Lab: Setting up the MQ Web Console on z/OS

In this lab, we will go through a basic configuration of the MQ web console on z/OS using basic authentication. The purpose of this exercise is to get the MQ web console up and running. The security configuration here is not sufficient to be used in any production environment.

Prerequisites:

-Java installed on z/OS

-MQ installed on z/OS. We are using MQ 9.3.5 here.

-Unix Systems Services active on z/OS

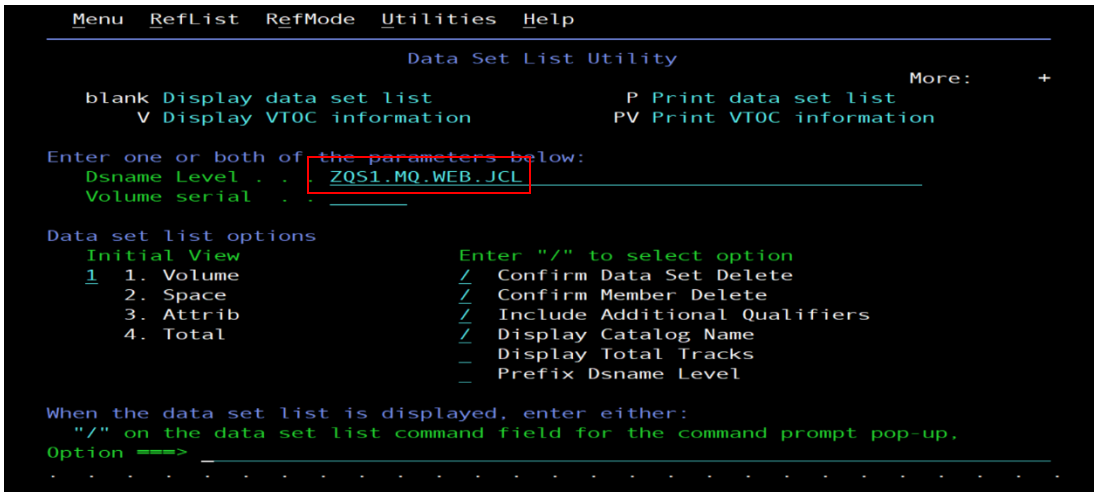
We've installed these all for you here!

Background and context:

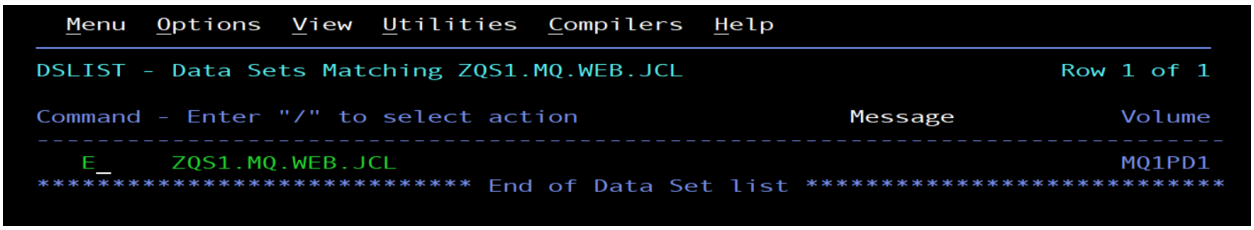
- MQ Console: These enhancements have been made to the MQ Console to provide a more enjoyable user experience and reduce manual tasks to speed administration activities:
  - Storage class administration
  - Queue manager dashboards
  - Application dashboards
  - MQ Console overview tab
  - Improved application visibility
  - Localized time zones

Lab Instructions:

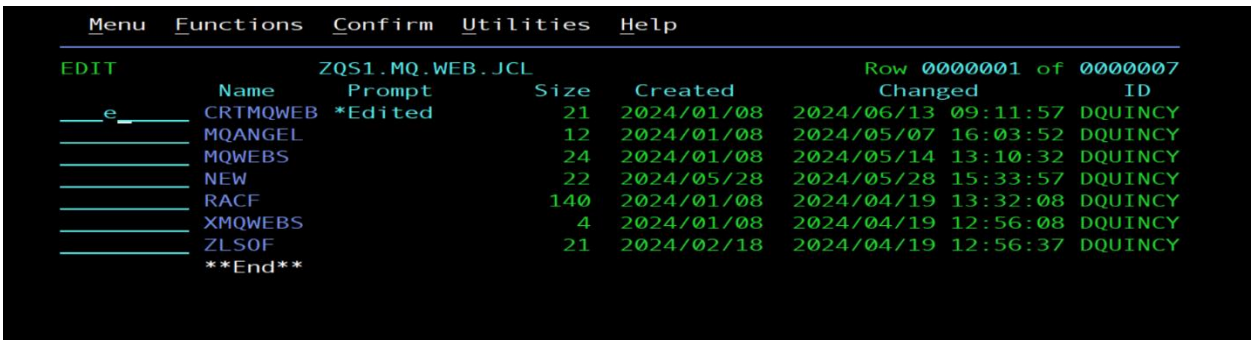
1. First, we will need to submit some JCL to create our web server. Our JCL executes the crtmqweb executable in the MQ installation. From the ISPF main menu, go to option 3.4 to search for our JCL. Our JCL will be located in ZQS1.MQ.WEB.JCL. Put that in 'Dsname Level' and hit enter.



