventually these explicit instructions to press the *Enter* key for example will not be included. If you are told to enter command then assume the appropriate *Enter* key should be pressed to have the command invoked or executed. Information which must be entered on a screen or panel will be in ***bold italics***. References to text on a screen or panel will be in normal *italics*.

**Tech-Tip:** Different 3270-terminal emulators will display an icon similar to the *Personal Communications*®'s icon below at the bottom of the screen when the keyboard is locked. If this occurs use the left **Ctrl** key to reset or free the keyboard.



In this emulator the **Pause** key is mapped to the clear function. If your laptop has a **Pause** key use it to clear the screen. For newer laptops without a **Pause** key, use the key sequence **Fn-P** to clear the screen. If none of these works, try a **Break** key or an **Alt-C** key sequence.

3. On the *TSO/E LOGON* panel, enter the password supplied by the instructors.



**Tech-Tip:** An **ISPF** or **PDF** command at the command prompt (===>) on this screen will automatically start ISPF. The copyright information displayed in Step 5 can be bypassed by entering ***ISPF NOLOGO*** or ***PDF NOLOGO*** at the command prompt.

**Tech-Tip:** If for some reason you are disconnected from your TSO session and cannot log in because your TSO session is still active, you can enter an **S** beside **-Reconnect** near the bottom of the panel to reconnect to your session.

4. In a TSO session whenever you see the string **\*\*\* (three asterisk)** appear (as below) as the last line in any terminal output, there is more output is waiting to be displayed. Press the **Enter** key when you are ready to see this additional output. Also remember that notification messages such as jobs completing, etc. will not be displayed unless the **Enter** key is pressed

```

Session A - [24 x 80]
File Edit View Communication Actions Window Help
ICH70001I USER1 LAST ACCESS AT 07:57:08 ON MONDAY, OCTOBER 31, 2016
IKJ56455I USER1 LOGON IN PROGRESS AT 08:03:54 ON OCTOBER 31, 2016
IKJ56951I NO BROADCAST MESSAGES

*** z/OS 02.01.00 - JES2 2.1 - DFSMS/MVS 2.01 - RACF 7790 - VTAM 6.2
*** SYSRES=Z01RS1(4408) - IODFDEV=B321 - LOADxx=LOADC1
*** SYSPLEX=WSPLEX - SYSNAME=WG31 - NODE=WG31 - CLPA=YES
*** Today's date is October 31, 2016
*** System IPLed on October 29, 2016 and has been up 2 days

PDF
***
MA A
Connected to remote server/host wg31 using lu/pool TCP00113 and port 23

```

5. You should now be at the main ISPF panel (see below). Press the **Enter** key to dismiss the *Copyright* statement.

```

Session A - [24 x 80]
File Edit View Communication Actions Window Help
Menu Utilities Compilers Options Status Help
-----
ISPF Primary Option Menu
Option ==> 3.4

0 Settings      Terminal and user parameters      User ID . : USER1
1 View         Display source data or listings    Time . . : 08:05
2 Edit         Create or change source data       Terminal. : 3278
3 Utilities     Perform utility functions          Screen . . : 1
4 Foreground   Interactive language processing    Language. : ENGLISH
5 Batch        Submit job for language processing  Appl ID . : ISR
6 Command      Enter TSO or Workstation commands   TSO logon : IKJACCNT
7 Dialog Test  Perform dialog testing              TSO prefix: USER1
8 LM Facility  Library administrator functions    System ID : WG31
9 IBM Products IBM program development products MVS acct. : D999
                                           Release . : ISPF 7.1

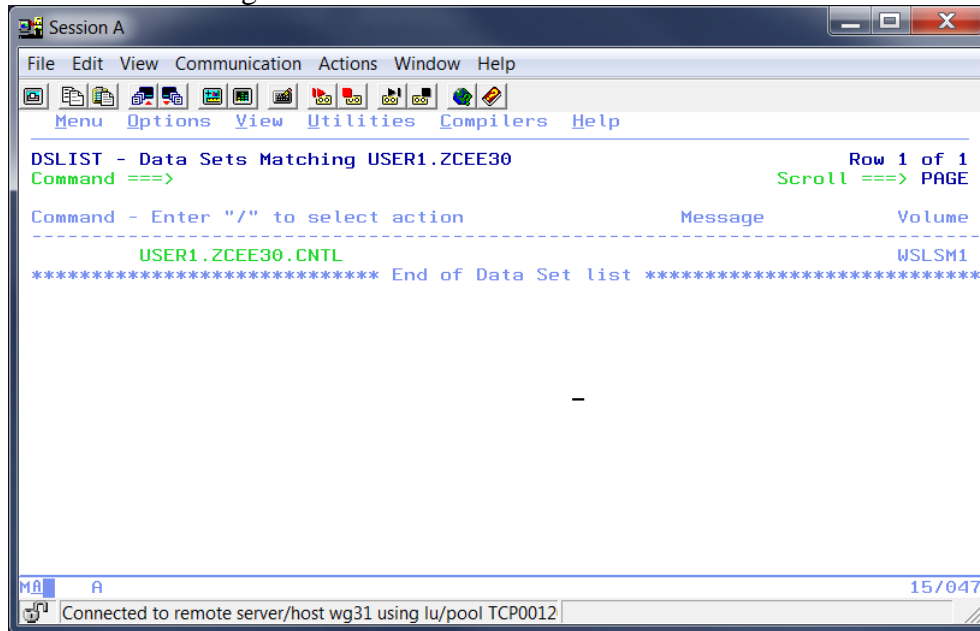
Licensed Materials - Property of IBM
5650-Z0S Copyright IBM Corp. 1980, 2013.
US Government Users Restricted Rights -
Use, duplication or disclosure restricted
by GSA ADP Schedule Contract with IBM Corp.

*ISR@PRI
MA A
Connected to remote server/host wg31 using lu/pool TCP00113 and port 23

```

6. Enter ISPF command **3.4** in the area after the *Option==>* prompt in the upper left (see above) and press the **Enter** key to display the *Data Set List Utility* panel.

7. On the *Data Set List Utility* panel, enter **USER1.ZCEE30** in the area beside *DSNAME level*, and press the **Enter** key to display a list of data sets whose names begin with *USER1.ZCEE30*. You should see something like the list below:



8. Enter **E** (for "edit") next to the data set *USER1.ZCEE30.CNTL*, and press **Enter**. You should see the a list of members in the partitioned data set.



9. Enter **B** (for "browse") next to member **ZCEERCF1**, and press **Enter**: You should see a job with several RACF commands (see below):

```

ADDGROUP LIBGRP OMVS(AUTOGID) OWNER(SYS1)

ADDUSER LIBANGE DFLTGRP(LIBGRP) OMVS(AUTOUID HOME(/u/libange/) -
  PROGRAM(/bin/sh)) NAME('LIBERTY ANGEL') NOPASSWORD NOOIDCARD
ADDUSER LIBSERV DFLTGRP(LIBGRP) OMVS(AUTOUID HOME(/u/libserv/) -
  PROGRAM(/bin/sh)) NAME('LIBERTY SERVER')
ALTUSER LIBSERV PASSWORD(LIBSERV) NOEXPIRED

RDEFINE STARTED BAQSTR.* UACC(NONE) -
  STDATA(USER(LIBSERV) GROUP(LIBGRP) -
  PRIVILEGED(NO) TRUSTED(NO) TRACE(YES))
RDEFINE STARTED BBGZANGL.* UACC(NONE) -
  STDATA(USER(LIBANGE) GROUP(LIBGRP) -
  PRIVILEGED(NO) TRUSTED(NO) TRACE(YES))
SETROPTS RACLIST(STARTED) REFRESH

RDEFINE SURROGAT BPX.SRV.LIBSERV
RDEFINE SURROGAT LIBSERV.SUBMIT
PERMIT BPX.SRV.LIBSERV CLASS(SURROGAT) ID(ZCEEADM) ACC(READ)
PERMIT LIBSERV.SUBMIT CLASS(SURROGAT) ID(ZCEEADM) ACC(READ)
SETROPTS RACLIST(SURROGAT) REFRESH

```

**Tech-Tip:** This job creates a RACF group and IDs for the Angel process and z/OS Connect EE V3.0 servers. It then it creates the two STARTED task profiles with these IDs assigned to these started tasks. Finally, the RACF resources that allows members of a z/OS Connect administrators group (ZCEEADM) to act as a surrogate of user LIBSERV and/or submit jobs as LIBSERV are defined.

10. Enter command **SUBMIT** in the area after the command prompt (*Command ==>*) at the top of the screen, and press the **Enter** key to submit this job for execution. You should get a message indicating the job has been submitted, along with the three asterisks indicating that additional output is being held before being displayed:

```

IKJ56250I JOB ZCEERCF1(JOB00060) SUBMITTED

```

11. Press the **Enter** to display the additional output. You should then see either a message indicating the job has completed with MAXCC=0000 (which is good) or a redisplay of the ISPF browse panel. If the latter, keep pressing **Enter** until the job completes.

```

08.19.07 JOB00060 $HASP165 ZCEERCF1 ENDED AT WG31 MAXCC=0000 CN(INTERNAL)
***

```

12. Press the **F3** key to return to the list of members.

13. Enter **B** next to the **ZCEERCF2** member and press the **Enter** key. This is another set of RACF commands that define the RACF *SERVER* resources which allow the use of various z/OS authorized services. *WP102604 Getting Started Guide* describes these commands in more detail. **Submit** this job for execution (see Step 10 above). Allow this job to complete before continuing.

```
RDEFINE SERVER BBG.ANGEL UACC(NONE) OWNER(SYS1)
PERMIT BBG.ANGEL CLASS(SERVER) ACCESS(READ) ID(LIBSERV)
RDEFINE SERVER BBG.AUTHMOD.BBGZSAFM UACC(NONE) OWNER(SYS1)
PERMIT BBG.AUTHMOD.BBGZSAFM -
  CLASS(SERVER) ACCESS(READ) ID(LIBSERV)
RDEFINE SERVER BBG.AUTHMOD.BBGZSAFM.SAFCRED UACC(NONE)
PERMIT BBG.AUTHMOD.BBGZSAFM.SAFCRED -
  CLASS(SERVER) ACCESS(READ) ID(LIBSERV)
RDEFINE SERVER BBG.AUTHMOD.BBGZSAFM.ZOSWLM UACC(NONE)
PERMIT BBG.AUTHMOD.BBGZSAFM.ZOSWLM -
  CLASS(SERVER) ACCESS(READ) ID(LIBSERV)
RDEFINE SERVER BBG.AUTHMOD.BBGZSAFM.TXRRS UACC(NONE)
PERMIT BBG.AUTHMOD.BBGZSAFM.TXRRS -
  CLASS(SERVER) ACCESS(READ) ID(LIBSERV)
RDEFINE SERVER BBG.AUTHMOD.BBGZSAFM.ZOSDUMP UACC(NONE)
PERMIT BBG.AUTHMOD.BBGZSAFM.ZOSDUMP -
  CLASS(SERVER) ACCESS(READ) ID(LIBSERV)
RDEFINE SERVER BBG.SECPF.BBGZDFLT UACC(NONE)
PERMIT BBG.SECPF.BBGZDFLT -
  CLASS(SERVER) ACCESS(READ) ID(LIBSERV)
RDEFINE SERVER BBG.AUTHMOD.BBGZSAFM.WOLA UACC(NONE) OWNER(SYS1)
PERMIT BBG.AUTHMOD.BBGZSAFM.WOLA -
  CLASS(SERVER) ACCESS(READ) ID(LIBSERV)
RDEFINE SERVER BBG.AUTHMOD.BBGZSAFM.LOCALCOM UACC(NONE) OWNER(SYS1)
PERMIT BBG.AUTHMOD.BBGZSAFM.LOCALCOM -
  CLASS(SERVER) ACCESS(READ) ID(LIBSERV)
RDEFINE SERVER BBG.AUTHMOD.BBGZSCFM UACC(NONE) OWNER(SYS1)
PERMIT BBG.AUTHMOD.BBGZSCFM -
  CLASS(SERVER) ACCESS(READ) ID(LIBSERV)
RDEFINE SERVER BBG.AUTHMOD.BBGZSCFM.WOLA UACC(NONE) OWNER(SYS1)
PERMIT BBG.AUTHMOD.BBGZSCFM.WOLA -
  CLASS(SERVER) ACCESS(READ) ID(LIBSERV)
PERMIT BBG.AUTHMOD.BBGZSCFM.WOLA -
  CLASS(SERVER) ACCESS(READ) ID(LIBSERV)
RDEFINE SERVER BBG.AUTHMOD.BBGZSAFM.PRODMGR UACC(NONE) OWNER(SYS1)
PERMIT BBG.AUTHMOD.BBGZSAFM.PRODMGR -
  CLASS(SERVER) ACCESS(READ) ID(LIBSERV)
RDEFINE SERVER BBG.AUTHMOD.BBGZSAFM.ZOSAIO UACC(NONE) OWNER(SYS1)
PERMIT BBG.AUTHMOD.BBGZSAFM.ZOSAIO -
  CLASS(SERVER) ACCESS(READ) ID(LIBSERV)
SETROPTS RACLIST(SERVER) REFRESH
```

14. Press **F3** to exit the browse session.

## Summary

You just created a set of essential SAF profiles for z/OS Connect EE V3.0 to use. These are detailed in the *WP102604 Getting Started Guide*. The process was streamlined for this lab by coding them in a job so submitting this job would create what was needed.

## Create a z/OS Connect EE 3.0 server

z/OS Connect EE V3.0 was installed using SMP/E ahead of the workshop and the `zconsetup` command has been invoked to create a symbolic link to the `/var/zosconnect/v3r0/extensions` directory. This directory contains property files which provide information required for locating z/OS Connect EE features and executables by Liberty at runtime.

Now it's time to create a z/OS Connect server. This is done by executing a relatively simple shell script (`zosconnect`) provided with z/OS Connect EE. Remember that a z/OS Connect server runs in a Liberty runtime, so occasionally there will be references to Liberty.

**Tech-Tip:** The OMVS directories and files created by this script must be owned by the same RACF identity associated with the z/OS Connect started task (e.g. BAQSTRT). This is problematic since this RACF identity is restricted and cannot be used to logon to TSO or submit jobs to create these directories and files.

There are various ways to ensure this ownership is set correctly. One is to create the directories and files invoking the OMVS commands using your regular RACF identity. Then use the Unix System Services (USS) command `chown` (change owner) to change the ownership of the directories and files to the RACF identity of the started task. Another way is to use Telnet clients like PuTTY or TeraTerm to access OMVS using the restricted RACF identity and then invoking OMVS commands with the restricted identity.

This exercise uses a mixture of all these techniques. In a z/OS OMVS shell, the USS command `su` (switch user) is used to switch the current OMVS RACF identity to the RACF identity of the started task before invoking any commands, or it uses the `chown` command to change ownership when executing commands in a MVS batch job. PuTTY is used start an OMVS shell with the restricted identity to update the z/OS Connect server configuration file with USS `cp` (copy) commands. All of these techniques are mostly interchangeable and are shown to demonstrate the options available.

1. In the existing 3270-terminal session, enter TSO command **OMVS** (e.g. **TSO OMVS**) at the command prompt and press **Enter** to start an ISPF OMVS shell session. You should see the screen below.

```

Session A - [24 x 80]
File Edit View Communication Actions Window Help
IBM
Licensed Material - Property of IBM
5650-ZOS Copyright IBM Corp. 1993, 2013
(C) Copyright Mortice Kern Systems, Inc., 1985, 1996.
(C) Copyright Software Development Group, University of Waterloo, 1989.
U.S. Government Users Restricted Rights -
Use, duplication or disclosure restricted by
GSA ADP Schedule Contract with IBM Corp.
IBM is a registered trademark of the IBM Corp.
USER1: /u/user1:
==> su -s libserv_
ESC=€ 1=Help 2=SubCmd 3=HlpRetrn 4=Top 5=Bottom 6=TSO 7=BackScr 8=Scroll 9=NextSess 10=Refresh 11=FwdRetr 12=Retrieve
Mf A 21/020
Connected to remote server/host wg31 using lu/pool TCP00104 and port 23

```

- \_\_\_2. At the prompt, enter OMVS command **`su -s libserv`** (see above) and press the 3270-terminal session's **Enter** key. Note that *libserv* is the RACF identity created and associated with the *BAQSTRT* started task when you ran the **ZCEERCF1** job in the previous section.

**Tech-Tip:** *LIBSERV* is the RACF identity created by the *ADDUSER* command executed when job *ZCEERCF1* job was executed in the previous section. The *ALTUSER* command in the same job set the password for *LIBSERV* to *LIBSERV*.

The ability to switch to this RACF identity without requiring a password in the *su* command was granted by defining the *BPX.SRV.LIBSERV* surrogate resource to RACF and then permitting user *USER1* read access to this surrogate resource in *ZCEERCF1*.

It is very important that this command or alternatives be considered when creating a z/OS Connect EE (zCEE) server, otherwise the server may not start or perform correctly. The RACF identity associated with the zCEE started task, e.g. *LIBSERV* and must be either be the owner the configuration directories and files or explicitly given read/write access to them in *the /var/zosconnect/server/myServer* directory structure.

