

Where is Wally, eh Zowe !

What is Zowe?



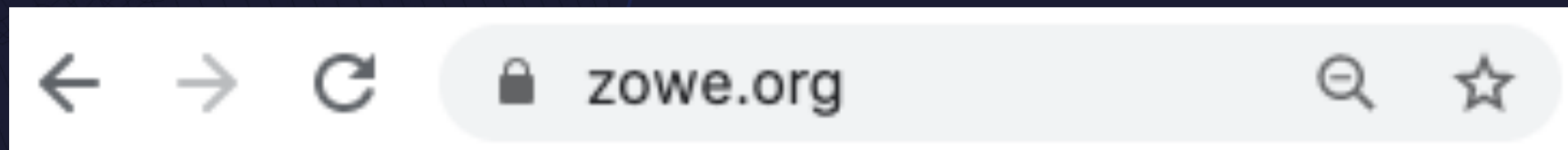
Tobias Leicher | zClient Architect

Charts by Joe Winchester, IBM Hursley



THE **LINUX** FOUNDATION

The Linux Foundation and the projects we support form the most ambitious and successful investment in the creation of shared technology




Open Mainframe Project

From Wikipedia, the free encyclopedia

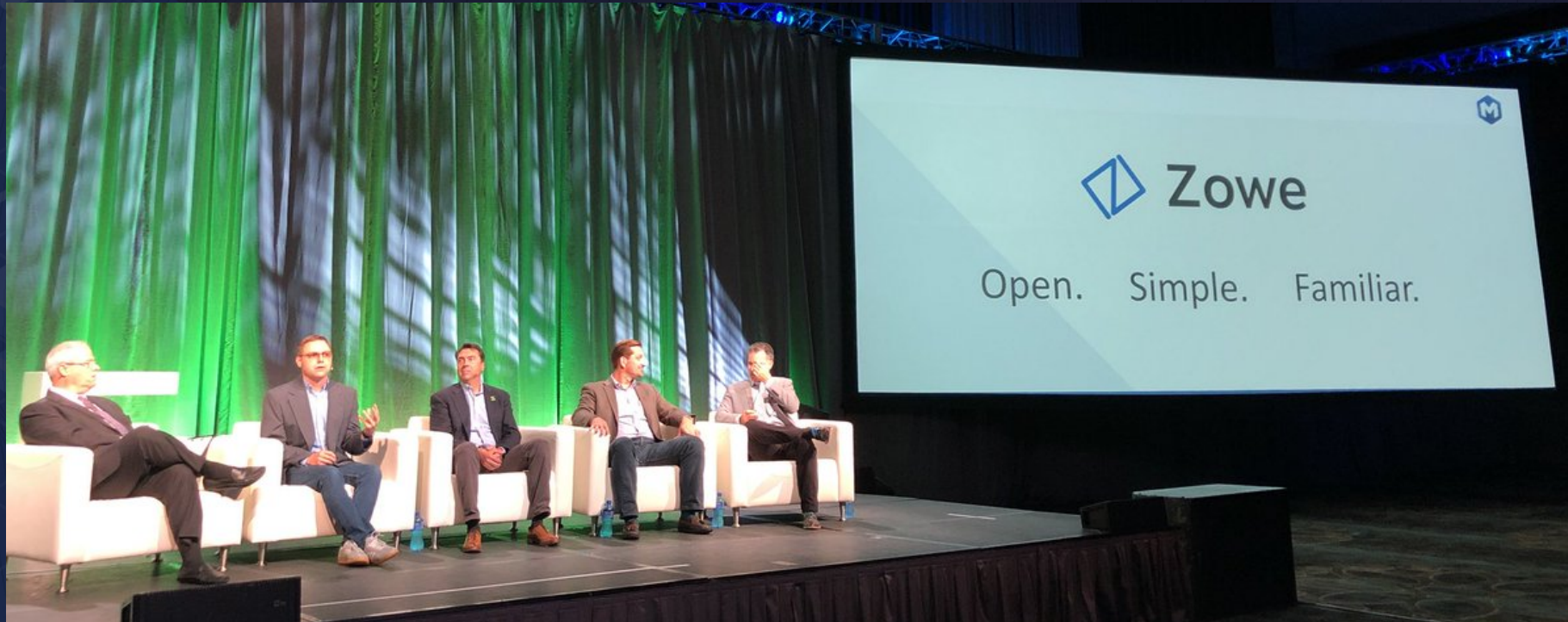
Open Mainframe Project is a Collaborative Project managed by the [Linux Foundation](#) to encourage the use of [Linux](#)-based operating systems and [open source software](#) on [mainframe computers](#).^[1] The project was announced on August 17, 2015 and was driven by [IBM](#), a major supplier of mainframe hardware, as well as 16 other founding members, that included SUSE, CA Technologies, BMC Software, Compuware as well as clients and partners such as RSM Partner, Vicom Infinity, L3C LLP and ADP, and academic institutions such as Marist College and University of Bedfordshire.^[2] Coincident with the announcement, IBM also announced a partnership with [Canonical](#) to make the [Ubuntu](#) operating system available for their high-end z Systems hardware.^{[3][4][5][6]}