2. Following the below screenshot, enter an ‘b’ to the left of the data set name to browse its contents.



3. Place an ‘e’ next to ‘CRTMQWEB’ and hit enter. This will put you in edit mode for the member.

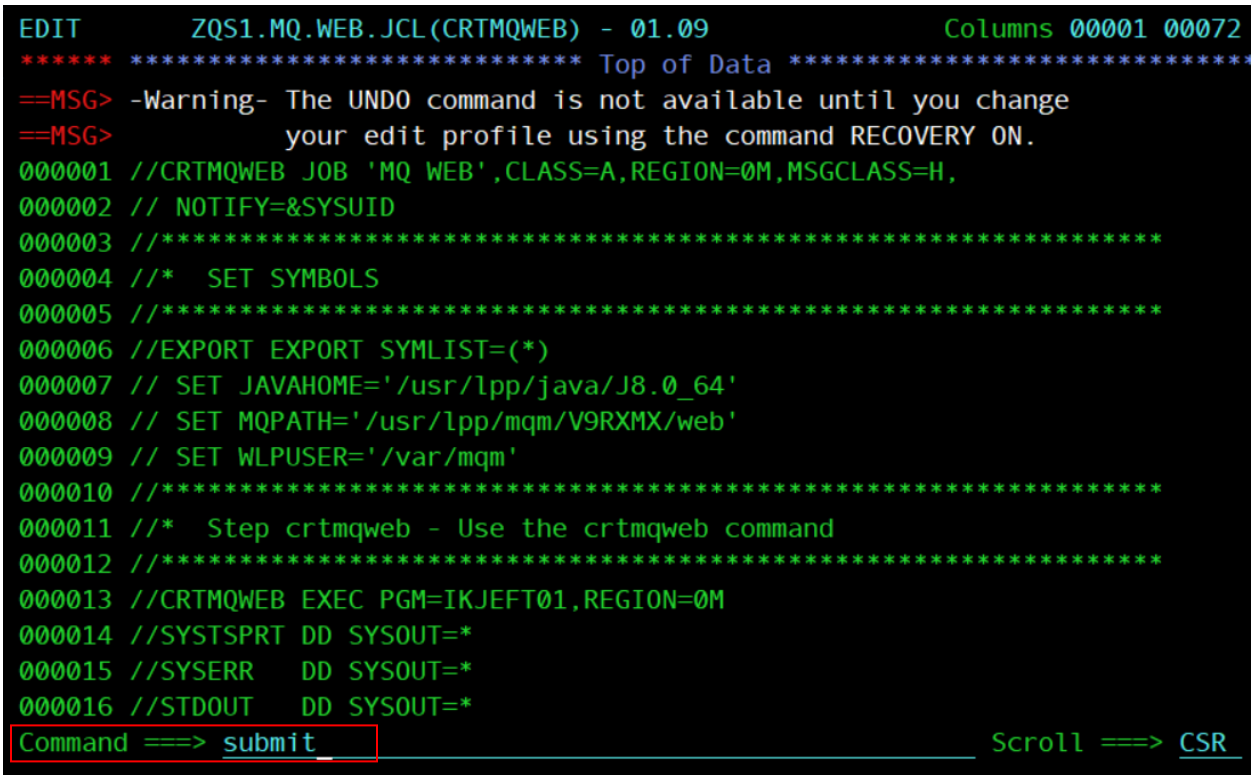


4. Normally, we will have to customize the JAVAHOME, MQPATH, and WLPUSER for our own z/OS environment. We will not have to customize here. We have done it for you.
    a. JAVAHOME will reflect when Java is installed on your z/OS environment
    b. MQPATH will reflect where the MQ binaries are accessible from
    c. WLPUSER will be where the server directory will be created upon submitting the JCL

Sample JCL for CRTMQWEB:

```
//CRTMQWEB JOB 'MQ WEB',CLASS=A,REGION=0M,MSGCLASS=H,
// NOTIFY=&SYSUID
//*****************************************************************
//*  SET SYMBOLS
//*****************************************************************
//EXPORT EXPORT SYMLIST=(*)
// SET JAVAHOME='/usr/lpp/java/J8.0_64'
// SET MQPATH='/usr/lpp/mqm/V9RXMX/web'
// SET WLPUSER='/var/mqm'
//*****************************************************************
//*  Step crtmqweb - Use the crtmqweb command
//*****************************************************************
//CRTMQWEB EXEC PGM=IKJEFT01,REGION=0M
//SYSTSPRT DD SYSOUT=*
//SYSERR   DD SYSOUT=*
//STDOUT   DD SYSOUT=*
//SYSTSIN  DD *,SYMBOLS=EXECSYS
 BPXBATCH SH +
 export JAVA_HOME=&JAVAHOME; +
 export WLP_USER_DIR=&WLPUSER; +
 &MQPATH/bin/crtmqweb &WLPUSER -p MQ
```

5. Once you have reviewed the 3 above paths, submit the JCL by writing ‘submit’ in the command line and hitting enter.



6. Next, let’s check out the angel process. What is that? The Liberty angel process is a started task that allows Liberty servers to use z/OS authorized services. It's long-lived and can be shared among your multiple Liberty servers. Use the F3 key to back out of CRTMQWEB and place an ‘e’ to the left of the MQANGEL member, then hit enter.