N.B. Instructions to press the **Enter** key will be omitted in subsequent steps; pressing **Enter** should be assumed whenever a command, script, etc. is to be executed.

- \_\_\_3. When you are logged on, enter the **`id`** command to confirm the user ID (*uid*) and group ID (*gid*) values are for *LIBSERV* and *LIBGRP*.

```
USER1:/u/user1:> su -s libserv
$ id
uid=200019(LIBSERV) gid=200017(LIBGRP)
$
```

- \_\_\_4. The z/OS Connect *zosconnect* shell script needs to be able to locate the Java executables. Use following command in the OMVS session to *export* the environment variable *JAVA\_HOME*. This environment variable identifies the directory containing the location of the Java binaries.

**`export JAVA_HOME=/usr/lpp/java/J8.0_64`**

- \_\_\_5. Change to the directory containing the z/OS Connection script *zosconnect* using an OMVS *cd* command:

**`cd /usr/lpp/IBM/zosconnect/v3r0/bin`**

**Tech-Tip:** If you every have any questions about which directory your session is current using use the *pwd* (print working directory) to display the current directory path.

- \_\_\_6. Export environment variable (*WLP\_USER\_DIR*) to identify the directory where z/OS Connect EE V3.0 server configurations will reside.

**`export WLP_USER_DIR=/var/zosconnect`**

**Tech-Tip:** The *WLP\_USER\_DIR* will be exported in the startup JCL. The value used in the JCL must be the same as the value used when the server was created.

\_\_\_7. Invoke the `zosconnect` script to create the z/OS Connect server named *myServer*:

```
./zosconnect create myServer --template=zosconnect:default
```

You should see the following in response.

```
Server myServer created.
```

**Tech-Tip:** Other templates which can be specified with the `zosconnect create` command are `zosconnect:apiRequester`, `zosconnect:sampleCicsIpicCatalogManager`, `zosconnect:sampleWolaCatalogManager`, `zosconnect:sampleImsPhonebook` and `imsmobile:imsdefault`.

**Tech-Tip:** Below is an example of creating a server using these same commands in an MVS batch job. Again, this job runs under the authority of RACF identity LIBSERV by using RACF SURROGAT resources. If a surrogate is not used, then the ownership of the directories and files created by running this job must be changed so they are owned by the user associated with the server's started task.

```
//ZCEESRVR JOB 'ZCEE Create',CLASS=A,REGION=0M,NOTIFY=&SYSUID,USER=LIBSERV
//*****
//* Step ZCEESRVR - Use the zocconnect command to create a server
//*****
//ZCEESRVR EXEC PGM=IKJEFT01,REGION=0M
//SYSERR DD SYSOUT=*
//STDOUT DD SYSOUT=*
//STDENV DD *
ZCEEPATH=/usr/lpp/IBM/zosconnect/v3r0
PATH=/usr/lpp/java/J8.0_64/bin:$PATH
JAVA_HOME=/usr/lpp/java/J8.0_64
WLP_USER_DIR=/var/zosconnect
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
BPXBATCH SH $ZCEEPATH/bin/zosconnect create myServer +
--template=zosconnect:default +
1 > /tmp/zceeStd.out 2> /tmp/zceeStd.err
//*****
//* Step COPY - Copy the zocconnect command output to the job log
//*****
//COPY EXEC PGM=IKJEFT01,DYNAMNBR=300
//SYSTSPRT DD SYSOUT=*
//ZCEEOUT DD PATH='/tmp/zceeStd.out',PATHDISP=(DELETE,DELETE)
//ZCEEERR DD PATH='/tmp/zceeStd.err',PATHDISP=(DELETE,DELETE)
//STDOUT DD SYSOUT=*,DCB=(LRECL=1000,RECFM=V)
//STDERR DD SYSOUT=*,DCB=(LRECL=1000,RECFM=V)
//SYSPRINT DD SYSOUT=*
//SYSTSIN DD *
OCOPY INDD(ZCEEERR) OUTDD(STDERR)
OCOPY INDD(ZCEEOUT) OUTDD(STDOUT)
```

- \_\_\_8. Enter the following OMVS command to display the contents of the `/var/zosconnect/servers` subdirectory:

```
ls -al /var/zosconnect/servers
```

You should see:

```
$ ls -al /var/zosconnect/servers
total 96
drwxr-x--T   6 LIBSERV  SYS1      8192 Oct 31 11:57 .
drwxrwxrwx   4 BAGWELL  SYS1      8192 Oct 29 15:17 ..
drwxr-x--T   3 LIBSERV  SYS1      8192 Oct 29 15:17 .classCache
drwxr-x---   2 LIBSERV  SYS1      8192 Oct 31 11:57 .logs
drwxr-x---   3 LIBSERV  SYS1      8192 Oct 31 11:57 myServer
```

The `myServer` directory was created by `zosconnect` command in the previous step.

- \_\_\_9. Now display the contents of the `/var/zosconnect/servers/myServer` directory by entering OMVS command:

```
ls -al /var/zosconnect/servers/myServer
```

You should see:

```
$ ls -al /var/zosconnect/servers/myServer
total 66
drwxr-x---   3 LIBSERV  SYS1      8192 Oct 31 11:57 .
drwxr-x--T   6 LIBSERV  SYS1      8192 Oct 31 11:57 ..
drwxr-x---   4 LIBSERV  SYS1      8192 Aug  1 16:23 resources
-rw-r-----   1 LIBSERV  SYS1         25 Oct 31 11:57 server.env
-rw-r-----   1 LIBSERV  SYS1        442 Oct 31 11:57 server.xml
drwxr-x---   3 LIBSERV  SYS1      8192 Oct 31 11:57 workarea
```

The "server.xml" file is the key configuration file for the Liberty z/OS server.

- \_\_\_10. Directory `/wasetc/zc3lab` contains xml files whose contents are Liberty and z/OS Connect configuration elements. This and subsequent exercises will have steps where these files will need to be included in the `server.xml` file in order to add features, functions, etc. There are advantages for accessing these *include files* directly from directory `/var/zosconnect/servers/myServer` using a symbolic link. A symbolic link can be defined by entering this OMSV command.

```
ln -s /wasetc/zc3lab /var/zosconnect/servers/myServer/includes
```

A `ls -al` command would now show the existence of this symbolic link.

```
drwxr-x---   5 LIBSERV  SYS1      8192 Apr  2 09:58 .
drwxrwxrwx   4 JOHNSON  SYS1      8192 Apr  1 11:34 ..
lrwxrwxrwx   1 USER1    SYS1        14 Apr  2 09:58 includes -> /wasetc/zc3lab
drwxrwxrwx   4 LIBSERV  SYS1      8192 Apr  1 14:25 logs
drwxr-x---   4 LIBSERV  SYS1      8192 Apr  1 11:35 resources
-rw-r-----   1 LIBSERV  SYS1        67 Apr  1 11:34 server.env
-rw-r-----   1 LIBSERV  SYS1     2805 Apr  1 13:21 server.xml
drwxr-x---   5 LIBSERV  SYS1      8192 Apr  1 14:25 workarea
```

- \_\_\_11. Terminate the OMVS session by entering the *exit* command twice to redisplay an ISPF panel. The first exit terminates the *libserv* session, and the second exit terminates the *user1* session.

**Tech-Tip:** Symbolic links have been created from a single directory to each of the server configuration directories on this image in order to simplify administration. Directory */var/zcee* was created and the command *ln -s /var/zosconnect/servers/myServer /var/zcee/myServer* was entered at an OMVS prompt to create this symbolic linkage for this server. Start an OMVS shell and enter command *ls -al /var/zcee* at the command prompt. You should see something like what is shown below:

```
lrwxrwxrwx  1 USER1    SYS1          32 Jan 31 12:43 myServer ->
/var/zosconnect/servers/myServer
lrwxrwxrwx  1 JOHNSON  SYS1          36 Sep 10 08:40 wlpoidop ->
/var/ats/zosconnect/servers/wlpoidop
lrwxrwxrwx  1 JOHNSON  SYS1          36 Sep 10 08:39 wlpoidrp ->
/var/ats/zosconnect/servers/wlpoidrp
lrwxrwxrwx  1 JOHNSON  SYS1          14 Jan 15 2019 zc3lab -> /wasetc/zc3lab
lrwxrwxrwx  1 JOHNSON  SYS1          36 Jan 15 2019 zceeapir ->
/var/ats/zosconnect/servers/zceeapir
lrwxrwxrwx  1 JOHNSON  SYS1          39 Jan 15 2019 zceecics ->
var/cicsts/zosconnect/servers/zceecics
lrwxrwxrwx  1 JOHNSON  SYS1          25 Jan 15 2019 zceehats ->
/var/wlp/servers/zceehats
lrwxrwxrwx  1 JOHNSON  SYS1          36 Sep 12 11:14 zceeopid ->
/var/ats/zosconnect/servers/zceeopid
lrwxrwxrwx  1 JOHNSON  SYS1          36 Jan 15 2019 zceesrv1 ->
/var/ats/zosconnect/servers/zceesrv1
lrwxrwxrwx  1 JOHNSON  SYS1          36 Jan 15 2019 zceesrv2 ->
/var/ats/zosconnect/servers/zceesrv2
lrwxrwxrwx  1 JOHNSON  SYS1          36 Jul  2 2019 zceesrv3 ->
/var/ats/zosconnect/servers/zceesrv3
```

Notice that the use of symbolic links provides a single directory for accessing the different Liberty servers configured on this LPAR. These servers were configured with 4 different values for the *WLP\_USER\_DIR* environment variable, */var/zosconnect*, */var/ats/zosconnect*, */var/wlp* and */var/cicsts/zosconnect*. There is also a symbolic link to a common or shared directory */var/zcee/zc3lab* was also created for containing common configuration elements.

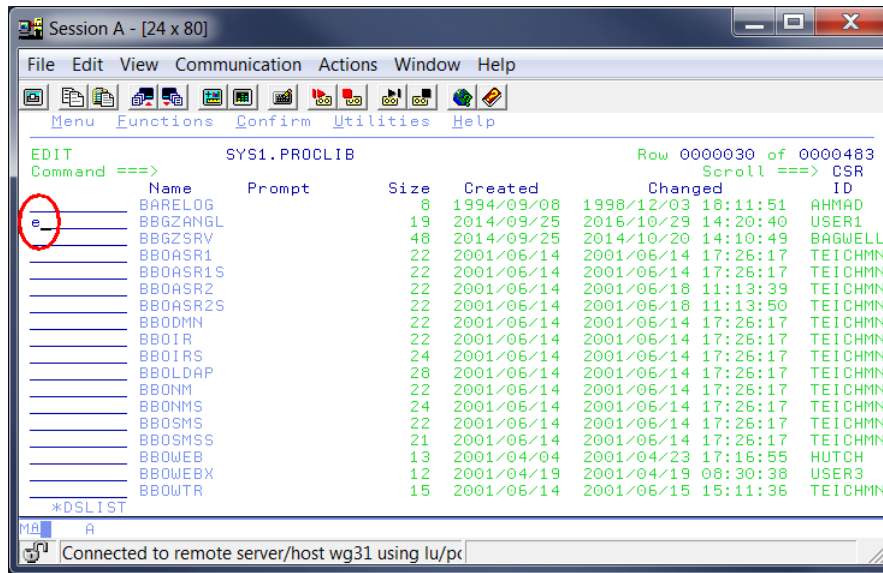
## Summary

You just created a server, but right now it's just a skeleton without any service definitions or deployed APIs. Next you will customize the JCL start procedures for the Angel process and the server.



## Customize the JCL start procedures and start the server

- \_\_\_1. In your 3270-terminal session, enter ISPF command =3.4 in the command prompt area and press **Enter**.
- \_\_\_2. Enter **SYS1.PROCLIB** in the area beside *Dsname Level* and press **Enter**.
- \_\_\_3. Type an **E** (edit) next to the **SYS1.PROCLIB** data set and press **Enter**.
- \_\_\_4. You will now see a long list of members in the **SYS1.PROCLIB** data set. Enter command **L BBG** after the command prompt (Command ==>) and press **Enter**. Locate the **BBGZANGL** member and place an **E** (edit) next to **BBGZANGL** to open this member in edit mode so changes can be made to the member.



This JCL procedure was copied this member to **SYS1.PROCLIB** prior to the workshop. This JCL is supplied in the z/OS Connect EE V3.0 SBAQSAMP target data set.

N.B. Instructions to press the **Enter** key should be assumed in subsequent steps as appropriate in order to execute commands, etc.

- \_\_\_5. Update the BBGZANGL JCL procedure to change the value of the **SET ROOT** parameter on line 3 to the location of the Liberty code shipped with z/OS Connect, e.g. **/usr/lpp/IBM/zosconnect/v3r0/wlp**.

```
// SET ROOT='/usr/lpp/IBM/zosconnect/v3r0/wlp'
```

```
***** ***** Top of Data *****
000001 //BBGZANGL PROC PARMS=",COLD=N, NAME="
000002 //*-----
000003 // SET ROOT='/usr/lpp/IBM/zosconnect/v3r0/wlp'
000004 //*-----
```



- \_\_\_6. Press **F3** to save the member and exit the edit session.
- \_\_\_7. Scroll up (**F7**) and locate member **BAQSTRT**. Place an **E** next to the member in order to open the member so changes can be made to the member.

This is the JCL procedure used to start the z/OS Connect EE V3.0 server. This JCL was copied to *SYS1.PROCLIB* prior to the workshop as well.

You are going to make three changes to this JCL procedure. The first is to replace the first line so the **PARMS=** specifies your server's name<sup>1</sup>.

- \_\_\_8. Clear line 1 by placing your cursor at the start of the line (on the first slash in column 1):

```
000001 //BAQSTRT  PROC PARMS='defaultServer'
000002 //*
000003 //* Licensed Materials - Property of IBM
```

- \_\_\_9. Then press the **End** key. That should clear the line of all characters.

- \_\_\_10. Copy/Paste the JCL statement below on line 1 by using **Ctrl-C** of the text from the copy/paste file and then selecting *Edit → Paste* or **Ctrl-V** in the 3270-terminal session.

```
//BAQSTRT  PROC PARMS='myServer'
```

The result should be:

```
000001 //BAQSTRT  PROC PARMS='myServer'
000002 //*
000003 //* Licensed Materials - Property of IBM
```

**Important:** In the above instructions the *Edit → Paste* translates to click the *Edit* tool in the tool bar of the 3270-terminal session and then select the *Paste* option. The → arrow will be used in later steps for selecting other tools in a tool bar and selecting an option in the 3270-terminal session and other windows.

You could have just overtyped *defaultServer* with *myServer*. We chose to go with copy-and-paste to eliminate any chance of a case mis-match or typo. The field was cleared first because the original text in the line was longer than its replacement.

---

<sup>1</sup> You could start the server with `S BAQSTRT,PARMS='myServer'` and avoid updating this line in the JCL. We chose to hard-code the server name for this workshop so the start is a simple `S BAQSTRT`.

\_\_\_11. The next two lines to be updated are the `// SET ZCONHOME=` (line 38) and the `JAVA_HOME=` (line 47) environment variables. Do the following:

- Scroll down (**F8**) until you see the `SET ZCONHOME=` line:
- Place your cursor at the start of the line (the first slash):
- From the copy-and-paste file, copy the line:

```
// SET ZCONHOME='/usr/lpp/IBM/zosconnect/v3r0'
```

```
000037 /**
000038 // SET ZCONHOME='/usr/lpp/IBM/zosconnect/v3r0'
000039 /**
```

- Scroll down (**F8**) and locate the `JAVA_HOME=` line. It should be line 47:
- Place your cursor at the start of that line and press the **End** key to clear the line.
- From the copy-and-paste file, copy the line:

```
JAVA_HOME=/usr/lpp/java/J8.0_64
```

- Place your cursor at the start of that line and press the **End** key to clear the line.
- Locate the `WLP_USER_DIR=` line. It should be line 48:
- Place your cursor at the start of that line and press the **End** key to clear the line.
- From the copy-and-paste file, copy the line:

```
WLP_USER_DIR=/var/zosconnect
```

- In your 3270-terminal session, select *Edit* → *Paste* (you may need to remove a trailing > character). The result should be:

```
000045 //STDENV DD *
000046 _BPX_SHAREAS=YES
000047 JAVA_HOME=/usr/lpp/java/J8.0_64
000048 WLP_USER_DIR=/var/zosconnect
000049 #JVM_OPTIONS=<Optional JVM parameters>
```

Now you're ready to start the *Angel* process and the server.

**Tech-Tip:** An additional DD statement can be added to the JCL which will provide useful information regarding the status of the z/OS Connect. Normally detailed Liberty messages are written to an OMVS file in ASCII but if you add a DD statement for DD name MSGLOG and specify SYSOUT=\* a subset of these messages will be included in the SPOOL of the task.

\_\_\_12. Go to SDSF in your 3270-terminal session by entering ISPF command `=sdsf.da` in the command field and pressing **Enter**. The *da* stands for *display active*, and it is the z/OS facility for show running tasks and completed jobs.