The screenshot shows the Open Mainframe Project website homepage. At the top left is the logo with the text "OPEN MAINFRAME PROJECT". To the right is a navigation menu with links for "ABOUT", "PROJECTS", "COMMUNITY", "NEWS", "RESOURCES", and "CONTACT". Further right are social media icons for Twitter, LinkedIn, YouTube, GitHub, and a settings gear, followed by a search icon. The main content area features three columns:

- WHAT IS THE MODERN MAINFRAME?**
Mainframe has transformed over the past 65 years, and still is the goto platform for high transactional and secure computing
[LEARN HOW](#)
- OPEN MAINFRAME 2020 ANNUAL REPORT**
Learn more about how the Open Mainframe Project doubled their projects, grew their global ecosystem, supported several technical milestones and responded to the need for COBOL resources.
[READ IT NOW](#)
- TRY YOUR APP ON MAINFRAME**
Get free cloud access to a mainframe right now, see how your app seamlessly works on mainframe
[TRY NOW](#)
By Proceeding to the trial, you will be leaving the openmainframeproject.org website

 openmainframeproject.org

<https://openmainframeproject.org>



Open

Simple

Familiar

```

Joe$ zowe
Joes-MBP-3:~ Joe$ zowe

DESCRIPTION

Welcome to Zowe CLI!

Zowe CLI is a command line interface (CLI) that provides a simple and
streamlined way to interact with IBM z/OS.

For additional Zowe CLI documentation, visit https://docs.zowe.org

For Zowe CLI support, visit https://www.zowe.org

USAGE

zowe <group>

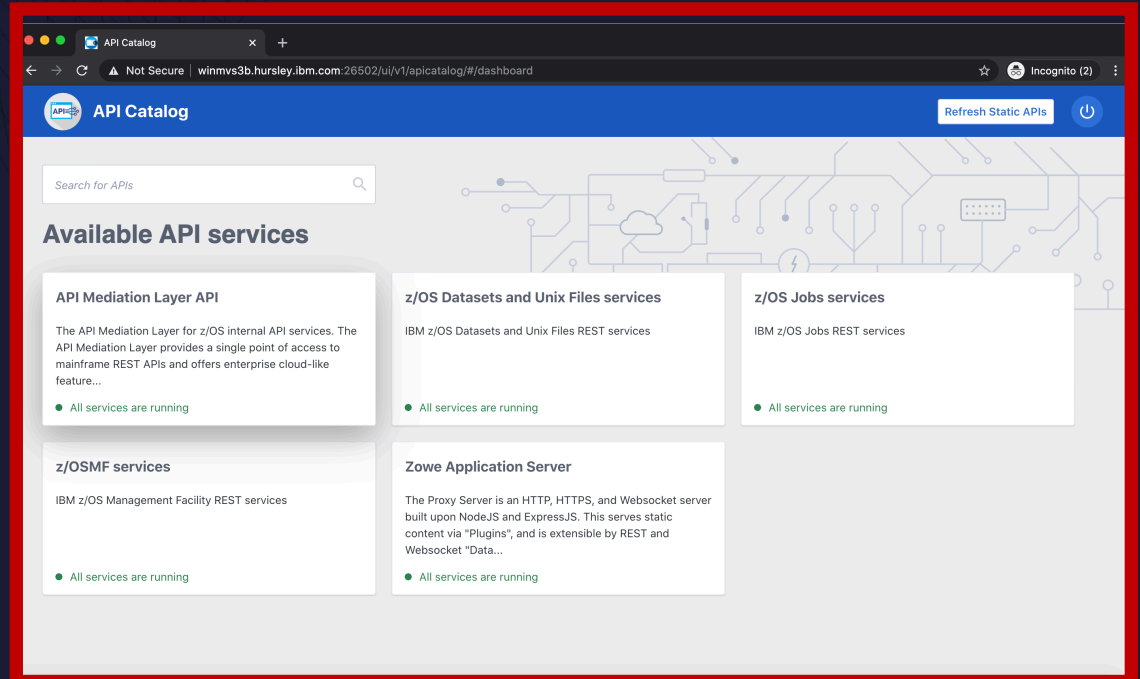
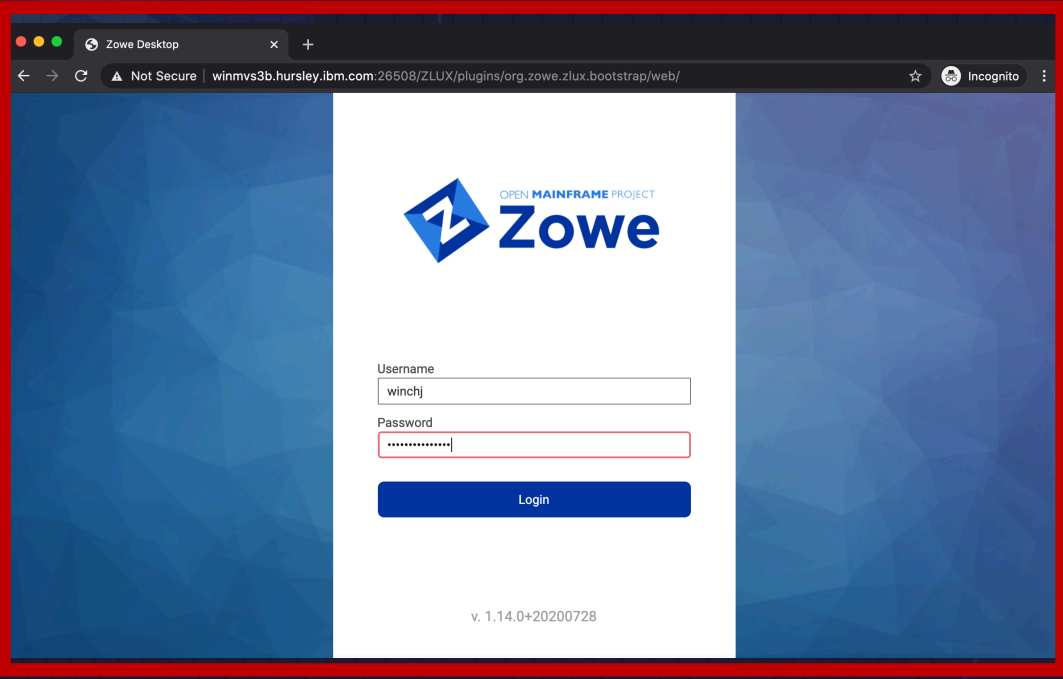
Where <group> is one of the following:

```

```

E:\WINCH\COBOL(SAMPLE).cbl x
1  *****
2  This program demonstrates the following Language
3  Environment callable
4  * services : CEEOUT, CEELECT, CEEDATE
5  *****
6  I D
7  D I V I S I O N
8  *****
9  Identification Division.
10 Program-Id,      AWDPRP
11 *****
12 D A T A
13 D I V I S I O N
14 *****
15 Data Division.
16 Working-Storage Section.
17 *****
18 * Declarations for the local date/time service.
19 *****
20 01 Feedback.
21   02  fb-severity..... PIC 9(4) Binary.
22   02  fb-detail..... PIC X(18).
23   77  fb-output..... PIC S9(9) Binary.
24   77  Ldate..... PIC S9(9) Binary.
25   77  Ltime..... COMP-2.
26   77  Greg..... PIC X(17).
27 *****
28 * Declarations for messages and pattern for date formatting.
29 *****
30 01 Pattern.
31   02..... PIC 9(4) Binary Value 45.
32   02..... PIC X(45) Value
33   "Today is %%%%%%%%%%%%%%%%%, %%%%%%%%%ZD, YYYY.".
34 *****
35 77 Start-Msg..... PIC X(88) Value
36   "Callable Service example starting.".
37 *****
38 77 Ending-Msg..... PIC X(88) Value
39   "Callable Service example ending.".
40 *****