Sample JCL for MQANGEL:

```
//MQANGEL PROC PARMS='',COLD=N,NAME='MQANGEL',SAFLOG=Y
//*-------------------------------------------------------------
// SET ROOT='/usr/lpp/mqm/V9RXMX/web'
//*-------------------------------------------------------------
//* Start the Liberty angel process for MQ
//*-------------------------------------------------------------
//STEP1 EXEC PGM=BPXBATA2,REGION=0M,TIME=NOLIMIT,
//  PARM='PGM &ROOT./lib/native/zos/s390x/bbgzangl COLD=&COLD NAME=X
//  &NAME &PARMS SAFLOG=&SAFLOG'
//STDOUT DD SYSOUT=*
//STDERR DD SYSOUT=*
```

7. Lastly, let's check out the MQWEBS JCL. Using the 'e' edit function, make the following changes:

```
  File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
EDIT       SYS1.PROCLIB(MQWEBS) - 01.08                Columns 00001 00072
****** ***************************** Top of Data ******************************
==MSG> -Warning- The UNDO command is not available until you change
==MSG>           your edit profile using the command RECOVERY ON.
000001 //MQWEBS PROC PARMS='mqweb --clean'
000002 //*
000003 //  SET INSTDIR='/usr/lpp/mqm/V9RXMX/web'
000004 //  SET USERDIR='/var/mqm'
000005 //*
000006 //MQCONSOL EXEC PGM=BPXBATSL,REGION=0M,TIME=NOLIMIT,
000007 //  PARM='PGM &INSTDIR./lib/native/zos/s390x/bbgzsrv &PARMS.'
000008 //WLPUDIR  DD PATH='&USERDIR.'
000009 //STEPLIB  DD DSN=MQ935CD.SCSQANLE,DISP=SHR
000010 //         DD DSN=MQ935CD.SCSQAUTH,DISP=SHR
000011 //STDOUT   DD SYSOUT=*
000012 //STDERR   DD SYSOUT=*
000013 //STDIN    DD DUMMY
000014 //STDENV   DD *
000015 JAVA_HOME=/usr/lpp/java/J8.0_64
000016 WLP_USER_DIR=/var/mqm
```

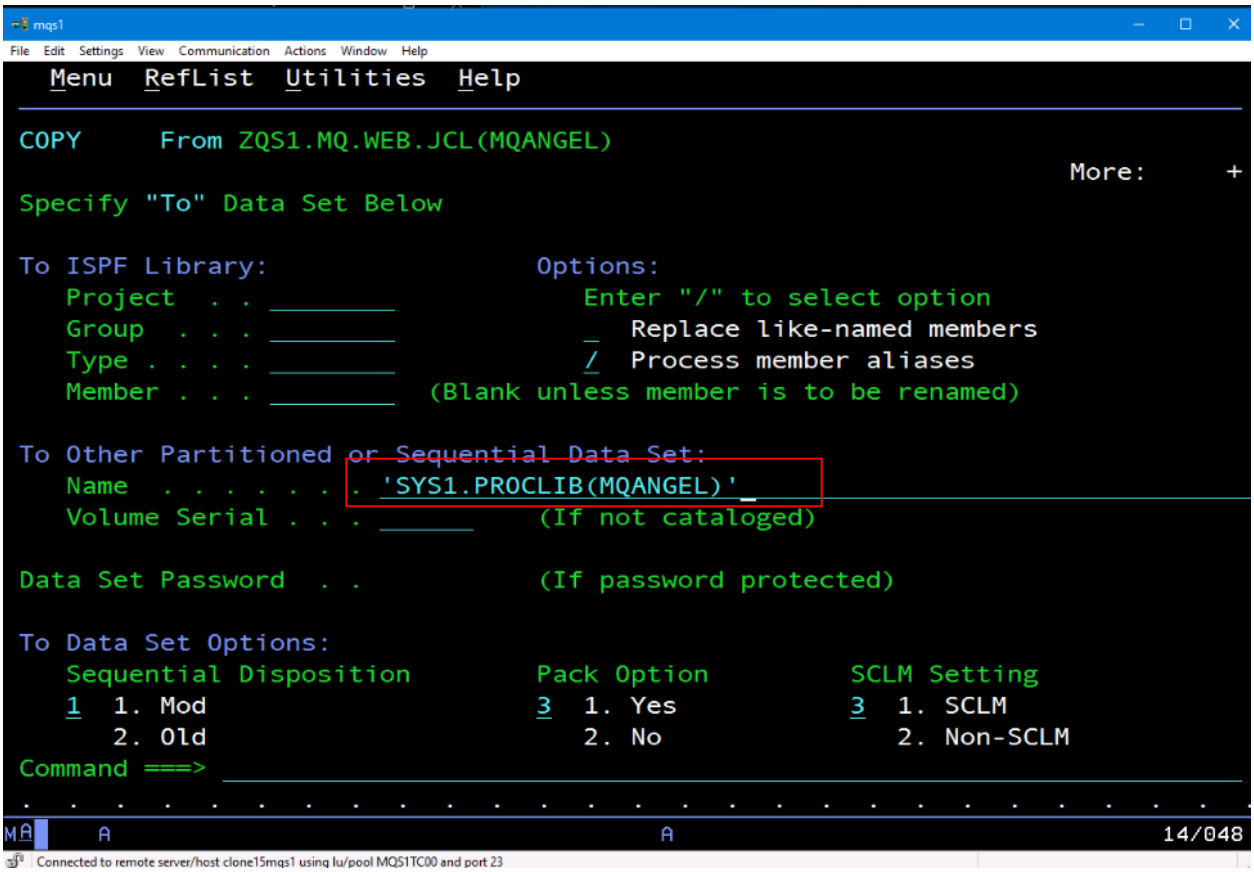
```
000017 PATH=/usr/lpp/mqm/V9RXMX/web/bin:/bin:/usr/bin
000018 LIBPATH=/usr/lpp/mqm/V9RXMX/java/lib
000019 //ZTDENV   DD *
000020 JAVA_HOME=/usr/lpp/java/J8.0_64
000021 WLP_USER_DIR=/var/mqm
000022 PATH=/usr/lpp/mqm/V9RXMX/web/bin:/bin:/usr/bin
000023 LIBPATH=/usr/lpp/mqm/V9RXMX/java/lib
000024 IBM_JAVA_OPTIONS=-Dcom.ibm.ws.zos.core.angelName=MQANGEL
```