\_\_\_13. Next, enter **PRE BBG\*** in the command field and press **Enter**. This sets a "prefix" so only tasks starting with string *BBG* are shown. The Angel task starts with *BBG*. After entering that command, you should see the new prefix is in effect.

\_\_\_15. Enter the MVS command ***S BBGZANGL*** after the command prompt.

**Tech-Tip:** MVS and JES2 commands can be entered from SDSF by enter a / (slash) on the command line followed by the command itself (e.g. /D T). The command results can be found in the system log. If a command is especially long then simply enter a / (slash) to display a *SDSF – System Command Extension* panel where a command can span multiple lines. When a MVS command must be entered, the instructions in these exercises will indicate that the command is a MVS command and you may enter the command at the prompt by using the / (slash) prefix or using the *SDSF – System Command Extension* panel.

\_\_\_16. Keep pressing the **Enter** key until you should see the Angel process is running.

Display   Filter   View   Print   Options   Search   Help												
-----												
SDSF DA WG31		WG31		PAG 0		CPU 0		LINE 1-1 (1)				
COMMAND INPUT ==>								SCROLL ==> CSR				
NP	JOBNAME	StepName	ProcStep	JobID	Owner	C	Pos	DP	Real	Paging	SIO	
	BBGZANGL	BBGZANGL	STEP1	STC00045	LIBANGE	NS	FE	352	0.00	0.00		

\_\_\_17. You should see ***LIBANGE*** is the ID that owns the started task. That is a result of the RACF generated by the ***ZCEERCF1*** job you ran earlier. The **STARTED** profile matched to the ***BBGZANGL*** start procedure and assigned the ID ***LIBANGE*** to this task.

Display	Filter	View	Print	Options	Search	Help						
-----												
SDSF DA WG31		WG31	PAG 0	CPU 0				LINE 1-1 (1)				
COMMAND INPUT ==>								SCROLL ==> CSR				
PREFIX=BBG* DEST=(ALL) OWNER=* SYSNAME=												
NP	JOBNAME	CPU-Time	ProcStep	SR	DP	Pos	C	Owner	Status	SysName	SPag	SCP
	BBGZANGL	0.00	STEP1		FE	NS		LIBANGE		WG31		0

\_\_\_18. Change the prefix again by entering ***PRE BAQ\**** at the command line. The ***BBGZANGL*** task no longer shows (it's still running, but the prefix no longer matches).

\_\_\_19. Enter the MVS command ***S BAQSTRT*** after the command prompt and press **Enter** until you see an active **BAQSTRT** task.

Display   Filter   View   Print   Options   Search   Help												
-----												
SDSF DA WG31		WG31		PAG 2		CPU 8		LINE 1-1 (1)				
COMMAND INPUT ==>								SCROLL ==> CSR =				
NP	JOBNAME	StepName	ProcStep	JobID	Owner	C Pos	DP	Real	Paging	SIO		
	BAQSTRT	BAQSTRT	ZCON	STC00048	LIBSERV	IN	F8	10T	0.00	21640		

You should see that ***LIBSERV*** is the ID that owns the started task. That is a result of the RACF generated by the ***ZCEERCF1*** job you ran earlier. The **BAQSTRT** started procedure matched **STARTED** profile resource and the task was assigned the ID ***LIBSERV*** to it.

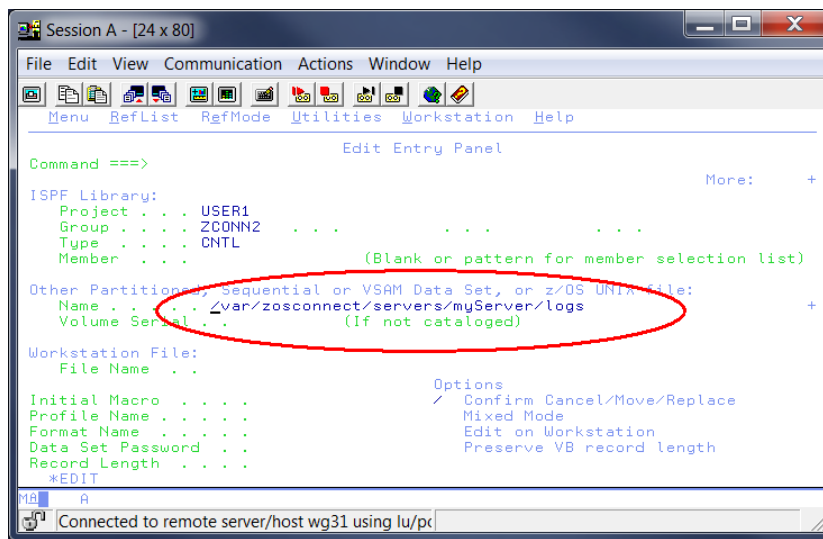
**Note:** The z/OS Connect EE server is running with *LIBSERV*'s RACF authority. This is why it was important to run the *zosconnect create* command as *LIBSERV* and why it is important in subsequent steps to use the *LIBSERV* identity when deploying Service ARhive (SAR) files. Also note that in later steps when files are included into the *server.xml* configuration file, the files that are included are owned by *LIBSERV*. All of this is done to avoid OMVS permission issues.

Both the Angel process and the z/OS Connect EE V3.0 server are started. Let's go look at the *messages.log* file for the server.

- \_\_\_ 20. Go to the ISPF Edit Entry Panel (option 2) by entering ISPF =2 on the command line and pressing **Enter**.

**Tech-Tip:** Most of the subsequent steps in this section can be performed using the IBM z/OS Explorer. If you are interested in doing these steps using this Eclipse tool, contact the instructor.

- \_\_\_ 21. Enter ***/var/zosconnect/servers/myServer/logs*** into the area beside *Name* under *Other Partitioned, Sequential or VSAM Data Set, or z/OS UNIX file:* and press **Enter**.



**Tech-Tip:** Try using the symbolic link discussed earlier, e.g. */var/zcee/myServer/logs* instead.

- \_\_\_22. This will display the contents of the OMVS directory where the z/OS Connection messages are written to file *messages.log*.

```

-----
                        z/OS UNIX Directory List                      Row 1 to 4 of 4
Command ==>                                           Scroll ==> PAGE
Time zone EST5EDT is used to calculate the displayed date and time values.
Pathname . : /SYSTEM/var/zosconnect/servers/myServer/logs
EUID . . . : 8470391
Command  Filename                                Message                Type Permission
-----
      .                                           Dir   rwxrwxrwx
      ..                                          Dir   rwxr-x---
      messages.log                               File  rw-rw-rw-
      state                                       Dir   rwxrwxrwx
***** Bottom of data *****

```

- \_\_\_23. Messages are written in ASCII so to view them in a 3270-terminal session use the **VA** (View ASCII) line command. Enter line command **VA** beside *messages.log* and press **Enter** twice.

24. You should see the *View* panel open and the *messages.log* file contents displayed in EBCDIC:

[illegible]

25. You need to scroll to the right to see more message details. Enter **100** after the command prompt (===>) and press the **F11** key.

```

File Edit Edit_Settings Menu Utilities Compilers Test Help
-----
VIEW          /SYSTEM/var/zosconnect/servers/myServer/logs/me Columns 00101 00172
Command ==>                                         Scroll ==> CSR
***** Top of Data *****
0.22.c1180320180905-2337)

WKE0001I: The server myServer has been launched.
WKB0103I: Authorized service group KERNEL is available.
WKB0103I: Authorized service group LOCALCOM is available.
WKB0103I: Authorized service group PRODMGR is available.
WKB0103I: Authorized service group SAFCRE is available.
WKB0103I: Authorized service group TXRRS is available.
WKB0103I: Authorized service group WOLA is available.
WKB0103I: Authorized service group ZOSAIO is available.
WKB0103I: Authorized service group ZOSDUMP is available.

```

Note that the **F8** key can be used to scroll forward and the **F7** key can be used to scroll backward. **F10** can be used to scroll leftward and **F11** key can be used to scroll rightward. A numeric value on the command prompt will be used to determine the number of lines to scroll forward or backward or the number of columns for scrolling leftward or rightward.

\_\_\_26. Look at the messages and note the following:

```

000012 WKB0103I: Authorized service group KERNEL is available.
000013 WKB0103I: Authorized service group LOCALCOM is available.
000014 WKB0103I: Authorized service group PRODMGR is available.
000015 WKB0103I: Authorized service group SAFCREd is available.
000016 WKB0103I: Authorized service group TXRRS is available.
000017 WKB0103I: Authorized service group WOLA is available.
000018 WKB0103I: Authorized service group ZOSAIO is available.
000019 WKB0103I: Authorized service group ZOSDUMP is available.
000020 WKB0103I: Authorized service group ZOSWLM is available.
000021 WKB0103I: Authorized service group CLIENT.WOLA is available.
000022 WKB0108I: IBM Corp product z/OS Connect version 03.00 successfully regis
000023 WKB0112I: The number of successfully registered products with z/OS is 1.
000024 WKE0002I: The kernel started after 3.05 seconds
000025 WKF0007I: Feature update started.
000026 WKS0007I: The security service is starting...
000027 NA1001I: WebSphere Dynamic Cache instance named baseCache initialized su
000028 NA1071I: The cache provider default is being used.
000029 NA1056I: Dynamic Cache (object cache) initialized successfully.
000030 WKO0229I: Native Asynchronous I/O support for z/OS has been activated.
000031 WKS4103I: Creating the LTPA keys. This may take a few seconds.
000032 WKS1123I: The collective authentication plugin with class name NullColle
000033 QR0000I: z/OS Connect Enterprise Edition version 3.0.15. (20181120-1404)
000034 WKO0219I: TCP Channel defaultHttpEndpoint has been started and is now li
000035 WKS4104A: LTPA keys created in 1.513 seconds. LTPA key file: /var/zoscon
000036 WKS4105I: LTPA configuration is ready after 1.527 seconds.
000037 WKF0015I: The server has the following interim fixes active in the runti
000038 WKF0012I: The server installed the following features: [servlet-3.1, ssl
000039 WKS0008I: Feature update completed in 5.539 seconds.
000040 WKF0011I: The server myServer is ready to run a smarter planet.
000041 VE0169I: Loading Web Module: z/OS Connect.
000042 VE0250I: Web Module z/OS Connect has been bound to default_host.
000043 WKT0016I: Web application available (default_host): http://wg31.washingt
000044 SN8501I: The session manager did not find a persistent storage location;
000045 SN0176I: A new session context will be created for application key defau
000046 SN0172I: The session manager is using the Java default SecureRandom impl
000047 NA1056I: Dynamic Cache (object cache) initialized successfully.
000048 VE0242I: [com.ibm.zosconnect] [/] [com.ibm.zosconnect.internal.web.Servi
000049 WKS9122I: For URL /* in application com.ibm.zosconnect, the following H

```

1. The "Authorized service .... is available" messages indicate the SERVER profiles (ZCEERCF2 job) are in effect for this server and the ID under which it runs. Of particular interest is SAFCREd (for later when we use SAF for security).
2. Version 3.0.15.0 (20181120-1404) indicates the maintenance made available in November 20th of 2018 is in effect. The values you see may be different.
3. The server is started and ready to run.

25. Use the **F3** key to go back to the *Edit Entry Panel* and use the space bar or **Backspace** key to remove the */logs* subdirectory from the directory name field. Press **Enter** to display the contents of */var/zosconnect/servers/myServer*. You should see:

```

Menu  Utilities  View  Options  Help
-----
                                z/OS UNIX Directory List
                                Row 1 to 8 of 8
Command ===>                                Scroll ===> PAGE
Time zone EST5EDT is used to calculate the displayed date and time values.
Pathname . : /SYSTEM/var/zosconnect/servers/myServer
EUID . . . : 8470391
Command  Filename                                Message                                Type  Permission
-----
      .                                           Dir    rwxr-x---
      ..                                          Dir    rwxr-x--T
      logs                                       Dir    rwxrwxrwx
      resources                                 Dir    rwxrwxrwx
      server.env                               File   rw-r-----
      server.xml                               File   rw-r-----
      workarea                                  Dir    rwxr-x---
***** Bottom of data *****

```

26. Use the **VA** (View ASCII) line command to open the *server.xml* file in browse mode. You should see:

```

000001 <?xml version="1.0" encoding="UTF-8"?>
000002 <server description="new server">
000003
000004     <!-- Enable features -->
000005     <featureManager>
000006         <feature>zoscconnect:zosConnect-2.0</feature>
000007         <feature>zoscconnect:zosConnectCommands-1.0</feature>
000008     </featureManager>
000009
000010     <!-- To access this server from a remote client add a host attribute
000011     <httpEndpoint id="defaultHttpEndpoint"
000012         host="*"
000013         httpPort="9080"
000014         httpsPort="9443" />
000015
000016     <!-- add cors to allow cross origin access, e.g. when using swagger
000017     <cors id="defaultCORSConfig"
000018         domain="/"
000019         allowedOrigins="*"
000020         allowedMethods="GET, POST, PUT, DELETE, OPTIONS"
000021         allowedHeaders="Origin, Content-Type, Authorization"
000022         allowCredentials="true"
000023         maxAge="3600" />
000024
000025     <!-- NOTE: Disabling automatic polling for changes to configuration fil
000026     deployed services and APIs is a prudent option for z/OS Connect EE
000027     Polling might be convenient for iterative development and test
000028     systems, but not for production.
000029
000030     Configuration elements that can drive significant polling activity
000031     default are specified below to explicitly disable automatic polling.
000032     Further element types to consider for polling interval include
000033     zoscconnect_zosConnectDataXform (default 2 seconds) and
000034     keyStore (default 500 milliseconds).

```



```

000035
000036 Consider setting the updateTrigger attribute to "polled" if changes
000037 to associated resources need to be picked up automatically, and tune
000038 the polling interval accordingly. The attribute that controls polling
000039 frequency for each of these elements is included, together with its
000040 associated default value.
000041 -->
000042
000043 <!-- config requires updateTrigger="mbean" for REFRESH command support
000044 <config updateTrigger="mbean" monitorInterval="500"/>
000045
000046 <!-- zosConnect APIs -->
000047 <zoscconnect_zosConnectAPIs updateTrigger="disabled" pollingRate="5s"
000048
000049 <!-- zosConnect Services -->
000050 <zoscconnect_services updateTrigger="disabled" pollingRate="5s"/>
000051
000052 <!-- applicationMonitor is not applicable for zCEE servers -->
000053 <applicationMonitor updateTrigger="disabled" dropinsEnabled="false"/>
000054
000055 </server>

```

That is a *very* minimal server.xml file created from the default template. Other templates will generate server.xml files based on their parameters.

In the next step we will have you copy in a new server.xml file that will add quite a few more XML elements. The dynamic nature of Liberty will incorporate the new configuration without requiring a server restart.

**Tech-Tip:** Note that by default the *updateTrigger* attributes are disabled for APIs and services. These will need to be enabled if auto update is desired.

27. Add the following include after the `<server description=<new server">` line.

```
<include location="${server.config.dir}/includes/basicSecurity.xml"/>
```

**Tech-Tip:** This include element is taking advantage of an environment variable (*server.config.dir*) whose value is set to the directory containing the *server.xml* file used to initialize this instance of a Liberty server. The *includes* subdirectory is the symbolic link that references directory */wasetc/zc3lab* (the location of the include files shared among multiple servers). The value of using environment variables and symbolic links in the server.xml makes this configuration element reusable across multiple instances of servers and multiple OMVS directory configurations.

- \_\_\_28. Including this file (see below) adds a new feature to the server, *appSecurity-2.0*, which enables application security and some of the other elements required for basic security.

```
<server description="basic security">

  <!-- Enable features -->
  <featureManager>
    <feature>appSecurity-2.0</feature>
  </featureManager>

  <keyStore id="defaultKeyStore" password="Liberty"/>

  <webAppSecurity allowFailOverToBasicAuth="true" />

  <basicRegistry id="basic1" realm="zosConnect">
    <user name="Fred" password="fredpwd" />
  </basicRegistry>

  <authorization-roles id="zos.connect.access.roles">
    <security-role name="zosConnectAccess">
      <user name="Fred"/>
    </security-role>
  </authorization-roles>

</server>
```

*Figure 1: Contents of basicSecurity.xml*

"Basic" means a user is authenticated by providing a user identity and password. In this case the user registry will be provided by the Liberty server itself. For now, it's a simple way to satisfy the security requirements of z/OS Connect EE V3.0. In subsequent section of this exercise, SAF (e.g. RACF) will be implemented.

- \_\_\_29. Use the **F3** key to end the edit session.

- \_\_\_30. Adding this file using an *include* statement refreshes the z/OS Connect EE V3.0 server. Enter the following MVS command to refresh the configuration **F BAQSTRT,ZCON,REFRESH**

Next, we're going to do a preliminary test of your z/OS Connect EE V30 server, even though you do not yet have any services or APIs defined.