```





```
Joes-MBP-3:drivers Joe$ git
usage: git [--version] [--help] [-C <path>] [-c name=value]
        [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
        [-p | --paginate | --no-pager] [--no-replace-objects] [--bare]
        [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
        <command> [<args>]

These are common Git commands used in various situations:

start a working area (see also: git help tutorial)
  clone      Clone a repository into a new directory
  init       Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)
  add        Add file contents to the index
  mv         Move or rename a file, a directory, or a symlink
  reset      Reset current HEAD to the specified state
  rm         Remove files from the working tree and from the index

examine the history and state (see also: git help revisions)
  bisect     Use binary search to find the commit that introduced a bug
  grep       Print lines matching a pattern
  log        Show commit logs
  show       Show various types of objects
  status     Show the working tree status

grow, mark and tweak your common history
  branch     List, create, or delete branches
  checkout   Switch branches or restore working tree files
  commit     Record changes to the repository
  diff       Show changes between commits, commit and working tree, etc
  merge      Join two or more development histories together
  rebase     Reapply commits on top of another base tip
  tag        Create, list, delete or verify a tag object signed with GPG

collaborate (see also: git help workflows)
```

```
Joes-MBP-3:drivers Joe$ docker

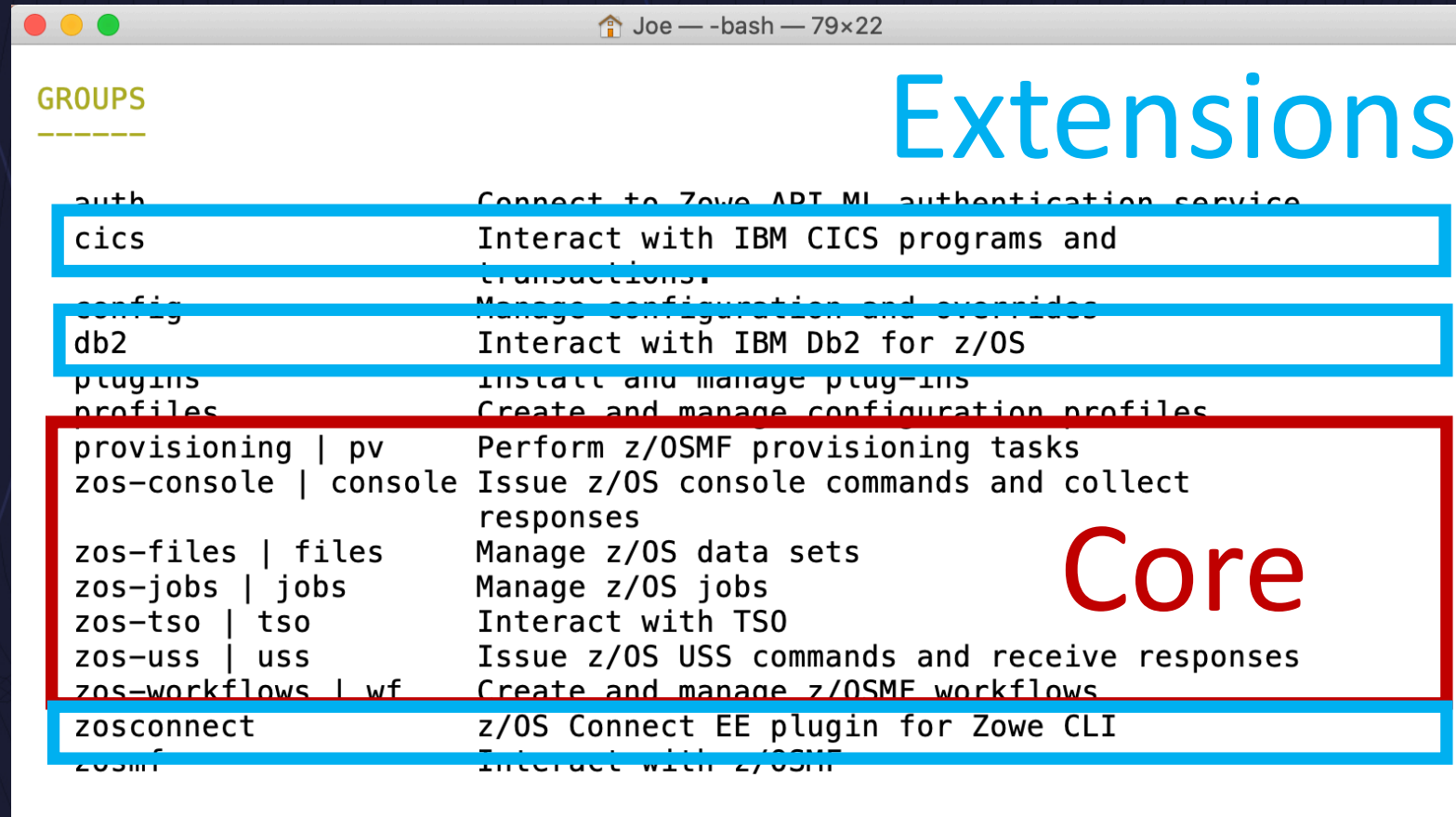
Usage:  docker [OPTIONS] COMMAND

A self-sufficient runtime for containers

Options:
  --config string      Location of client config files (default
                       "/Users/Joe/.docker")
  -c, --context string Name of the context to use to connect to the daemon
                       (overrides DOCKER_HOST env var and default context
                       set with "docker context use")
  -D, --debug          Enable debug mode
  -H, --host list      Daemon socket(s) to connect to
  -l, --log-level string Set the logging level
                       ("debug"|"info"|"warn"|"error"|"fatal") (default "info")
  --tls               Use TLS; implied by --tlsverify
  --tlscacert string  Trust certs signed only by this CA (default
                       "/Users/Joe/.docker/ca.pem")
  --tlscert string    Path to TLS certificate file (default
                       "/Users/Joe/.docker/cert.pem")
  --tlskey string     Path to TLS key file (default
                       "/Users/Joe/.docker/key.pem")
  --tlsverify         Use TLS and verify the remote
  -v, --version       Print version information and quit

Management Commands:
  app*      Docker App (Docker Inc., v0.9.1-beta3)
  builder   Manage builds
  buildx*   Build with BuildKit (Docker Inc., v0.5.1-docker)
  config    Manage Docker configs
  container Manage containers
  context   Manage contexts
  image     Manage images
  manifest  Manage Docker image manifests and manifest lists
  network   Manage networks
```