8. We won't be submitting MQANGEL or MQWEBS, instead we will be adding it to the SYS1.PROCLIB. SYS1.PROCLIB is a system library in which the procedures that are included with the product are placed when you run the starter job.
9. Back out of MQANGEL and navigate to option 3.3 from the main menu. Enter the option 'C' for copy and then specify the member you'd like to copy next to 'Name'. NOTE! You must use single

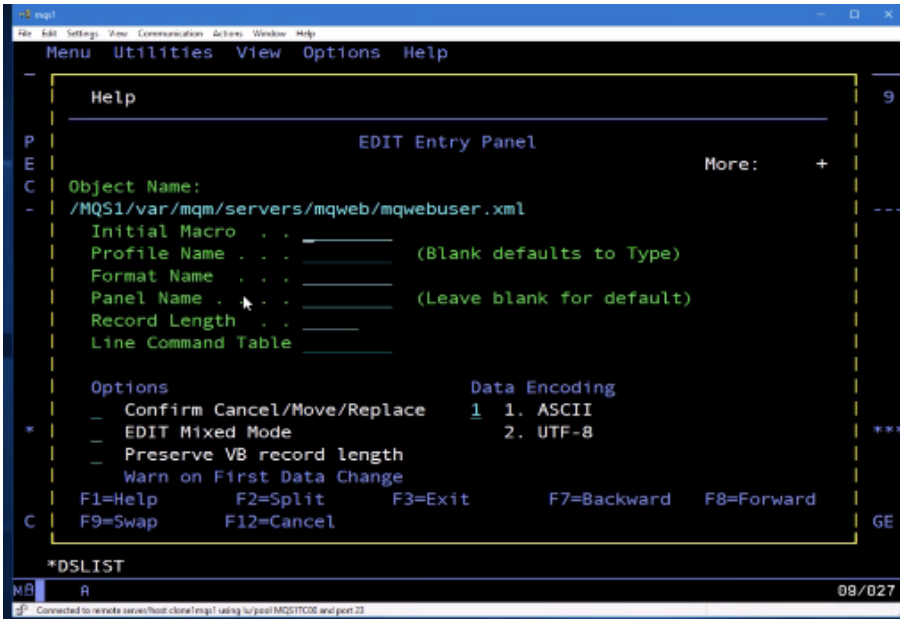
```
                          Move/Copy Utility
 Option ===> c

   C  Copy data set or member(s)          CP Copy and print
   M  Move data set or member(s)          MP Move and print

 Specify "From" Data Set below, then press Enter key

 From ISPF Library:
    Project . . .              (--- Options C and CP only          ---)
    Group . . . .          . . .          . . .          . . .
    Type  . . . .
    Member  . . .                (Blank or pattern for member list,
                                  "*" for all members)

 From Other Partitioned or Sequential Data Set:
    Name  . . . . . .          'ZQS1.MQ.WEB.JCL(MQANGEL)'
    Volume Serial . . .        (If not cataloged)

 Data Set Password  . .        (If password protected)

  F1=Help      F3=Exit      F4=Return    F10=Actions  F12=CRetriev
```

quotes around the data set names.

10. You want the destination for the copy to be ‘SYS1.PROCLIB(MQANGEL)’ . Specify this here.



11. Repeat this copying process for ‘ZQS1.MQ.WEB.JCL(MQWEBS)’
12. Excellent! Now both MQANGEL and MQWEBS will be included in the SYS1.PROCLIB. You can navigate to SYS1.PROCLIB using 3.4 if you’d like to confirm!
13. Now, head over to /var/mqm/ directory from option 3.4 in ISPF, you should be able to see a ‘servers’ directory has been created. In the servers directory, you will see several XML files. We will modify these files.
14. You can browse the servers by placing a ‘l’ to the left of the ‘servers’ option to see its contents. Repeat for ‘mqweb’ directory until you see several XML files.
15. You can modify the files by using option ‘ea’ or ‘EDIT ASCII’ to open the XML file in edit mode. Use



‘ea’ on mqwebuser.xml

16. Once in edit mode, you will want to make the following additions to your XML file. I recommend using copy and paste here, clearing out the old text and put in new text below.