**Tech-Tip:** MVS commands can be entered using SDSF. In ISPF, enter the command **=SDSF.LOG** and add the / (slash) prefix to the command, e.g. **/F BAQSTRT,ZCON,REFRESH**

- \_\_\_31. When the message that the server configuration has been successfully updated appears in the *messages.log* file (see below), open the Firefox browser on your desktop.

```
CWPki0803A: SSL certificate created in 5.313 seconds. SSL key file: /var/zosconnect/servers/myServer/resources/security/key.jks
CWWKS9112A: The web application security settings have changed. The following properties were modified: allowFailOverToBasicAuth=true
CWWKS9120I: Authorization roles with id="zos.connect.access.roles" have been successfully processed
CWWKG0017I: The server configuration was successfully updated in 5.481 seconds.
CWWKO0219I: TCP Channel defaultHttpEndpoint-ssl has been started and is now listening for requests on host * (IPv4) port 9443.
```

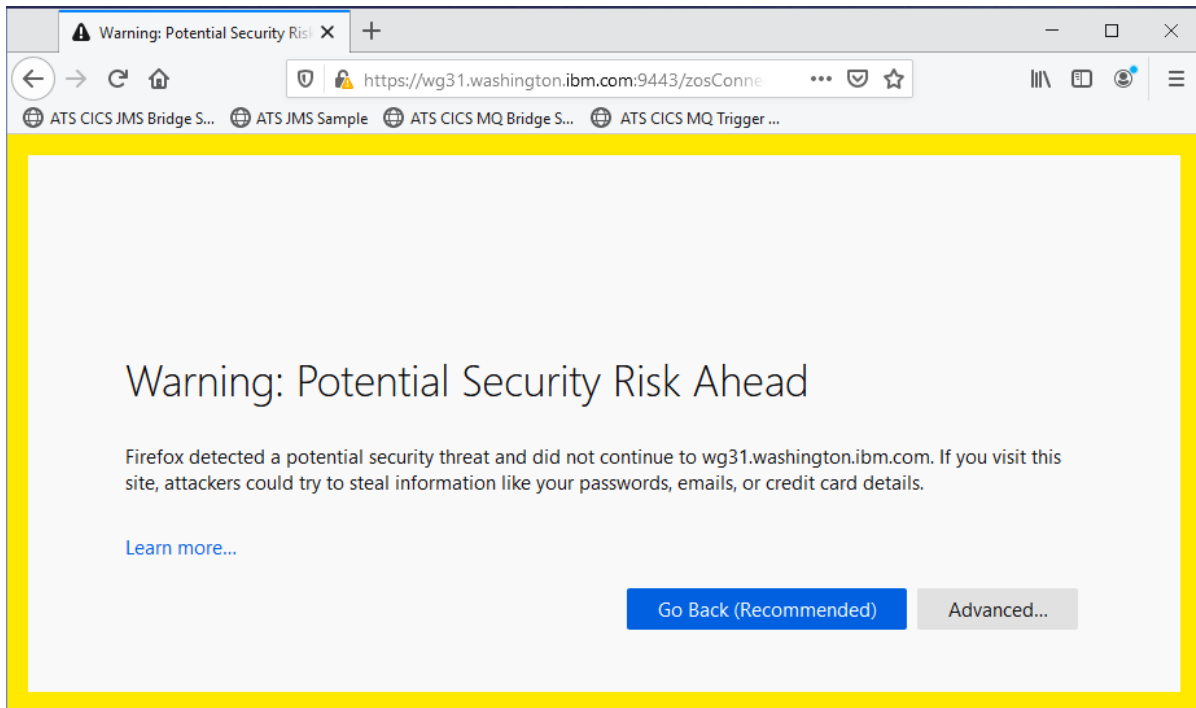
## Verify the z/OS Connect server configuration

\_\_\_1. Enter the following as the URL:

**<https://wg31.washington.ibm.com:9443/zosConnect/apis>**

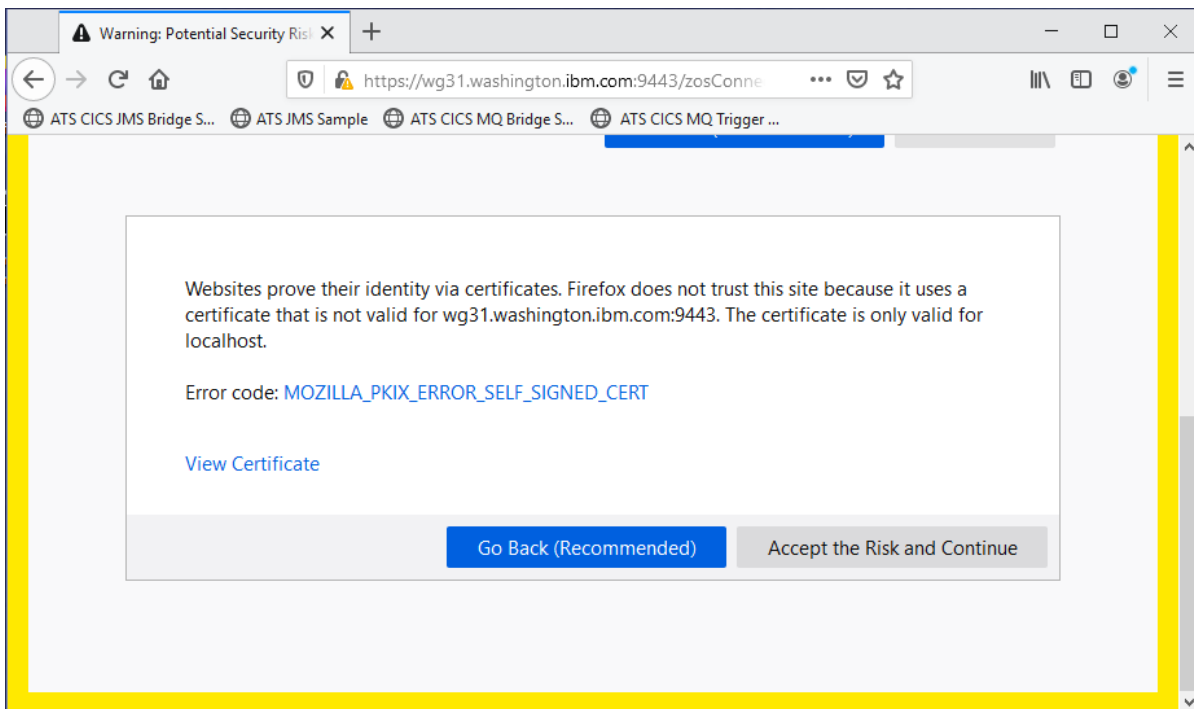
This will query the server for configured APIs. Even though you will receive a response indicating no APIs are configured, this is still a good test of connectivity to the server. But before we see the screen showing no APIs are defined, we first must address an issue where the z/OS Connect EE server is using a self-signed certificate that is not recognized by the browser, and we must authenticate to the z/OS Connect EE server with a user identity and password.

\_\_\_2. Initially you will be challenged by Firefox because the digital certificate used by the Liberty z/OS server is self-signed (recall that we're using simple basic security for the time being). Click on the **Advanced** button to continue.



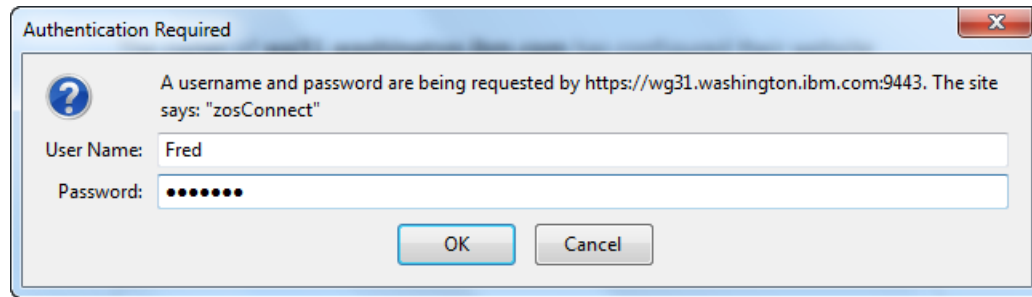
**Tech-Tip:** It is very important to access the z/OS Connect server from a browser prior to any testing using the Swagger UI. Accessing a z/OS Connect URL from a browser starts an SSL handshake between the browser and the server. If this handshake has not performed prior to performing any test, the test will fail with no message in the browser and no explanation. Ensuring this handshake has been performed is why you may be directed to access a z/OS Connect URL prior to using the Swagger UI during this exercise.

\_\_\_3. Scroll down and click the **Accept the Risk and Continue** button to continue.



**Tech-Tip:** The configuration of this Firefox browser has been configured to accept self-signed certificates and to not permanently save certificates under these circumstances. This means that these security risk screens will be displayed every time a browser is restarted until a permanent certificate is installed. This action was enabled by changing the Firefox `security.cererrors.permanentOverride` preference to false.

\_\_\_4. Next you will see a prompt you for a userid and password:



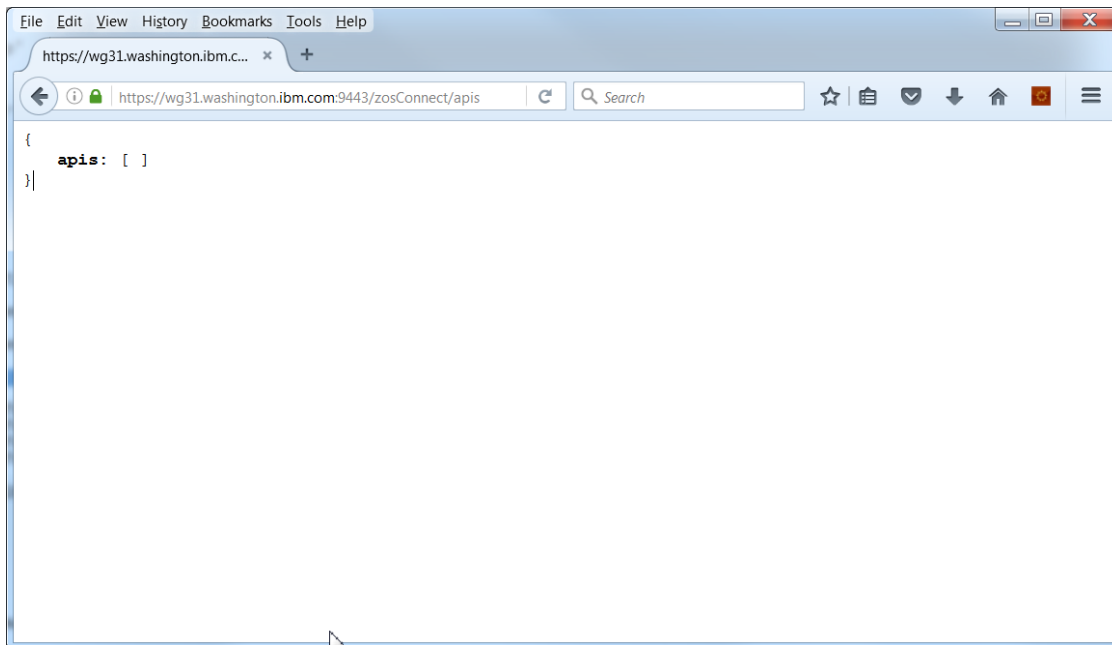
Enter a *User name* of **Fred** and a *Password* of **fredpwd** (case matters) and click **OK**.

**Note:** The ID and password is defined in the basicSecurity.xml file:

```
<basicRegistry id="basic1" realm="zosConnect">
  <user name="Fred" password="fredpwd" />
</basicRegistry>
```

Our objective for security at this point in the workshop is simplicity. That's why we're using the security elements coded in the server.xml. Later we will use RACF.

\_\_\_5. You should now see the following. That is the expected result since no APIs have been installed.



\_\_\_6. Enter the following as the URL:

<https://wg31.washington.ibm.com:9443/zosConnect/services>

That will query the server for configured services. You should also get a response indicating no services are configured. This is expected.

\_\_\_7. An alternative to using a browser, you can verify the server by using the cURL command in a *Command Prompt* session.

```
curl -X GET --user Fred:fredpwd --header "Content-Type: application/json" --insecure  
https://wg31.washington.ibm.com:9443/zosConnect/apis
```

with results:

```
{"apis":[]}
```

```
curl -X GET --user Fred:fredpwd --header "Content-Type: application/json" --insecure  
https://wg31.washington.ibm.com:9443/zosConnect/services
```

with results:

```
{"zosConnectServices":[]}
```

## Summary

You customized the JCL start procedures and started both the Angel process and the server. You have added basic security for z/OS Connect EE V3.0. Finally, you verified basic operations with the browser and optionally the cURL command.

## Deploying Services and APIs

In this section, you will deploy previously generated service archive and API archives files using z/OS Connect EE RESTful administrative interfaces. But before the services can be deployed, the features required by the services must be installed in the running server.

### Update the z/OS Connect EE Server Configuration

Before Services and APIs for the application can be deployed, the *server.xml* needs additional configuration information.

1. Edit the *server.xml* in */var/zosconnect/servers/myServer* and add the following *include* after the *<server description=<new server>* line.

```
<include location="{server.config.dir}/includes/ipic.xml"/>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<server description="new server">
<include location="/wasetc/zc3lab/basicSecurity.xml"/>
<include location="/wasetc/zc3lab/ipic.xml"/>
```

**Tech-Tip:** If a feature required by a service is not currently installed in the server, the deployment of the service will fail.

2. Including this file adds a new feature to the server, *cicsService-1.0*, which enables support for connecting to CICS regions using IP interconnectivity (IPIC). In this scenario, a CICS IPIC *TCPIPService* (see below) has been defined in the target CICS region to listen for input TCP/IP request on port 1491.

```
<server description="CICS IPIC - Catalog">

  <featureManager>
    <feature>zoscconnect:cicsService-1.0</feature>
  </featureManager>

  <zoscconnect_cicsIpicConnection id="catalog"
    host="wg31.washington.ibm.com"
    port="1491"/>

</server>
```

Figure 2: Contents of *ipic.xml*

**Tech-Tip:** Service names do not have to be explicitly identified in a *zosconnect\_services* element. Services archive files in the *services* directory will be automatically installed either a server restart or based on the setting of the *updateTrigger* property (see *server.xml* above). One reason for adding explicit definitions for services (*zosconnect\_services*) is when security attributes are required.

Note, the corresponding CICS TCPIPService definition.

```

Session A
File Edit View Communication Actions Window Help

OVERTYPE TO MODIFY                                CICS RELEASE = 0710
CEDA Alter TCPIPService( IPIC )
TCPIPService : IPIC
GRoup       : SYSPGRP
DEscription ==> DFHISAIP
Urname      ==> 01491
Portnumber  ==> 01491      1-65535
Status      ==> Open      Open | Closed
PROtocol    ==> IPic      Http | Eci | User | IPic
TRANsaction ==> CISS
Backlog     ==> 00000      0-32767
TSqprefix   :
Host        ==> ANY
(Mixed Case) ==>
Ipaddress   ==> ANY
SPecifctps ==>
SOcketclose ==> No        No | 0-240000 (HHMMSS)
MAXPersist  ==> No        No | 0-65535
MAXDataLen  ==> 000032     3-524288

SYSID=CICS APPLID=CICS53Z

PF 1 HELP 2 COM 3 END          6 CRSR 7 SBH 8 SFH 9 MSG 10 SB 11 SF 12 CNCL
MA A                           06/022
Connected to remote server/host wg31 using lu/pool TCP0010

```

- \_\_\_3. Use the **F3** key to end the edit session.
- \_\_\_4. Adding this file using an *include* statement refreshing the z/OS Connect EE V3.0 server. Enter the following MVS command to refresh the configuration **FBAQSTRT,ZCON,REFRESH**

**Tech-Tip:** MVS commands can be only entered on any Spool Search and Display Facility (SDSF) ISPF panel if they are prefixed with a slash. To enter the MVS command **D A,L** then you must enter **/D A,L** at the Command ==> prompt on the panel. The best way to review the results of the command is to use the SDSF **LOG** command to access the system log and go to the time in the log the command was entered.