Command Line Interface (CLI)



The image shows a terminal window titled "Joe --bash-- 79x22". The terminal displays a list of Zowe CLI groups and their descriptions. The word "Extensions" is written in large blue text at the top right, and "Core" is written in large red text at the bottom right. A red box highlights the "Core" group, which includes provisioning, console, files, jobs, tso, uss, and workflows. Other groups like cics, db2, and zosconnect are highlighted with blue boxes.

```
GROUPS
-----

auth          Connect to Zowe API ML authentication service
cics          Interact with IBM CICS programs and
              transactions
config       Manage configuration and overrides
db2          Interact with IBM Db2 for z/OS
plugins      Install and manage plug-ins
profiles     Create and manage configuration profiles
provisioning | pv      Perform z/OSMF provisioning tasks
zos-console | console Issue z/OS console commands and collect
              responses
zos-files   | files    Manage z/OS data sets
zos-jobs   | jobs      Manage z/OS jobs
zos-tso    | tso       Interact with TSO
zos-uss    | uss       Issue z/OS USS commands and receive responses
zos-workflows | wf      Create and manage z/OSMF workflows
zosconnect |          z/OS Connect EE plugin for Zowe CLI
zosmf      |          Interact with z/OSMF
```

Extensions

Core

Progressive Discovery – >zowe files

```
Joe — -bash — 82x34
Joes-MBP-3:~ Joe$ zowe files
```

DESCRIPTION

Manage z/OS data sets, create data sets, and more

USAGE

zowe zos-files <group>

Where <group> is one of the following:

GROUPS

| | |
|----------------------------|---|
| copy cp | Copy a data set |
| create cre | Create data sets |
| delete del | Delete a data set or Unix System Services file |
| download dl | Download content from z/OS data sets and USS files |
| invoke call | Invoke various z/OS utilities |
| list ls | List the details for data sets and the members in the data sets |
| migrate hmigr hMigrate | Migrate data sets. |
| mount | Mount file systems |
| recall hrec hRecall | Recall migrated data sets. |
| rename | Rename a data set or member. |
| unmount umount | Unmount file systems |
| upload ul | Upload the contents of a file to z/OS data sets |

>zowe files list

```
Joe@MBP-3:~$ zowe files list
```

DESCRIPTION

List data sets and data set members. Optionally, you can list their details and attributes.

USAGE

```
zowe zos-files list <command>
```

Where <command> is one of the following:

COMMANDS

```
all-members | am List all members of a pds
data-set | ds List data sets
file-system | fs Listing mounted z/OS filesystems
uss-files | uss List USS files
```

GLOBAL OPTIONS

```
--response-format-json | --rfj (boolean)
```

Produce JSON formatted data from a command

```
--help | -h (boolean)
```

Display help text

>zowe files list ds -help

```
Joe — -bash — 82x34

--help-examples (boolean)
    Display examples for all the commands in a the group

--help-web | --hw (boolean)
    Display HTML help in browser

EXAMPLES
-----

- Show the data set "ibmuser.asm":
    $ zowe zos-files list data-set "ibmuser.asm"

- Show attributes of the data set "ibmuser.cntl":
    $ zowe zos-files list data-set "ibmuser.cntl" -a

- Show all data sets of the user "ibmuser":
    $ zowe zos-files list data-set "ibmuser.*"

- Show attributes of all data sets of the user
  "ibmuser":
    $ zowe zos-files list data-set "ibmuser.*" -a

- Show the first 5 data sets of the user "ibmuser":
    $ zowe zos-files list data-set "ibmuser.cntl" --max 5

Joes-MBP-3:~ Joe$
```