Sample mqwebuser.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
 <server>
   <featureManager>
     <feature>appSecurity-2.0</feature>
   </featureManager>
   <webAppSecurity allowFailOverToBasicAuth="true"/>
   <variable name="httpsPort" value="9443"/>
   <variable name="httpHost" value="-1"/>
   <variable name="mqRestMessagingEnabled" value="true"/>
   <httpEndpoint host="*" httpPort="-1" httpsPost="9443"
     id="defaultHttpEndpoint"/>
</server>
```

17. Save the XML file and back out. Now, let's edit the sever.xml. Make the following adjustments by adding the line in red:

Sample server.xml:

```
<server>
   <include location="${wlp.install.dir}/mq/etc/mqweb.xml"/>
   <include location="mqwebuser.xml"/>
   <include location="basic_registry.xml"/>
 </server>
```

18. As you can see, we added a reference to basic_registry.xml file, but we do not have one currently. Let's change that.
19. Let's pull over a sample from the MQ installation configuration. it's going to be located at /usr/lpp/mqm//V9RXMX/web/mq/samp/configuration. Navigate there using 3.4 from the ISPF menu.
20. Now, type in 'tso omvs' into your command line.
21. Once in OMVS, type in 'ls /usr/lpp/mqm/V9RXMX/web/mq/samp/configuration'. When you hit enter, you should see a list of the XML files in the directory, including basic_registry.xml.