An alternative is to simply enter the SDSF **/** (slash) command at the command prompt to display the System Command Extension panel (see below).

```

Session A - wg31
File Edit View Communication Actions Window Help

Display Filter View Print Options Search Help
SDSF SYSLOG 3375.101 S0W1 S0W1 09/02/2018 2W 4,986 COLUMNS 52- 131
COMMAND INPUT ==> /

0000
0001
0002
0003
0004
0005
0006
0007
0008
0009
0010
0011
0012
0013
0014
0015
0016
0017
0018
0019
0020
0021
0022
0023
0024
0025
0026
0027
0028
0029
0030
0031
0032
0033
0034
0035
0036
0037
0038
0039
0040
0041
0042
0043
0044
0045
0046
0047
0048
0049
0050
0051
0052
0053
0054
0055
0056
0057
0058
0059
0060
0061
0062
0063
0064
0065
0066
0067
0068
0069
0070
0071
0072
0073
0074
0075
0076
0077
0078
0079
0080
0081
0082
0083
0084
0085
0086
0087
0088
0089
0090
0091
0092
0093
0094
0095
0096
0097
0098
0099
0100
0101
0102
0103
0104
0105
0106
0107
0108
0109
0110
0111
0112
0113
0114
0115
0116
0117
0118
0119
0120
0121
0122
0123
0124
0125
0126
0127
0128
0129
0130
0131
0132
0133
0134
0135
0136
0137
0138
0139
0140
0141
0142
0143
0144
0145
0146
0147
0148
0149
0150
0151
0152
0153
0154
0155
0156
0157
0158
0159
0160
0161
0162
0163
0164
0165
0166
0167
0168
0169
0170
0171
0172
0173
0174
0175
0176
0177
0178
0179
0180
0181
0182
0183
0184
0185
0186
0187
0188
0189
0190
0191
0192
0193
0194
0195
0196
0197
0198
0199
0200
0201
0202
0203
0204
0205
0206
0207
0208
0209
0210
0211
0212
0213
0214
0215
0216
0217
0218
0219
0220
0221
0222
0223
0224
0225
0226
0227
0228
0229
0230
0231
0232
0233
0234
0235
0236
0237
0238
0239
0240
0241
0242
0243
0244
0245
0246
0247
0248
0249
0250
0251
0252
0253
0254
0255
0256
0257
0258
0259
0260
0261
0262
0263
0264
0265
0266
0267
0268
0269
0270
0271
0272
0273
0274
0275
0276
0277
0278
0279
0280
0281
0282
0283
0284
0285
0286
0287
0288
0289
0290
0291
0292
0293
0294
0295
0296
0297
0298
0299
0300
0301
0302
0303
0304
0305
0306
0307
0308
0309
0310
0311
0312
0313
0314
0315
0316
0317
0318
0319
0320
0321
0322
0323
0324
0325
0326
0327
0328
0329
0330
0331
0332
0333
0334
0335
0336
0337
0338
0339
0340
0341
0342
0343
0344
0345
0346
0347
0348
0349
0350
0351
0352
0353
0354
0355
0356
0357
0358
0359
0360
0361
0362
0363
0364
0365
0366
0367
0368
0369
0370
0371
0372
0373
0374
0375
0376
0377
0378
0379
0380
0381
0382
0383
0384
0385
0386
0387
0388
0389
0390
0391
0392
0393
0394
0395
0396
0397
0398
0399
0400
0401
0402
0403
0404
0405
0406
0407
0408
0409
0410
0411
0412
0413
0414
0415
0416
0417
0418
0419
0420
0421
0422
0423
0424
0425
0426
0427
0428
0429
0430
0431
0432
0433
0434
0435
0436
0437
0438
0439
0440
0441
0442
0443
0444
0445
0446
0447
0448
0449
0450
0451
0452
0453
0454
0455
0456
0457
0458
0459
0460
0461
0462
0463
0464
0465
0466
0467
0468
0469
0470
0471
0472
0473
0474
0475
0476
0477
0478
0479
0480
0481
0482
0483
0484
0485
0486
0487
0488
0489
0490
0491
0492
0493
0494
0495
0496
0497
0498
0499
0500
0501
0502
0503
0504
0505
0506
0507
0508
0509
0510
0511
0512
0513
0514
0515
0516
0517
0518
0519
0520
0521
0522
0523
0524
0525
0526
0527
0528
0529
0530
0531
0532
0533
0534
0535
0536
0537
0538
0539
0540
0541
0542
0543
0544
0545
0546
0547
0548
0549
0550
0551
0552
0553
0554
0555
0556
0557
0558
0559
0560
0561
0562
0563
0564
0565
0566
0567
0568
0569
0570
0571
0572
0573
0574
0575
0576
0577
0578
0579
0580
0581
0582
0583
0584
0585
0586
0587
0588
0589
0590
0591
0592
0593
0594
0595
0596
0597
0598
0599
0600
0601
0602
0603
0604
0605
0606
0607
0608
0609
0610
0611
0612
0613
0614
0615
0616
0617
0618
0619
0620
0621
0622
0623
0624
0625
0626
0627
0628
0629
0630
0631
0632
0633
0634
0635
0636
0637
0638
0639
0640
0641
0642
0643
0644
0645
0646
0647
0648
0649
0650
0651
0652
0653
0654
0655
0656
0657
0658
0659
0660
0661
0662
0663
0664
0665
0666
0667
0668
0669
0670
0671
0672
0673
0674
0675
0676
0677
0678
0679
0680
0681
0682
0683
0684
0685
0686
0687
0688
0689
0690
0691
0692
0693
0694
0695
0696
0697
0698
0699
0700
0701
0702
0703
0704
0705
0706
0707
0708
0709
0710
0711
0712
0713
0714
0715
0716
0717
0718
0719
0720
0721
0722
0723
0724
0725
0726
0727
0728
0729
0730
0731
0732
0733
0734
0735
0736
0737
0738
0739
0740
0741
0742
0743
0744
0745
0746
0747
0748
0749
0750
0751
0752
0753
0754
0755
0756
0757
0758
0759
0760
0761
0762
0763
0764
0765
0766
0767
0768
0769
0770
0771
0772
0773
0774
0775
0776
0777
0778
0779
0780
0781
0782
0783
0784
0785
0786
0787
0788
0789
0790
0791
0792
0793
0794
0795
0796
0797
0798
0799
0800
0801
0802
0803
0804
0805
0806
0807
0808
0809
0810
0811
0812
0813
0814
0815
0816
0817
0818
0819
0820
0821
0822
0823
0824
0825
0826
0827
0828
0829
0830
0831
0832
0833
0834
0835
0836
0837
0838
0839
0840
0841
0842
0843
0844
0845
0846
0847
0848
0849
0850
0851
0852
0853
0854
0855
0856
0857
0858
0859
0860
0861
0862
0863
0864
0865
0866
0867
0868
0869
0870
0871
0872
0873
0874
0875
0876
0877
0878
0879
0880
0881
0882
0883
0884
0885
0886
0887
0888
0889
0890
0891
0892
0893
0894
0895
0896
0897
0898
0899
0900
0901
0902
0903
0904
0905
0906
0907
0908
0909
0910
0911
0912
0913
0914
0915
0916
0917
0918
0919
0920
0921
0922
0923
0924
0925
0926
0927
0928
0929
0930
0931
0932
0933
0934
0935
0936
0937
0938
0939
0940
0941
0942
0943
0944
0945
0946
0947
0948
0949
0950
0951
0952
0953
0954
0955
0956
0957
0958
0959
0960
0961
0962
0963
0964
0965
0966
0967
0968
0969
0970
0971
0972
0973
0974
0975
0976
0977
0978
0979
0980
0981
0982
0983
0984
0985
0986
0987
0988
0989
0990
0991
0992
0993
0994
0995
0996
0997
0998
0999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276

```



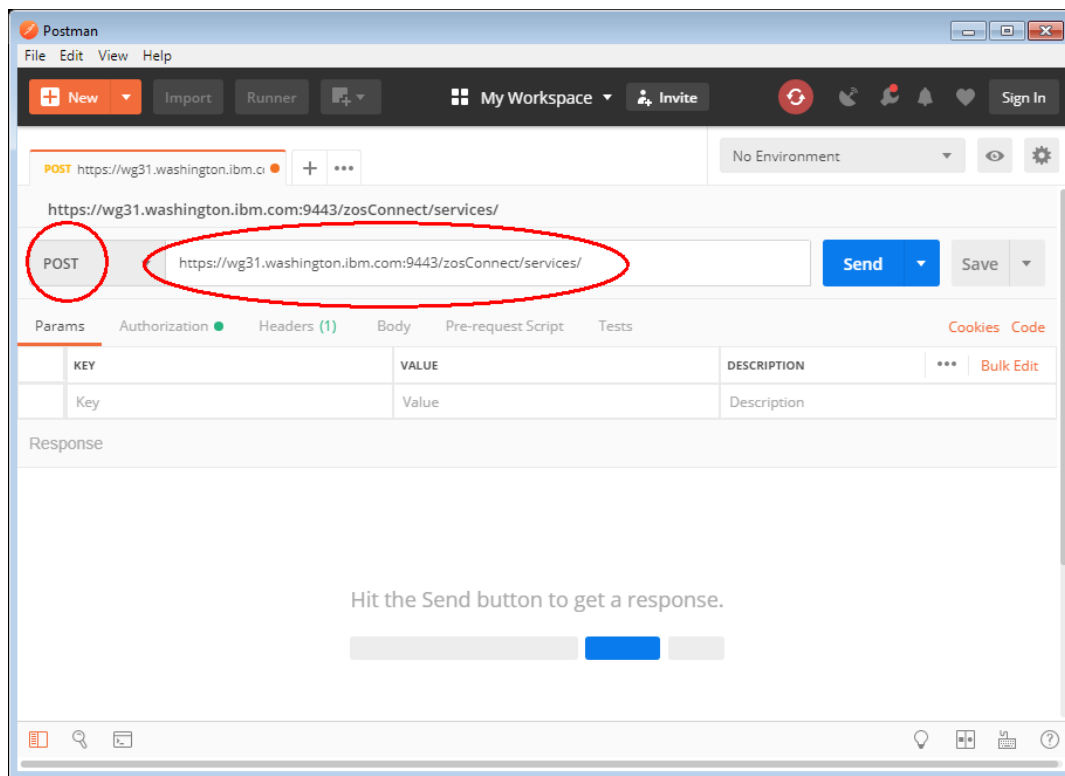
## Deploy the Services

The common application artifacts installed into z/OS connect server are services and APIs (to see how these artifacts are developed, review any of the other exercises in this workshop). A *service* describes a specific interaction with a back end subsystem whereas an API describes the RESTful interface to one or more services. Both of these artifacts are archive (e.g. zipped) files. They should be installed using the z/OS Connect EE RESTful administrative interface. Two products which seem to be most popular tools for invoking z/OS Connect EE administrative RESTful APIs are *Postman* which is available for downloading from <https://www.getpostman.com/apps> and *cURL* (*client URL*) which is available for downloading from <https://curl.haxx.se/download.html>. The use of both *Postman* and *cURL* will be shown in this section of the exercise.

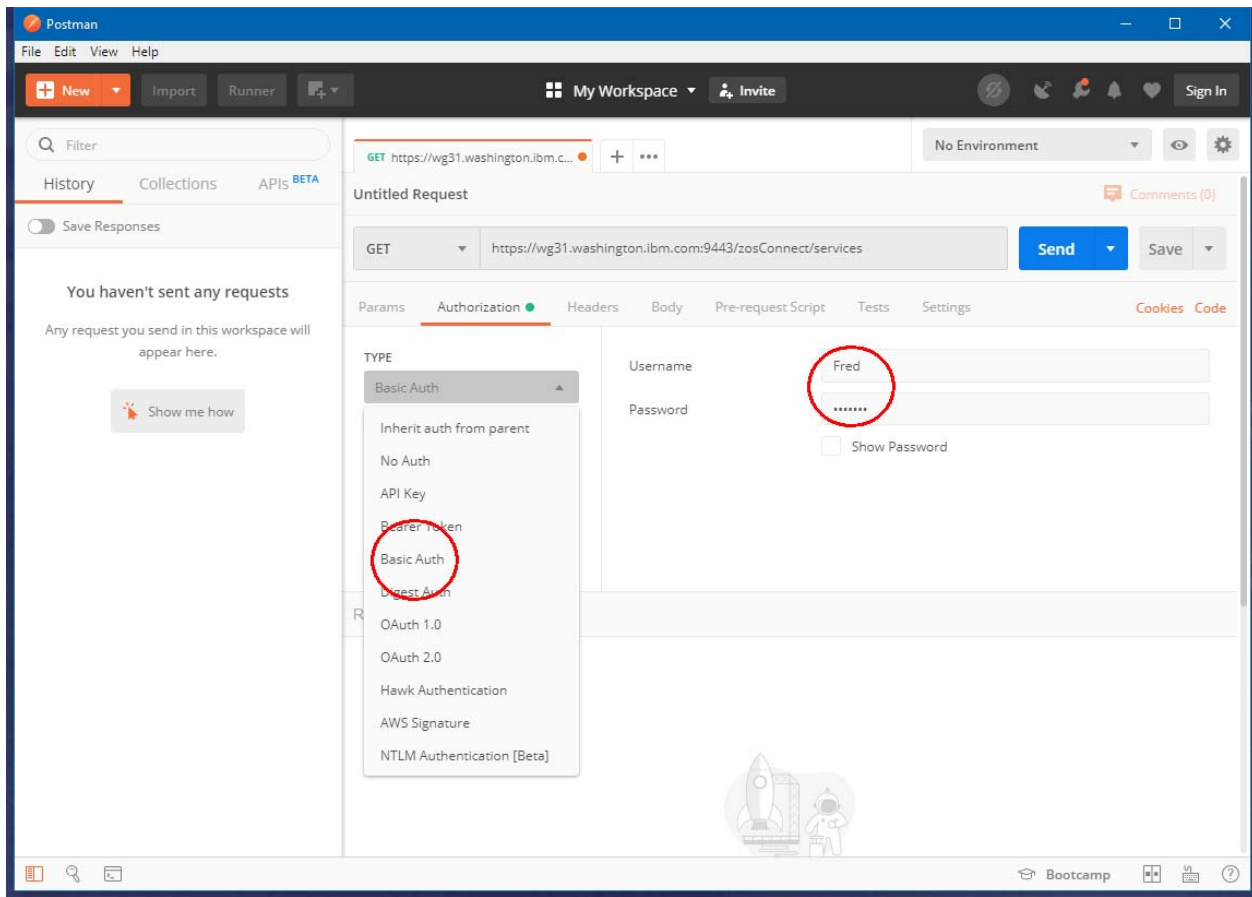
These instructions provide details on both. Choose either *Postman* or *cURL* to deploy the service.

### Using Postman

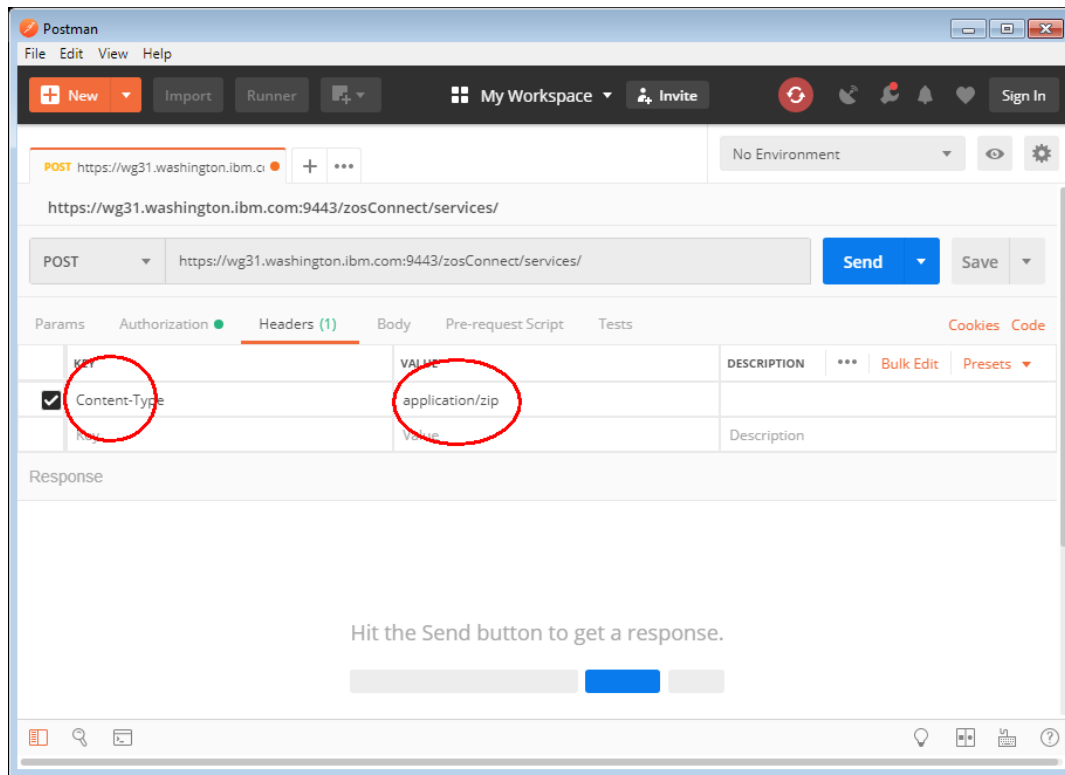
1. Open the *Postman* tool icon on the desktop. If necessary reply to any prompts and close any welcome messages, use the down arrow to select **POST** and enter <https://wg31.washington.ibm.com:9443/zosConnect/services/> in the URL area (see below).



2. Next, select the *Authorization* tab to enter an authorization identity and password. Use the pull down arrow to select *Basic Auth* and enter *Fred* as the username and *fredpwd* as the Password.