>zowe files list ds WINCHJ.*

```
Joe — -bash — 82x34
Joes-MBP-3:~ Joe$ zowe files list ds "WINCHJ.*"
WINCHJ.ISPF.ISPPROF
WINCHJ.SPFL0G1.LIST
WINCHJ.SPFT0MP1.CNTL
WINCHJ.Z0WE.LOG
WINCHJ.ZWE118.SZWEAUTH
WINCHJ.ZWE118.SZWESAMP
WINCHJ.ZWE1181.SZWEAUTH
WINCHJ.ZWE1181.SZWESAMP
Joes-MBP-3:~ Joe$ zowe files list ds "WINCHJ.*" | grep 181
WINCHJ.ZWE1181.SZWEAUTH
WINCHJ.ZWE1181.SZWESAMP
Joes-MBP-3:~ Joe$
```



Data Sets

```
Joe$ zowe files list data-set WINCHJ.J*
WINCHJ.JCL
WINCHJ.JCL.DEMO

Joe$ zowe files list all-members WINCHJ.JCL
BOBBY
COPYJOB
FROM

Joe$ zowe files download data-set WINCHJ.JCL(COPYJOB)
Data set downloaded successfully.
Destination: winchj/jcl/copyjob.txt
Joe$ cat winchj/jcl/copyjob.txt
//TDM2020 JOB 123456, 'WINCHJ',NOTIFY='WINCHJ',
//          CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1)
//STEP1    EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSIN    DD DUMMY
//SYSUT1   DD DISP=SHR,DSN=WINCHJ.JCL(FROM)
//SYSUT2   DD DISP=SHR,DSN=WINCHJ.JCL(TO)

Joe$ zowe jobs submit data-set "WINCHJ.JCL(COPYJOB)"
jobid:   JOB23416
retcode: null
jobname: TDM2020
status:  INPUT

Joe$ zowe files list all-members WINCHJ.JCL
BOBBY
COPYJOB
FROM
TO
Joe$
```

Jobs

```
Joe - bash - 75x51
Joes-MBP-3:~ Joe$ zowe jobs list jobs --prefix T*
JOB23417 CC 0000 TDM2020 OUTPUT
Joes-MBP-3:~ Joe$ zowe jobs list spool-files-by-jobid JOB23417
2 JESMSGLG JES2
3 JESJCL JES2
4 JESYSMSG JES2
101 SYSPRINT STEP1
Joes-MBP-3:~ Joe$ zowe jobs view spool-file-by-id JOB23417 4
ICH70001I WINCHJ LAST ACCESS AT 19:03:37 ON TUESDAY, JANUARY 19, 2021
IEFA111I TDM2020 IS USING THE FOLLOWING JOB RELATED SETTINGS:
        SWA=BELOW,TIOT SIZE=32K,DSENQSHR=DISALLOW,GDGBIAS=JOB
IEF236I ALLOC. FOR TDM2020 STEP1
IEF237I JES2 ALLOCATED TO SYSPRINT
IEF237I DMY ALLOCATED TO SYSIN
IGD103I SMS ALLOCATED TO DDNAME SYSUT1
IGD103I SMS ALLOCATED TO DDNAME SYSUT2
IEF142I TDM2020 STEP1 - STEP WAS EXECUTED - COND CODE 0000
IEF285I WINCHJ.TDM2020.JOB23417.D0000101.?          SYSOUT
IGD104I WINCHJ.JCL          RETAINED, DDNAME=SYSUT1
IGD104I WINCHJ.JCL          RETAINED, DDNAME=SYSUT2
IEF373I STEP/STEP1 /START 2021019.1912
IEF032I STEP/STEP1 /STOP 2021019.1912
        CPU:      0 HR  00 MIN  00.00 SEC          SRB:      0 HR  00 MIN  00.00 SEC
        VIRT:     80K  SYS:   264K  EXT:           4K  SYS:   11220K
        ATB- REAL:                1036K  SLOTS:                0K
        VIRT- ALLOC:                11M  SHRD:                0M
IEF375I JOB/TDM2020 /START 2021019.1912
IEF033I JOB/TDM2020 /STOP 2021019.1912
        CPU:      0 HR  00 MIN  00.00 SEC          SRB:      0 HR  00 MIN  00.00 SEC
Joes-MBP-3:~ Joe$ █
```