```
DQUINCY:/Z31RE1/usr/lpp/mqm/V9RXMX/web/mq/samp/configuration: >ls
basic_registry.xml     no_security.xml
ldap_registry.xml      zos_saf_registry.xml
```

22. We want that basic_registry in our servers directory! Now, execute

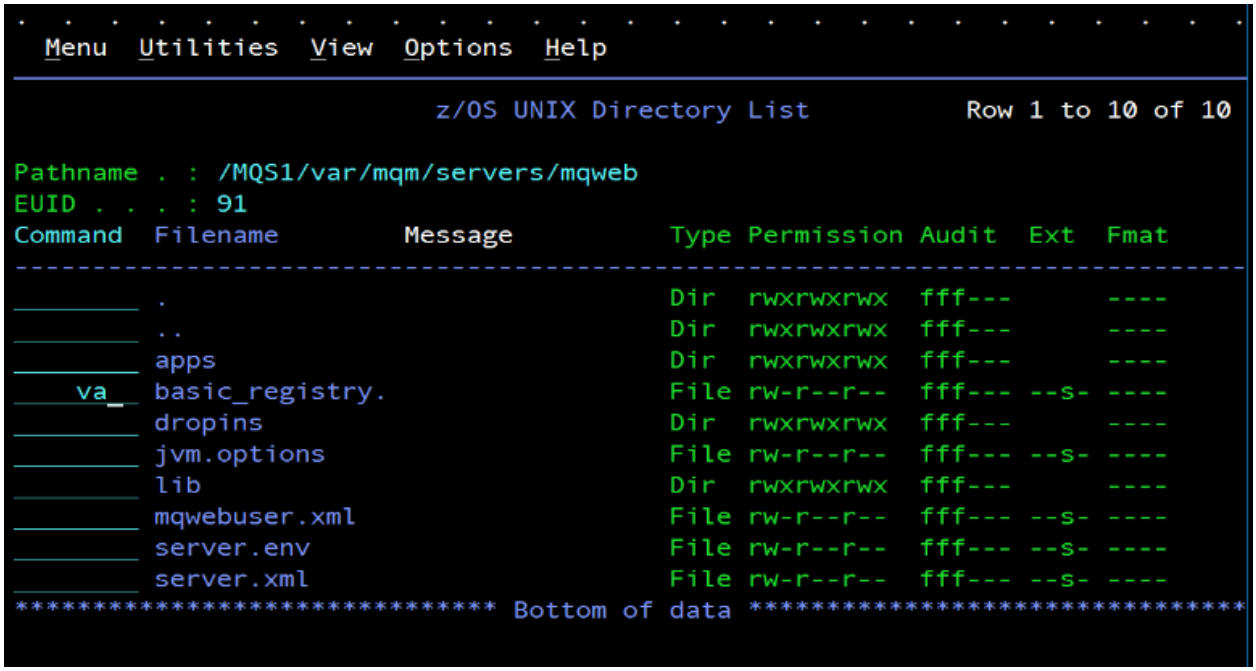
    `cd /var/mqm/servers/mqweb`

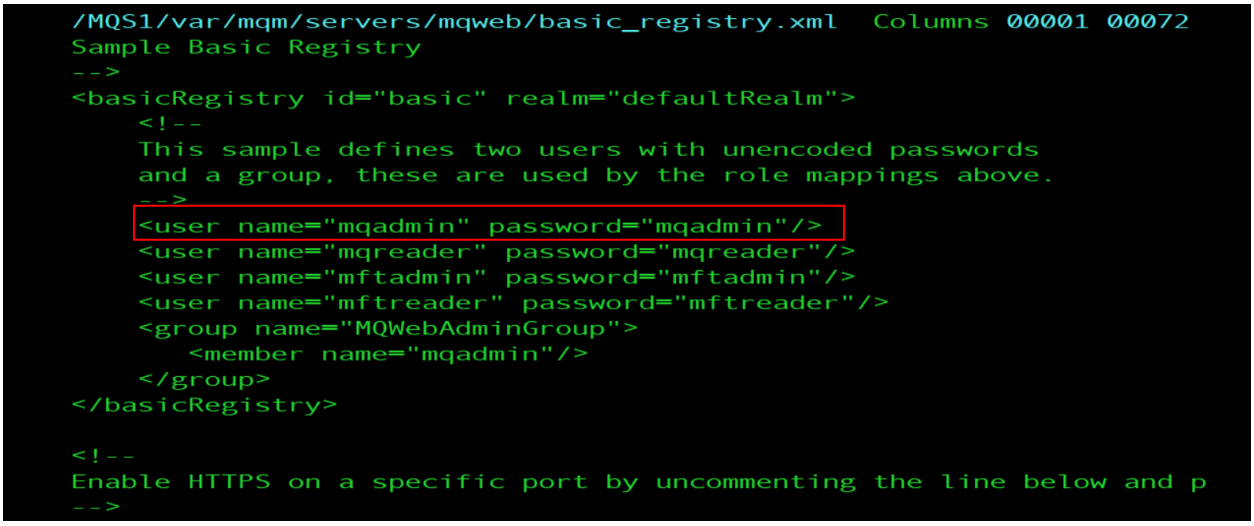
23. Then execute

    `cp /usr/lpp/mqm/V9RXMX/web/mq/samp/configuration/basic_registry.xml .`

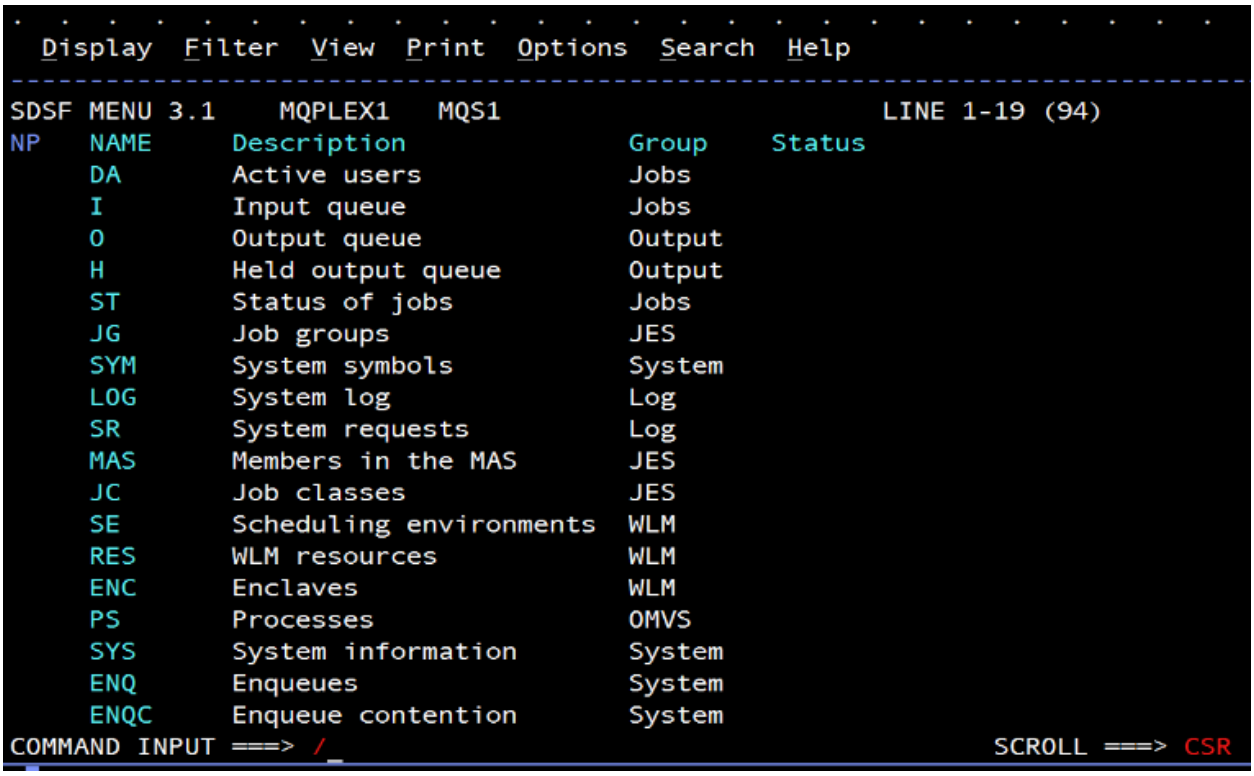
24. Nice! We have now copied the basic registry file to our web console set up. Run an 'ls' to confirm.

25. Enter ‘exit’ to quit out of OMVS. Now, if you back out of the /MQS1/var/mqm/servers/mqweb directory and re-enter it, you will see the basic registry file. Browse it using ‘va’ to look at what credentials users will be able to use for the web console, make note of the mqadmin username and password. You will need those later.

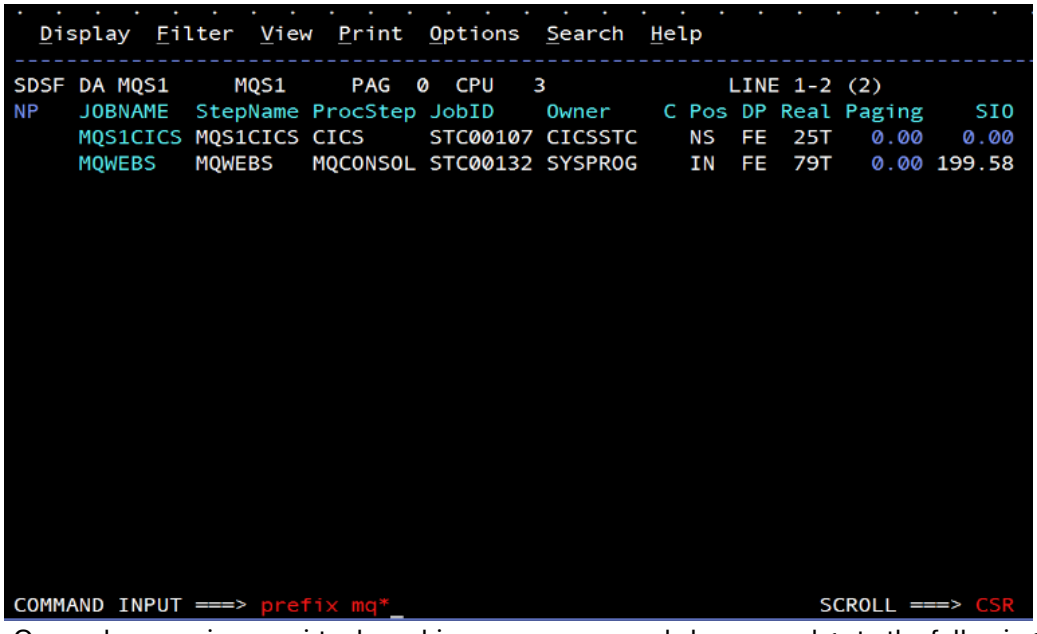