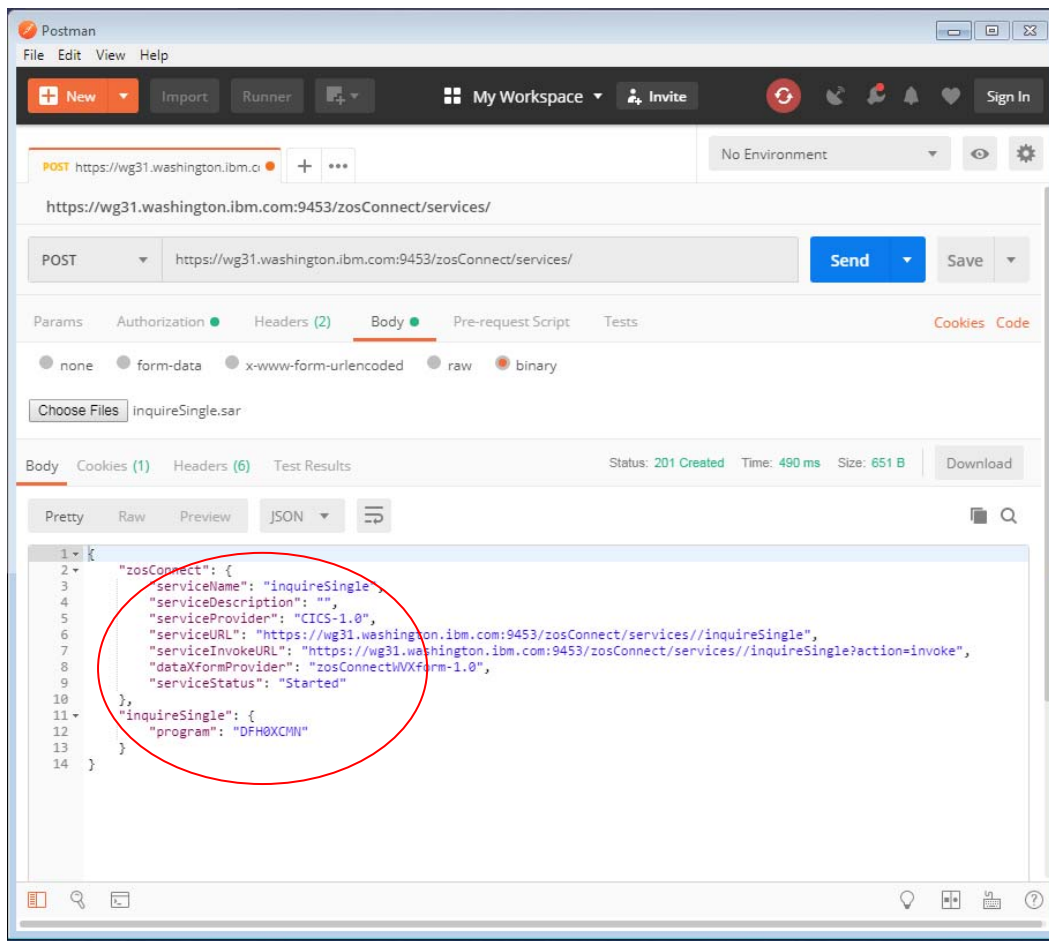


- \_\_\_3. Next, select the *Headers* tab. Under *KEY* use the code assist feature to enter ***Content-Type***, and under *VALUE*, use the code assist feature to enter ***application/zip***.

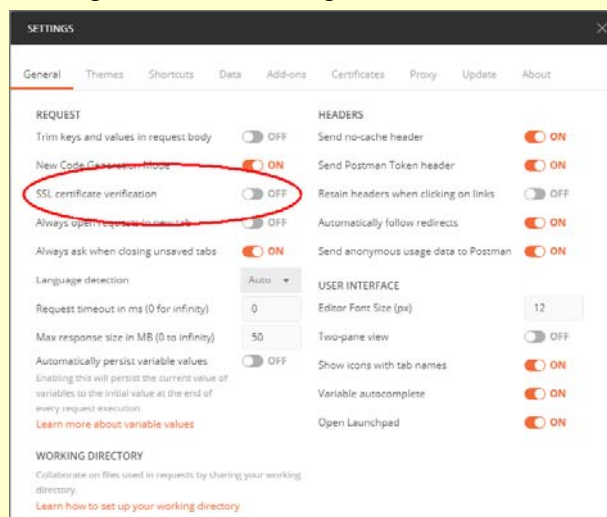


**Tech-Tip:** Code assist simply means that when text is entered in field, all the valid values for that field that match the typed text will be displayed. You can select the desired value for the field from the list displayed and that value will populate that field.

4. Next select the *Body* tab and select the *binary* radio button. Then use the **Choose Files** button to navigate to folder *c:/z/admin*. Select the *inquireSingle.Sar* file. Then press the **Send** button. A response message should come back indicating the service has been started and other details about the service.



**Tech-Tip:** The Postman settings have been changed to disable *SSL certificate verification*, a settings option.



## Using *cURL*

The *cURL* tool provides a command line interface to REST APIs. The same administrative API service invoked with *Postman* can be invoked with *cURL* as shown here.

- \_\_\_1. Use the *Command Prompt* icon on the desktop to open a DOS command prompt session.
- \_\_\_2. In the session use the change directory (cd) command to go to directory c:\z\admin, e.g.  
***cd c:\z\admin***
- \_\_\_3. Paste the command below at the command prompt and press **Enter**.

```
curl -X POST --user Fred:fredpwd --data-binary @inquireSingle.sar --header "Content-Type: application/zip" --insecure https://wg31.washington.ibm.com:9443/zosConnect/services
```

Microsoft Windows [Version 6.1.7601]

Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\workstation>***cd c:\z\admin***

```
c:\z\admin>curl -X POST --user Fred:fredpwd --header "Content-Type: application/zip"
-d @inquireSingle.json --insecure https://wg31.washington.ibm.com:9443/zosConnect/services
{"zosConnect":{"serviceName":"inquireSingle","serviceDescription":"","
"serviceProvider":"CICS-
1.0","serviceURL":"https://wg31.washington.ibm.com:9443/zosConnect/services/inquireSingle",
"serviceInvokeURL":"https://wg31.washington.ibm.com:9443/zosConnect/services/inquireSingle?action
=invoke",
"dataXformProvider":"zosConnectWVXform-1.0","serviceStatus":"Started"},
"inquireSingle":{"program":"DFH0XCMN"}}
```

**Tech-Tip:** In the above example:

***--user Fred:fredpwd*** could have been specified as ***--header "Authorization: Basic RnJlZDpmcmVkcHdk"***

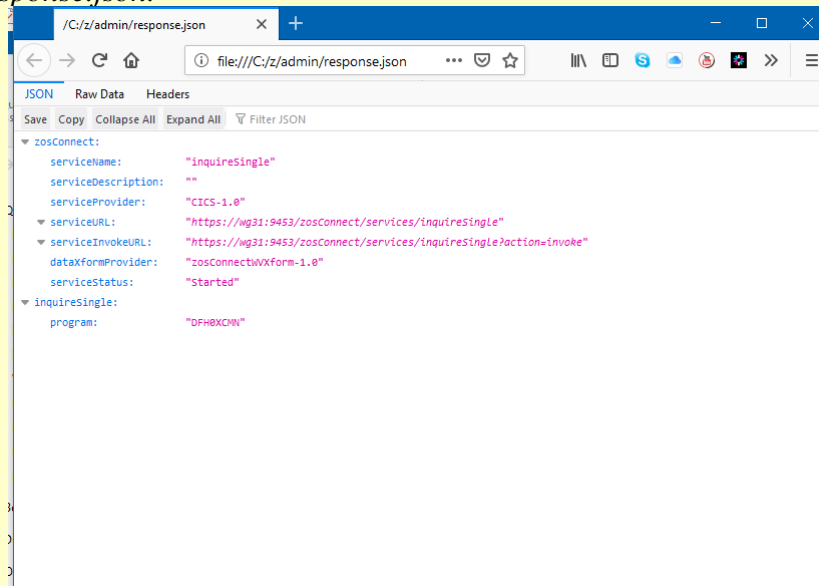
***inquireCatalog.sar*** is a file in the same directory in which the command is executed.

***--insecure*** is a *cURL* directive that tells *cURL* to ignore the self-signed certificate sent by the z/OS Connect EE server.

The text in **green** is the JSON response message.

**Tech-Tip:** Another useful cURL directive is `-o response.json`.

When this directive is used, the JSON response message is written to a file named `response.json` which then can be opened with Firefox and viewed in a more readable format, e.g. command `firefox file:///c:/z/admin/response.json`.



\_\_\_4. Use either *Postman* or *cURL* to deploy the *inquireCatalog.sar* file.

```
curl -X POST --user Fred:fredpwd --data-binary @inquireCatalog.sar --header "Content-Type: application/zip" --insecure https://wg31.washington.ibm.com:9443/zosConnect/services
```

\_\_\_5. Use either *Postman* or *cURL* to deploy the *placeOrder.sar* file.

```
curl -X POST --user Fred:fredpwd --data-binary @placeOrder.sar --header "Content-Type: application/zip" --insecure https://wg31.washington.ibm.com:9443/zosConnect/services
```

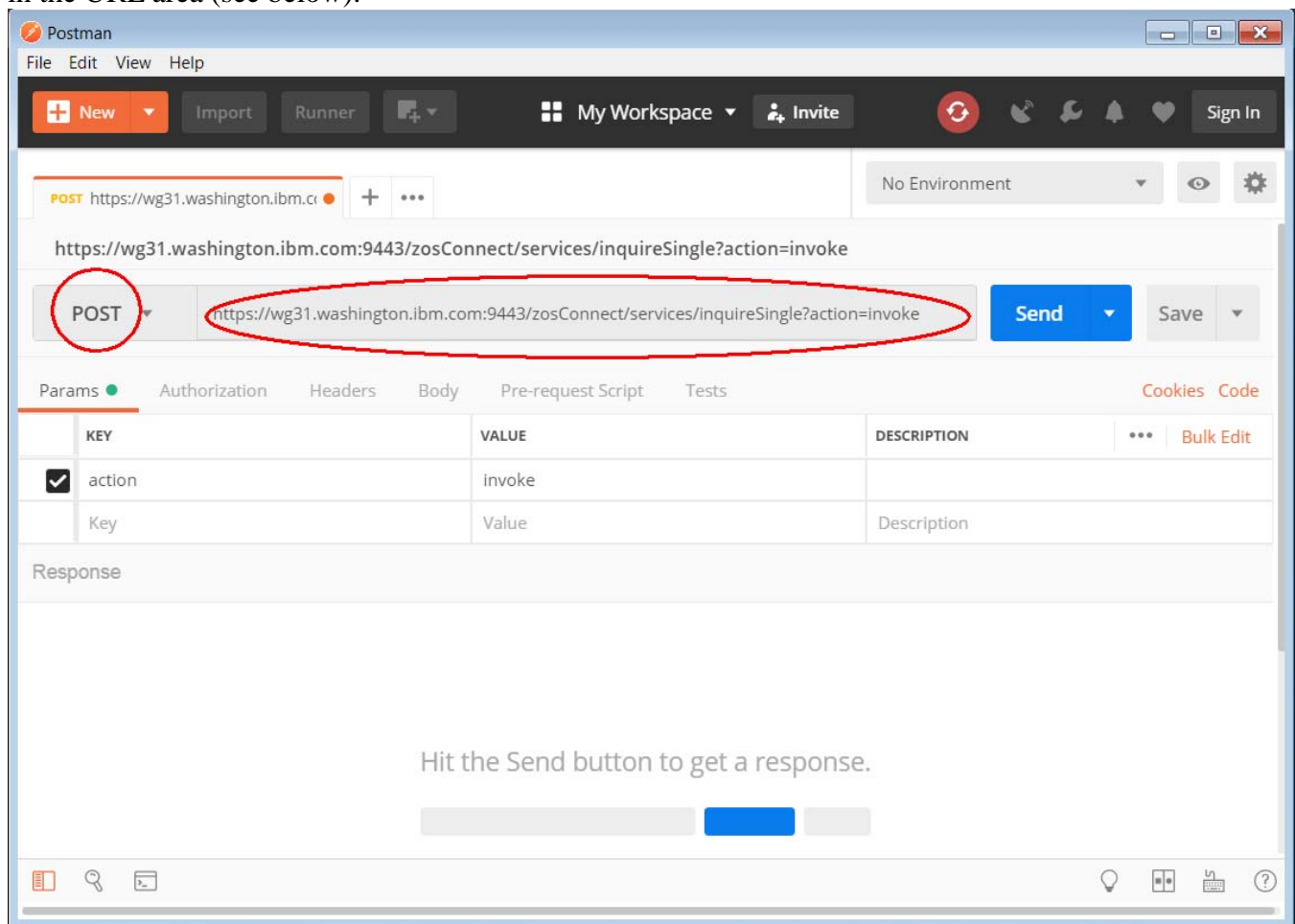
**Tech-Tip:** The only differences between the commands is the name of the service archive file being deployed.

## Test the Services

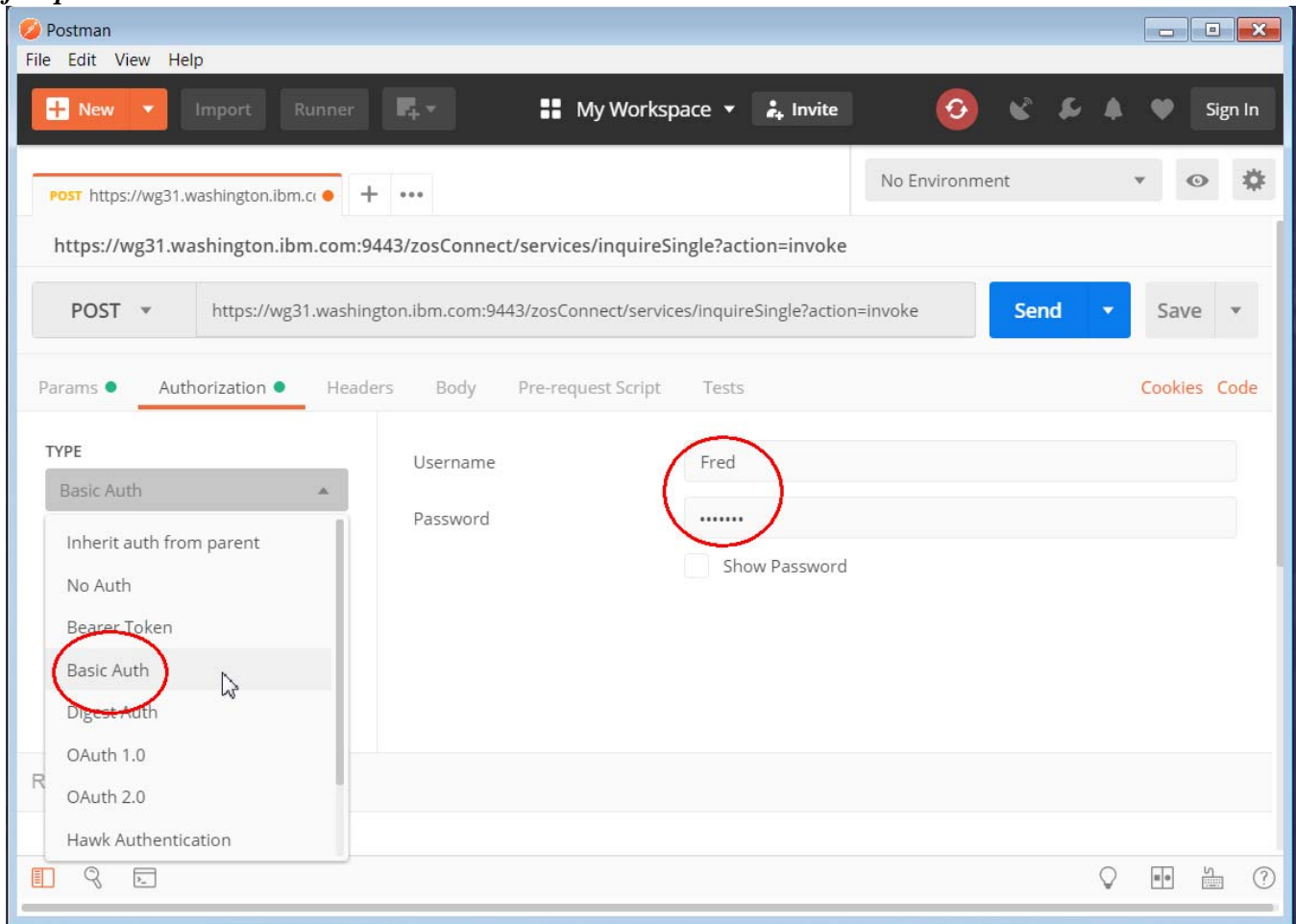
The services should be tested to ensure the infrastructure and the request and response messages are as expected.

### Using Postman

1. Open the *Postman* tool icon on the desktop and if necessary, reply to any prompts. Close any welcome messages. The use the down arrow to select **POST** and enter **<https://wg31.washington.ibm.com:9443/zosConnect/services/inquireSingle?action=invoke>** in the URL area (see below).