Unix System Services (USS)

```
Joe — bash — 72x26
Joes-MBP-3:~ Joe$ zowe files create uss-directory /u/winchj/zowe/files
USS file or directory created successfully.
Joes-MBP-3:~ Joe$ touch new_file.txt
Joes-MBP-3:~ Joe$ zowe files upload file-to-uss new_file.txt /u/winchj/zowe/files/from_pc.txt
success: true
from:    new_file.txt
to:      /u/winchj/zowe/files/from_pc.txt

USS file uploaded successfully.
Joes-MBP-3:~ Joe$ zowe files list uss /u/winchj/zowe/files
.          drwxr-xr-x  8192  225 WINCHJ
..         drwxrwxrwx  8192  225 WINCHJ
from_pc.txt -rw-r--r--    225 WINCHJ
Joes-MBP-3:~ Joe$ zowe uss issue ssh "chmod a+x *" --cwd /u/winchj/zowe/files

@@START OF COMMAND@@
$ cd /u/winchj/zowe/files && chmod a+x *

Joes-MBP-3:~ Joe$ zowe files list uss /u/winchj/zowe/files
.          drwxr-xr-x  8192  225 WINCHJ
..         drwxrwxrwx  8192  225 WINCHJ
from_pc.txt -rwxr-xr-x    225 WINCHJ
Joes-MBP-3:~ Joe$ █
```

```
DEMO_PDS="STEVENH.DEMO.JCL"
```

```
ZOSMF_PROFILE=3bsh
```

```
# Check and see if pds already exists
MATCHES=`zowe zos-files list data-set "$DEMO_PDS" --zosmf-p $ZOSMF_PROFILE --response-format-json | jq -r '.data.apiResponse.returnedRows'
if [ $MATCHES -gt 0 ]; then
    echo "Data set $DEMO_PDS already exists, deleting"
    zowe zos-files delete data-set -f "$DEMO_PDS" --zosmf-p $ZOSMF_PROFILE
fi
```

```
zowe zos-files create data-set-classic $DEMO_PDS --zosmf-p $ZOSMF_PROFILE
zowe zos-files upload stdin-to-data-set "$DEMO_PDS(INPUT)" <<< $1 --zosmf-p $ZOSMF_PROFILE
zowe zos-files upload stdin-to-data-set --zosmf-p $ZOSMF_PROFILE "$DEMO_PDS(COPY)" <<EOF
```

```
//COPY JOB 123456, 'TSTRADM',NOTIFY='TSTRADM',
```

```
//          CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1)
```

```
//STEP1     EXEC PGM=IEBGENER
```

```
//SYSPRINT DD SYSOUT=A
```

```
//SYSIN     DD DUMMY
```

```
//SYSUT1    DD DISP=SHR,DSN=$DEMO_PDS(INPUT)
```

```
//SYSUT2    DD DISP=SHR,DSN=$DEMO_PDS(OUTPUT)
```

```
//STEP2     EXEC PGM=AOPBATCH,PARM='sleep 5'
```

```
EOF
```

```
JOBID=`zowe jobs submit data-set "$DEMO_PDS(copy)" --zosmf-p $ZOSMF_PROFILE --response-format-json | jq -r '.data.jobid'`
```

```
echo "JOBID is $JOBID"
```

```
i="0"
```

```
while [ $i -lt 5 ]
```

```
do
```

```
    sleep 1s
```

```
    STATUS=`zowe jobs view job-status-by-jobid $JOBID --response-format-json --zosmf-p $ZOSMF_PROFILE | jq -r '.data.status'`
```

```
    if [ "$STATUS" = "OUTPUT" ]; then
```

```
        echo "Job $JOBID has now completed"
```

```
        i=5
```

```
    else
```