26. You are all configured! Now, lets start up the console! From the ISPF main menu, enter in ‘sdsf’ and hit enter. You will see a new menu popup and you want to enter in a ‘/’ and hit enter in the command line.



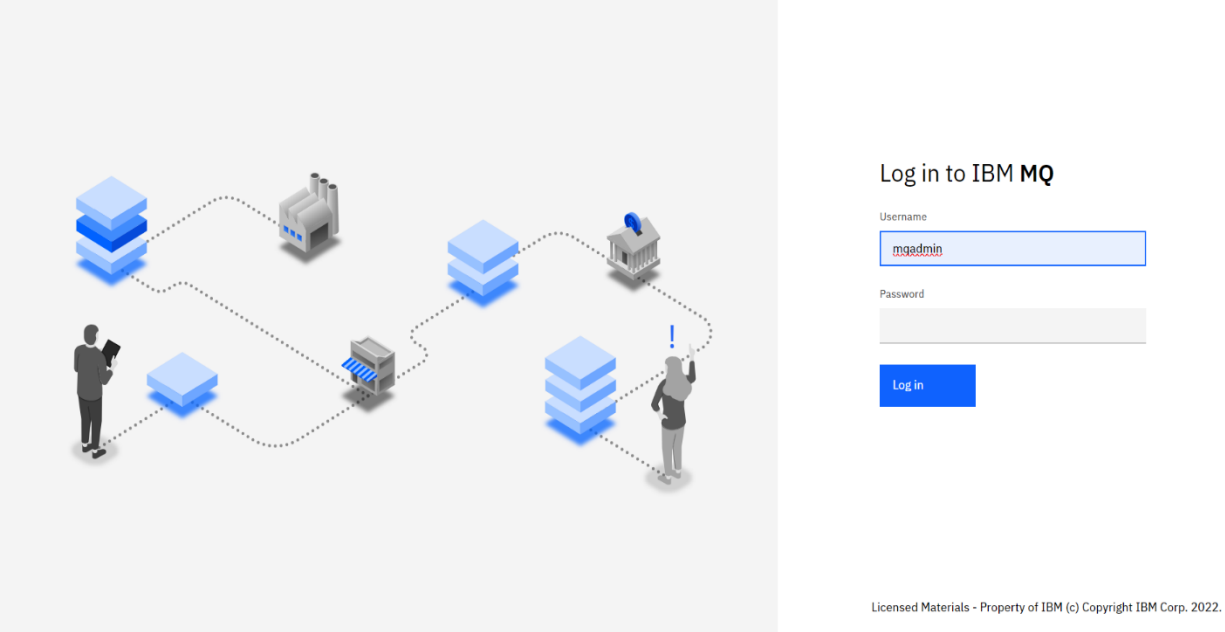
27. A command window will pop up and you can enter the command ‘s mqwebs’ to start the web console.
28. If you navigate to ‘DA’ from the SDSF main menu, you will see ‘MQWEBS’ is now running!



29. Open a browser in your virtual machine, we recommend chrome and go to the following web address:

https://cloneXmqs1:9443/ibmmq/console where X is your lab assigned z/OS environment

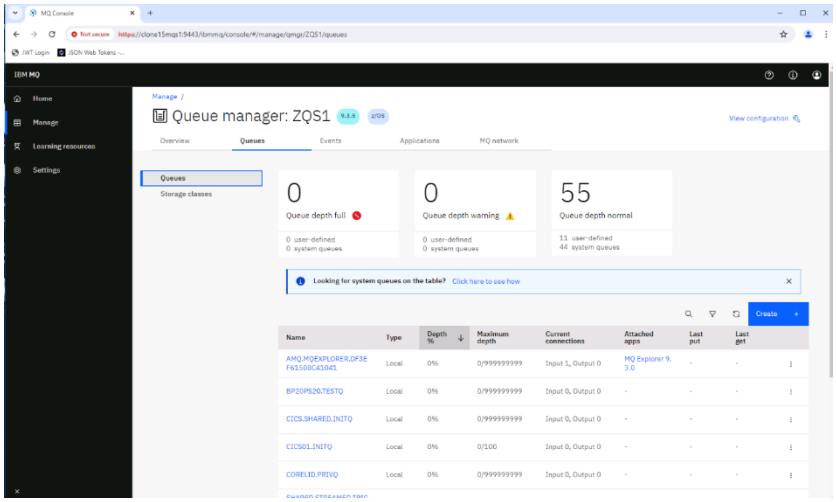
30. A screen like this should pop up. Use the credentials you saw in the basic_registry.xml to login (hint! Username: mqadmin, password: mqadmin)



31. You may notice when you get inside no queue managers running, if that's the case, go back to your z/OS environment and from the sdsf command shell, enter the commands:

```
ZQS1 START QMGR
ZQS1 START CHINIT
ZQS3 START QMGR
ZQS3 START CHINIT
```

32. When you go back to the console, you should now see queue managers! Feel free to investigate the environment, paying attention to all the MQ objects you can interact with via the console.



33. Try and challenge yourself by finding out the name of the queue sharing group ZQS1 and ZQS3 are involved in

Name of Queue Sharing Group:

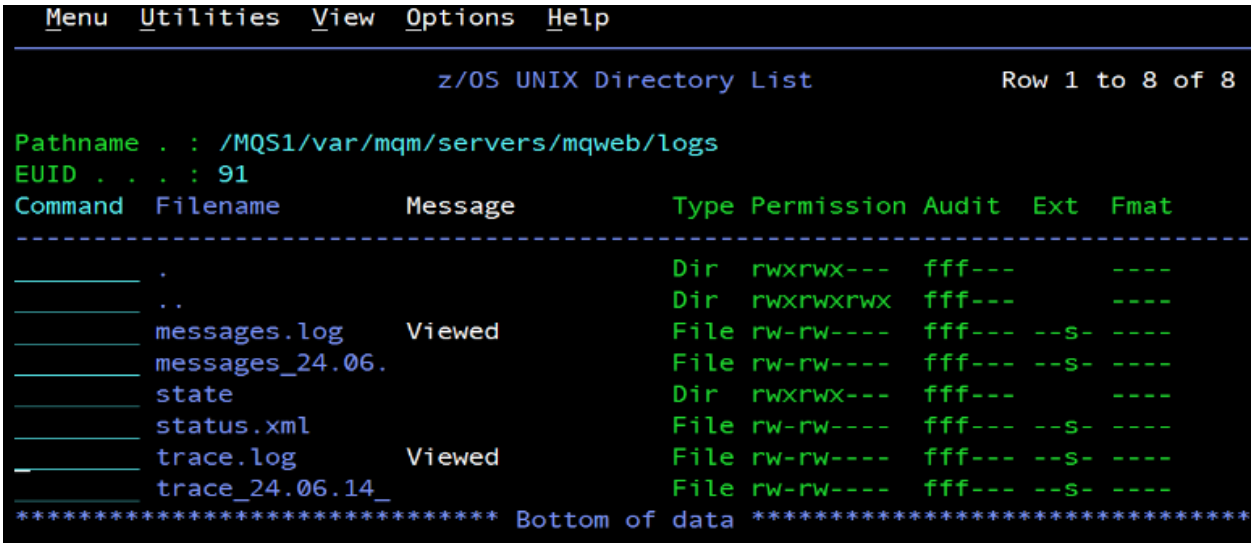
34. Lab complete!

Appendix and Gotcha's:

-Troubleshoot using the 'ST' function of SDSF, you may have a JCL error with MQWEBS if the console doesn't start up with the 's mqwebs' command

-You can take down the console by using the command 'p mqwebs' from the SDSF command shell

Troubleshoot any errors in the MQ web console itself by using the logs provided by the liberty server.



- You need to create the SYSTEM.REST.REPLY.QUEUE in order to use the Liberty server. Do this by using the latest **CSQ4INSG** sample in your MQ*.SCSQPROC