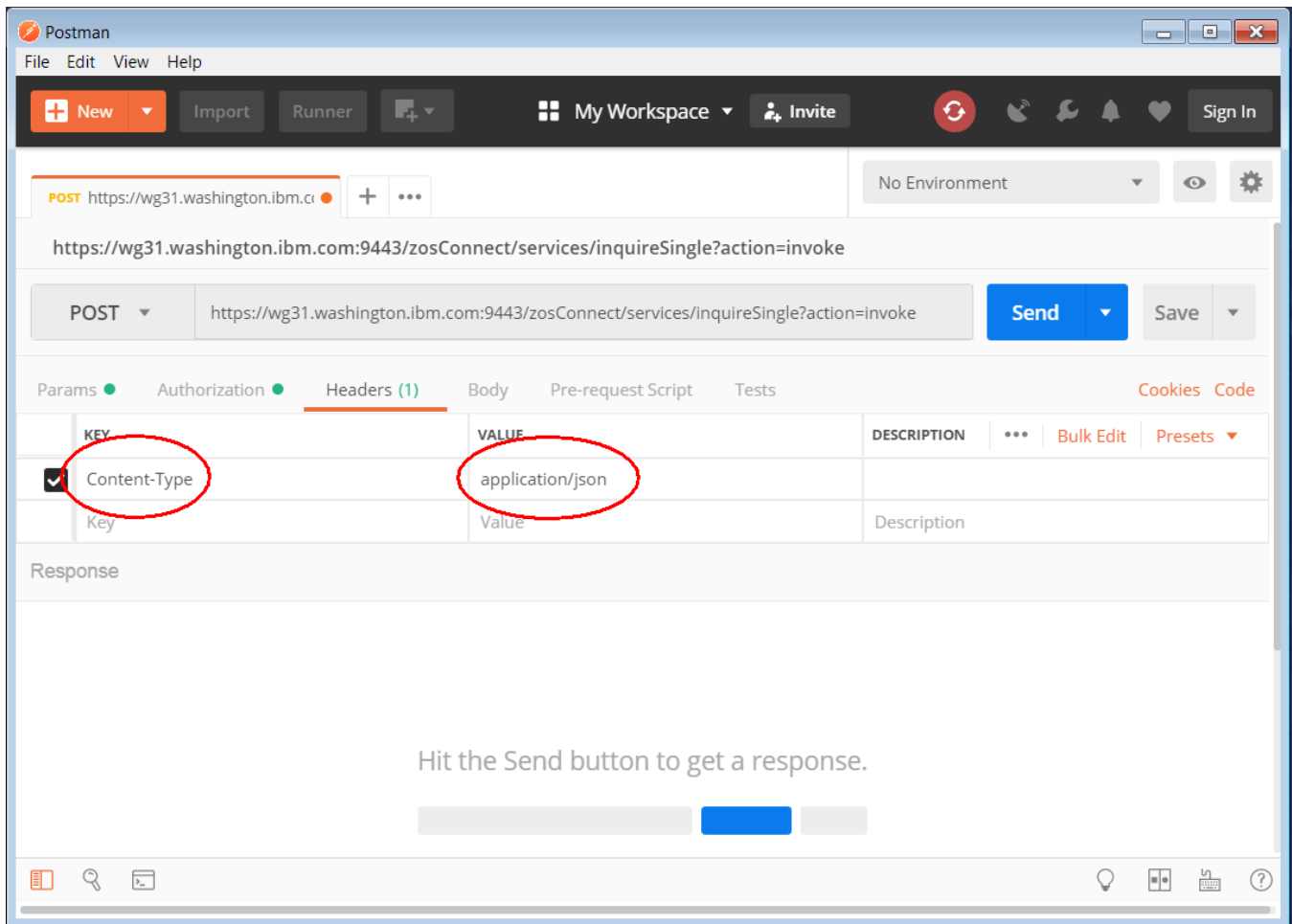


2. No *query* or *path* parameters are required so next select the *Authorization* tab to enter an authorization identity and password. Use the pull down arrow to select *Basic Auth* and enter ***Fred*** as the username and ***fredpwd*** as the Password.





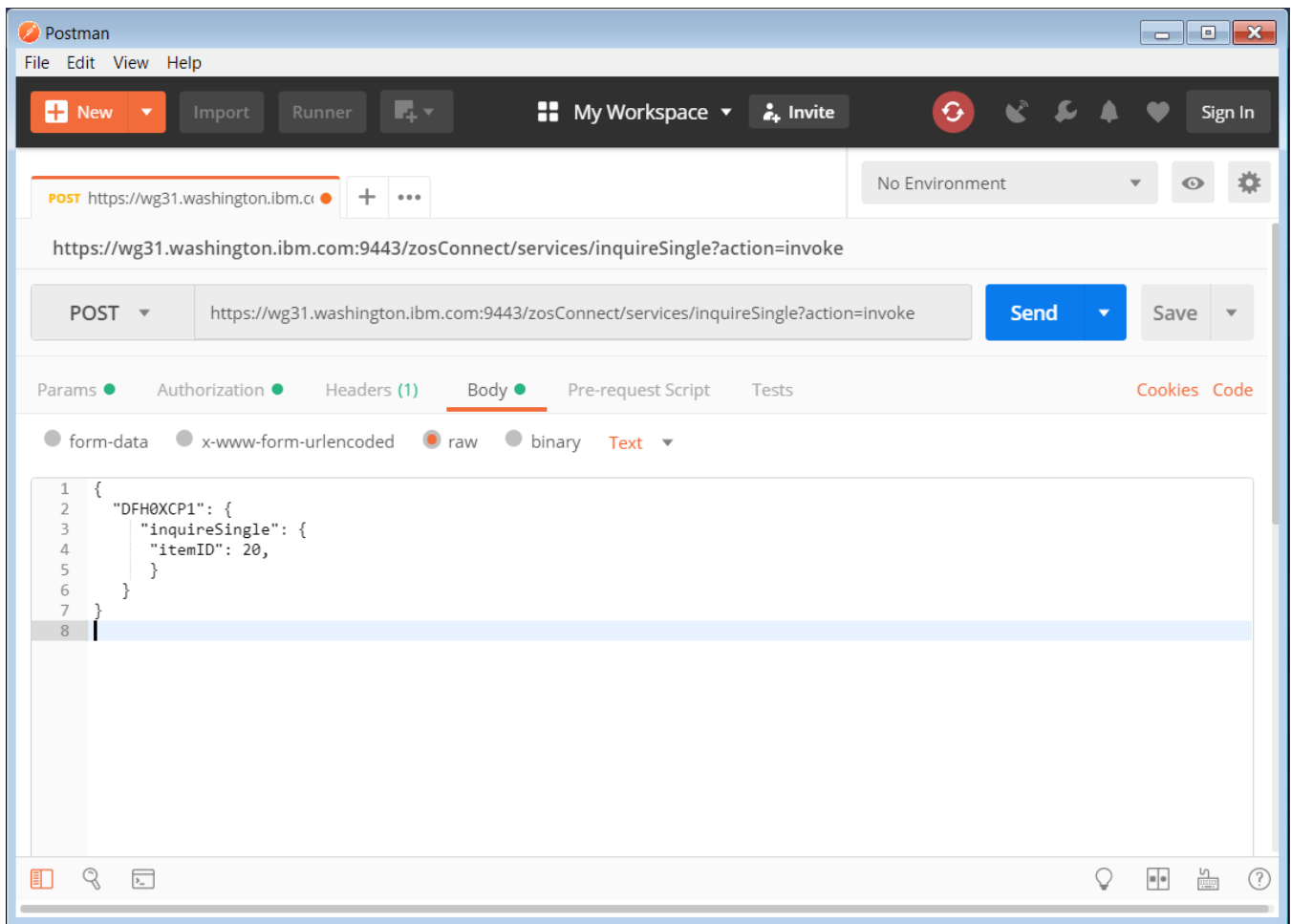
3. Next, select the *Headers* tab. Under *KEY*, use the code assist feature to enter ***Content-Type***, and under *VALUE*, use the code assist feature to enter ***application/json***.



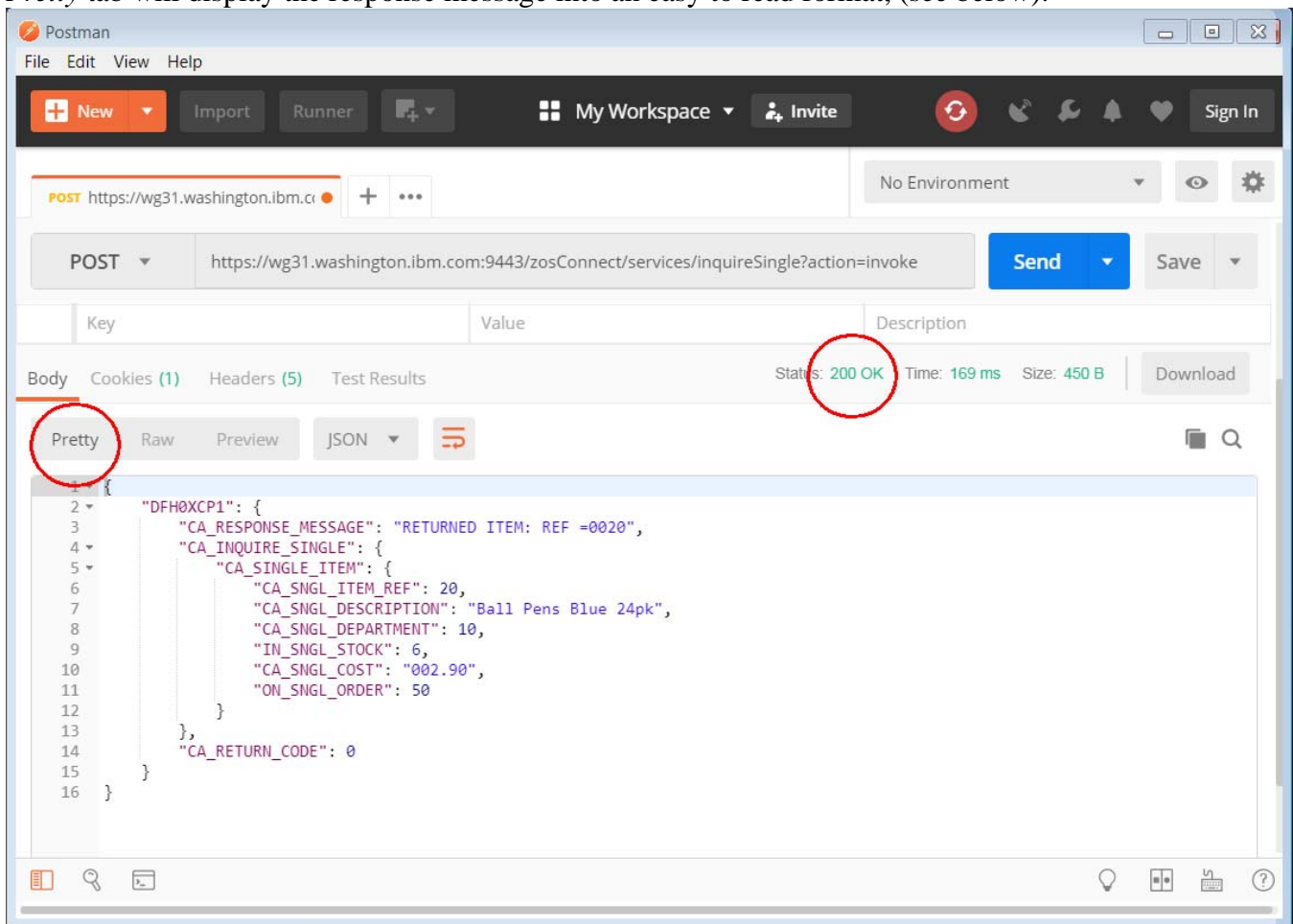
**Tech-Tip:** Code assist simply means that when text is entered in field, all the valid values for that field that match the typed text will be displayed. You can select the desired value for the field from the list displayed and that value will populate that field.

- \_\_\_4. Next select the *Body* tab and select the *raw* radio button and enter the JSON message below in the *Body* area and press the **Send** button.

```
{
  "DFH0XCP1": {
    "inquireSingle": {
      "itemID": 20,
    }
  }
}
```



- \_\_\_5. Pressing the **Send** button invokes the API. The Status of request should be 200 OK, and pressing the *Pretty* tab will display the response message into an easy to read format, (see below).



### Using cURL

The *cURL* tool provides a command line interface to REST APIs. The same service just tested with *Postman* can be tested with *cURL* as shown here.

- \_\_\_1. Use the *Command Prompt* icon on the desktop to open a DOS command prompt session.
- \_\_\_2. In the session use the change directory (cd) command to go to directory `c:\z\admin`, e.g.  
***cd c:\z\admin***

\_\_\_3. Paste the command below at the command prompt and press **Enter**.

```
curl -X POST --user Fred:fredpwd --header "Content-Type: application/json" -d @inquireSingle.json --insecure https://wg31.washington.ibm.com:9443/zosConnect/services/inquireSingle?action=invoke
```

```
jMicrosoft Windows [Version 6.1.7601]
```

```
Copyright (c) 2009 Microsoft Corporation. All rights reserved.
```

```
C:\Users\workstation>cd c:\z\admin
```

```
c:\z\admin>curl -X POST --user Fred:fredpwd --header "Content-Type: application/json"
-d @inquireSingle.json --insecure
https://wg31.washington.ibm.com:9443/zosConnect/services/inquireSingle?action=invoke
{"DFH0XCP1":{"CA_RESPONSE_MESSAGE":"RETURNED ITEM: REF
=0020","CA_INQUIRE_SINGLE":{"CA_SINGLE_ITEM":{"CA_SNGL_ITEM_REF":20,"CA_SNG
L_DESCRIPTION":"Ball Pens Blue 24pk", "CA_SNGL_DEPARTMENT":10,
"IN_SNGL_STOCK":6, "CA_SNGL_COST":"002.90", "ON_SNGL_ORDER":50}},
"CA_RETURN_CODE":0}}
```

**Tech Tip:** In the above example:

**--user Fred:fredpwd** could have been specified as **--header "Authorization: Basic RnJlZDpmcmVkcHdk"**

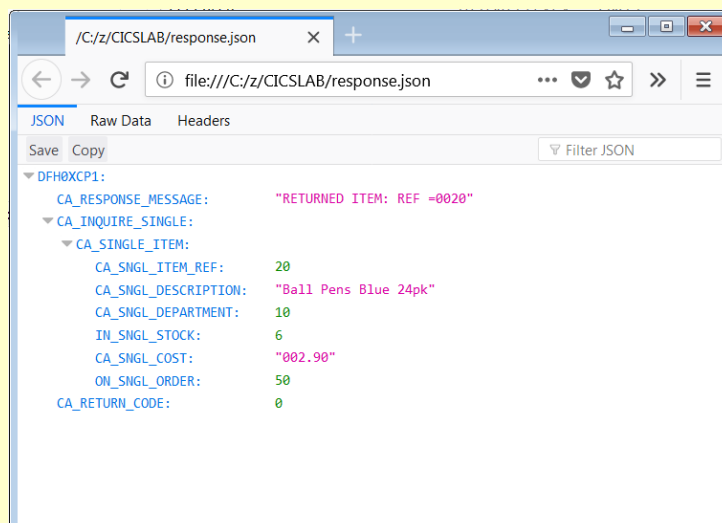
**inquireSingle.json** is a file in the same directory that contains the JSON request message.

**--insecure** is a *cURL* directive that tells *cURL* to ignore the self-signed certificate sent by the z/OS Connect EE server.

The text in **green** is the JSON response message.

**Tech-Tip:** Another useful cURL directive is `-o response.json`.

When this directive is used, the JSON response message is written to a file named `response.json` which then can be opened with Firefox and viewed in a more readable format, e.g. command `firefox file:///c:/z/admin/response.json`



4. The `inquireCatalog` service can be tested with *Postman* or *cURL* with URL

**`https://wg31.washington.ibm.com:9443/zosConnect/services/inquireCatalog?action=invoke`**

and JSON request message.

```

{
  "DFH0XCP1": {
    "inquireCatalog": {
      "startItemID": 20
    }
  }
}

```

The corresponding cURL command and results are shown below:

**`curl -X POST --user Fred:fredpwd --header "Content-Type: application/json"`**

**`-d @inquireCatalog.json --insecure`**

**`https://wg31.washington.ibm.com:9443/zosConnect/services/inquireCatalog?action=invoke`**

```
c:\z\admin>curl -X POST --user Fred:fredpwd --header "Content-Type: application/json"
-d @inquireCatalog.json --insecure https://wg31.washington.ibm.com:9443/
zosConnect/services/inquireCatalog?action=invoke
{"DFH0XCP1":{"CA_RESPONSE_MESSAGE":"+15 ITEMS RETURNED","CA_INQUIRE_REQUEST":{"C
A_LAST_ITEM_REF":150,"CA_CAT_ITEM":[{"ON_ORDER":0,"CA_ITEM_REF":10,"CA_COST":"00
2.90","IN_STOCK":135,"CA_DESCRIPTION":"Ball Pens Black 24pk","CA_DEPARTMENT":10}
,{"ON_ORDER":50,"CA_ITEM_REF":20,"CA_COST":"002.90","IN_STOCK":6,"CA_DESCRIPTION
":"Ball Pens Blue 24pk","CA_DEPARTMENT":10},{"ON_ORDER":0,"CA_ITEM_REF":30,"CA_C
OST":"002.90","IN_STOCK":106,"CA_DESCRIPTION":"Ball Pens Red 24pk","CA_DEPARTMEN
T":10},{"ON_ORDER":0,"CA_ITEM_REF":40,"CA_COST":"002.90","IN_STOCK":80,"CA_DESCR
IPTION":"Ball Pens Green 24pk","CA_DEPARTMENT":10},{"ON_ORDER":0,"CA_ITEM_REF":5
0,"CA_COST":"001.78","IN_STOCK":83,"CA_DESCRIPTION":"Pencil with eraser 12pk","C
A_DEPARTMENT":10},{"ON_ORDER":40,"CA_ITEM_REF":60,"CA_COST":"003.89","IN_STOCK":
13,"CA_DESCRIPTION":"Highlighters Assorted 5pk","CA_DEPARTMENT":10},{"ON_ORDER":
20,"CA_ITEM_REF":70,"CA_COST":"007.44","IN_STOCK":101,"CA_DESCRIPTION":"Laser Pa
per 28-lb 108 Bright 500\ream","CA_DEPARTMENT":10},{"ON_ORDER":0,"CA_ITEM_REF":
80,"CA_COST":"033.54","IN_STOCK":25,"CA_DESCRIPTION":"Laser Paper 28-lb 108 Brig
```

The *placeOrder* service can be tested using *Postman* or *cURL* with URL:

***https://wg31.washington.ibm.com:9443/zosConnect/services/placeOrder?action=invoke***

and JSON request message.

```
{
  "DFH0XCP1": {
    "orderRequest": {
      "itemID": 70,
      "orderQuantity": 1
    }
  }
}
```

The corresponding cURL command and results are shown below:

```
curl -X POST --user Fred:fredpwd --header "Content-Type: application/json" -d @placeOrder.json
--insecure https://wg31.washington.ibm.com:9443/zosConnect/services/placeOrder?action=invoke
```

```
c:\z\admin>curl -X POST --user Fred:fredpwd --header "Content-Type: application/json"
-d @placeOrder.json --insecure https://wg31.washington.ibm.com:9443/zosC
onnect/services/placeOrder?action=invoke
{"DFH0XCP1":{"CA_RESPONSE_MESSAGE":"ORDER SUCCESSFULLY PLACED","CA_RETURN_CODE":
0}}
```

## Deploy and Test the API

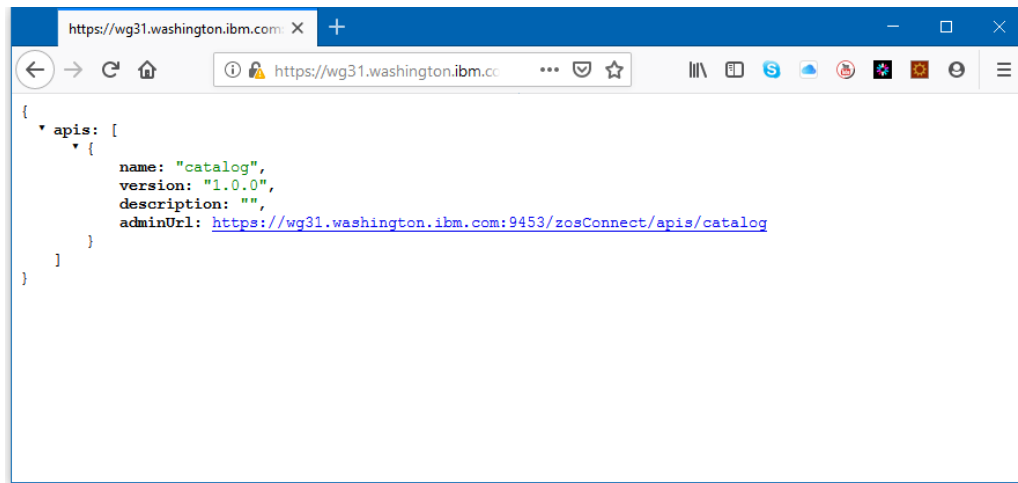
The API artifact should be deployed using z/OS Connect RESTful administrative interface. cURL will be used in this section, but Postman could have been used also.

\_\_\_1. Deploy the API archive file using cURL. Paste the command below at the command prompt and press **Enter**.

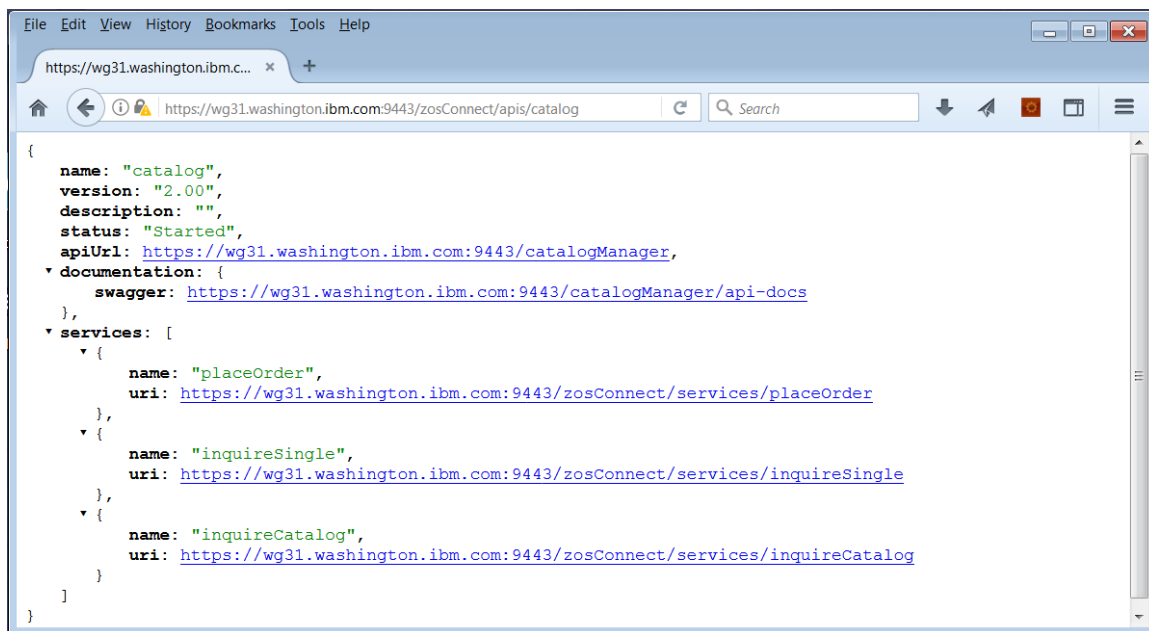
```
curl -X POST --user Fred:fredpwd --data-binary @catalog.aar --header "Content-Type: application/zip" --insecure https://wg31.washington.ibm.com:9443/zosConnect/apis
```

```
C:\z\admin>curl -X POST --user Fred:fredpwd --data-binary @catalog.aar --header
"Content-Type: application/zip" --insecure
https://wg31.washington.ibm.com:9443/zosConnect/apis
{"name":"catalog","version":"1.0.0","description":"","status":"Started",
"apiUrl":"https://wg31.washington.ibm.com:9453/catalog",
"documentation":{"swagger":"https://wg31.washington.ibm.com:9443/catalog/api-docs"},
"services":[
{"name":"placeOrder","uri":"https://wg31.washington.ibm.com:9443/zosConnect/services/place
Order"},
{"name":"inquireSingle","uri":"https://wg31.washington.ibm.com:9443/zosConnect/services/inq
uireSingle"},
{"name":"inquireCatalog","uri":"https://wg31.washington.ibm.com:9443/zosConnect/services/in
quireCatalog"}
]}
```

2. Next enter URL <https://wg31.washington.ibm.com:9443/zosConnect/apis> in the Firefox browser and you should see the window below. The *catalog* API now shows as being available.

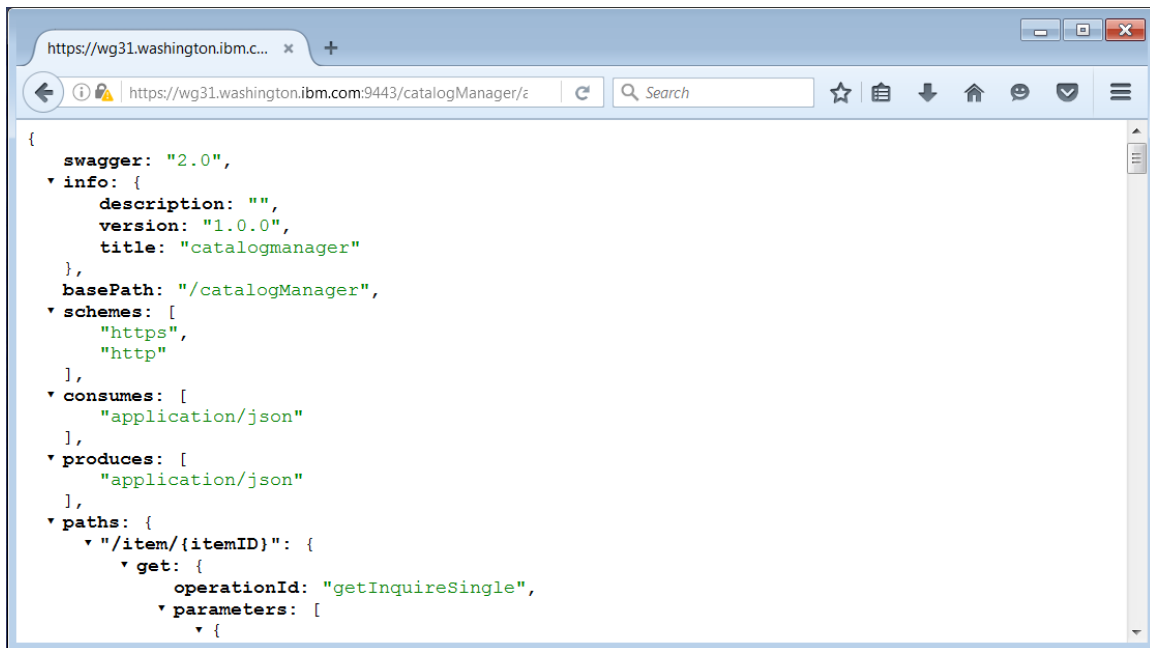


3. If you click on *adminUrl* URL, the window below should be displayed.



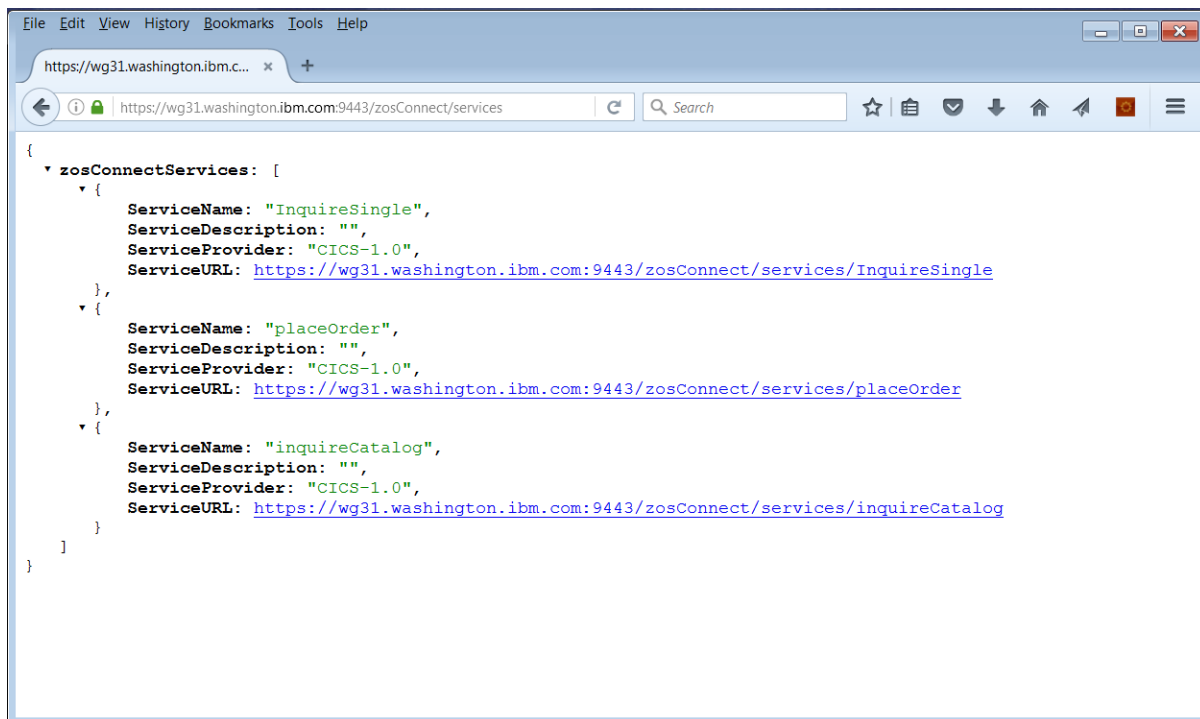


4. Next click on the *swagger* URL and you should see the Swagger document associated with this API:



Explore this Swagger document and you will see the results of the request and response mapping performed earlier. This Swagger document can be used by a developer or other tooling to develop REST clients for this specific API.

5. Next enter URL <https://wg31.washington.ibm.com:9443/zosConnect/services> in the Firefox browser and you should see the window below. The three services are now available.



Use cURL or Postman to test the API.

```
curl -X GET --user Fred:fredpwd --insecure  
https://wg31.washington.ibm.com:9443/catalog/item/0010
```

```
curl -X GET --user Fred:fredpwd --insecure  
https://wg31.washington.ibm.com:9443/catalog/items?startItemID=0010
```

```
curl -X POST --user Fred:fredpwd --data @placeOrder.json --header "Content-Type:  
application/json" --insecure https://wg31.washington.ibm.com:9443/catalog/order
```

The available catalog items are listed below.

Item#	Description	Dept	Cost	In Stock	On Order
0010	Ball Pens Black 24pk	010	002.90	0135	000
0020	Ball Pens Blue 24pk	010	002.90	0006	050
0030	Ball Pens Red 24pk	010	002.90	0106	000
0040	Ball Pens Green 24pk	010	002.90	0080	000
0050	Pencil with eraser 12pk	010	001.78	0083	000
0060	Highlighters Assorted 5pk	010	003.89	0013	040
0070	Laser Paper 28-lb 108 Bright 500/ream	010	007.44	0102	020
0080	Laser Paper 28-lb 108 Bright 2500/case	010	033.54	0025	000
0090	Blue Laser Paper 20lb 500/ream	010	005.35	0022	000
0100	Green Laser Paper 20lb 500/ream	010	005.35	0003	020
0110	IBM Network Printer 24 - Toner cart	010	169.56	0012	000
0120	Standard Diary: Week to view 8 1/4x5 3/4	010	025.99	0007	000
0130	Wall Planner: Eraseable 36x24	010	018.85	0003	000
0140	70 Sheet Hard Back wire bound notepad	010	005.89	0084	000
0150	Sticky Notes 3x3 Assorted Colors 5pk	010	005.35	0036	045
0160	Sticky Notes 3x3 Assorted Colors 10pk	010	009.75	0067	030
0170	Sticky Notes 3x6 Assorted Colors 5pk	010	007.55	0064	030
0180	Highlighters Yellow 5pk	010	003.49	0088	010
0190	Highlighters Blue 5pk	010	003.49	0076	020
0200	12 inch clear rule 5pk	010	002.12	0014	010
0210	Clear sticky tape 5pk	010	004.27	0073	000

## Optional

If you are familiar with CICS Execution Diagnostic Facility (EDF), start a 3270-terminal session with CICS. Clear the screen and enter CICS transaction **CEDX CSMI**. When you repeat any of the above test, you should be able to trace the flow of the request through CICS.

```

Session A
File Edit View Communication Actions Window Help
TRANSACTION: CSMI PROGRAM: DFHMIRS TASK: 0000092 APPLID: CICS53Z DISPLAY: 00
STATUS: PROGRAM INITIATION

EIBTIME      = 184802
EIBDATE      = 0117226
EIBTRNID     = 'CSMI'
EIBTASKN     = 92
EIBTRMID     = '/AC3'

EIBCPOSN     = 0
EIBCALEN     = 0
EIBRID       = X'00'
EIBFN        = X'0000'
EIBRCODE     = X'000000000000'
EIBDS        = .....
+ EIBREQID   = .....

ENTER: CONTINUE
PF1 : UNDEFINED
PF4 : SUPPRESS DISPLAYS
PF7 : SCROLL BACK
PF10: PREVIOUS DISPLAY

PF2 : SWITCH HEX/CHAR
PF5 : WORKING STORAGE
PF8 : SCROLL FORWARD
PF11: EIB DISPLAY

PF3 : END EDF SESSION
PF6 : USER DISPLAY
PF9 : STOP CONDITIONS
PF12: UNDEFINED

01/001
Connected to remote server/host wg31a using lu/pool TCP001

```

## Summary

You have verified the API. The API layer operates above the service layer you defined. The API layer provides a further level of abstraction, allows a more flexible use of HTTP verbs, and provides better mapping of data via the API editor function.