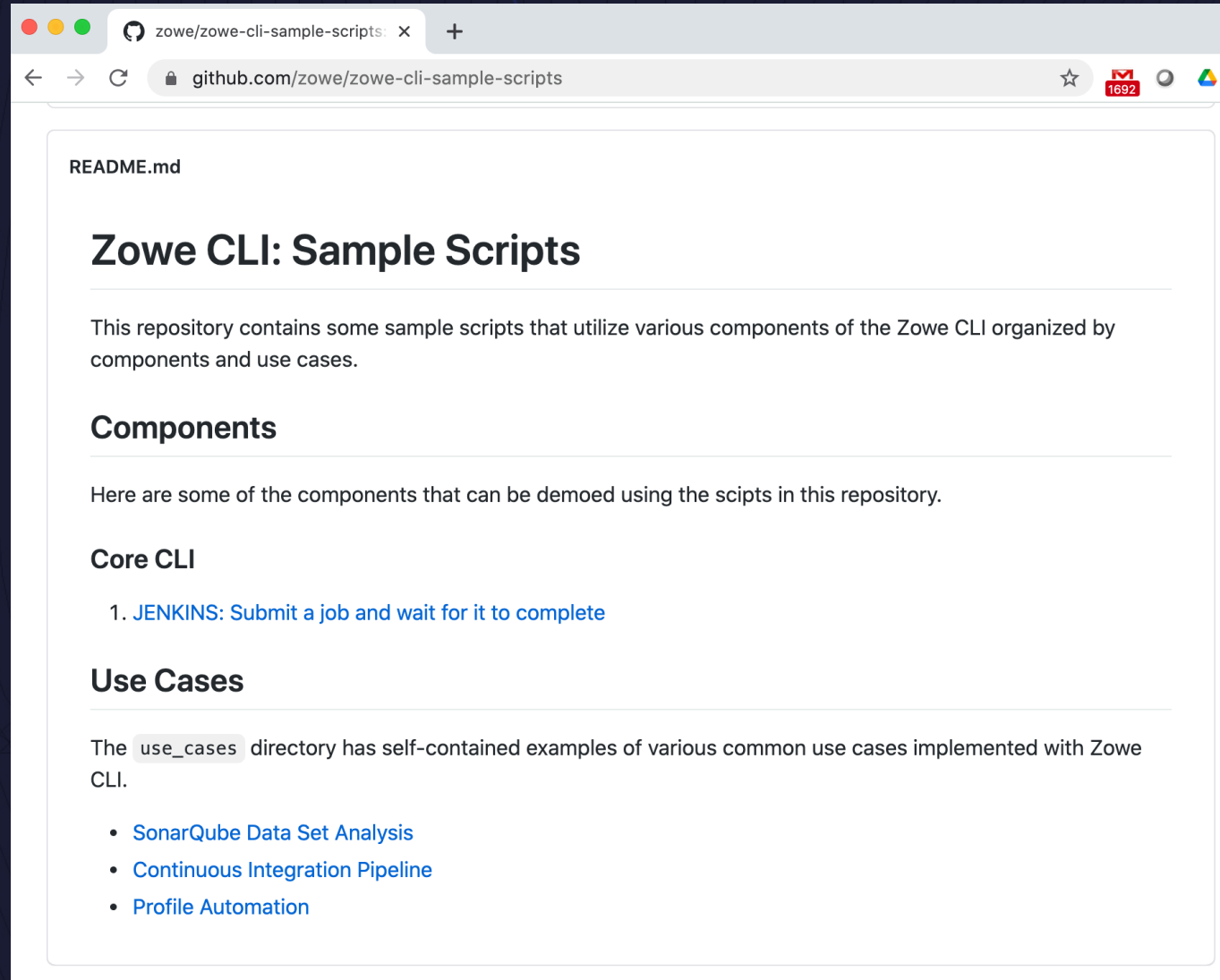
```
        echo "Waiting for job output to complete. Current status is $STATUS"
```

```
    fi
```

```
    i=$((i+1))
```

```
done
```

<https://github.com/zowe/zowe-cli-sample-scripts>



The image shows a browser window with the URL `github.com/zowe/zowe-cli-sample-scripts`. The page content is a README file. It features a main heading, a descriptive paragraph, and three sub-sections: Components, Core CLI, and Use Cases. The Use Cases section includes a bulleted list of links to specific scripts.

zowe/zowe-cli-sample-scripts: x +

github.com/zowe/zowe-cli-sample-scripts

README.md

Zowe CLI: Sample Scripts

This repository contains some sample scripts that utilize various components of the Zowe CLI organized by components and use cases.

Components

Here are some of the components that can be demoed using the scripts in this repository.

Core CLI

1. [JENKINS: Submit a job and wait for it to complete](#)

Use Cases

The `use_cases` directory has self-contained examples of various common use cases implemented with Zowe CLI.

- [SonarQube Data Set Analysis](#)
- [Continuous Integration Pipeline](#)
- [Profile Automation](#)

https://github.com/zowe/zowe-cli-sample-scripts/blob/master/Jenkins/Simple%20Pipeline/demo_content.sh

```
6 # #
7 # SPDX-License-Identifier: EPL-2.0 #
8 # #
9 # Copyright Contributors to the Zowe Project. #
10 #
11
12
13 #test
14
15 #submit our job
16 jobid=$(zowe zos-jobs submit data-set "solsu01.mimpds.cntl(cblrun)" --rff jobid --rft string)
17
18 echo "Submitted our job, JOB ID is $jobid"
19
20 #wait for it to go to output
21 status="UNKNOWN"
22 while [ "$status" != "OUTPUT" ]; do
23     echo "Checking status of job $jobid"
24     status=$(zowe zos-jobs view job-status-by-jobid "$jobid" --rff status --rft string)
25     echo "Current status is $status"
26     sleep 5s
27 done;
28
29 echo "Job completed in OUTPUT status. Final result of job: "
30 zowe zos-jobs view job-status-by-jobid "$jobid"
31 zowe zos-jobs list spool-files-by-jobid "$jobid"
32
33 zowe zos-jobs view sfbi "$jobid" 2
34 zowe zos-jobs view sfbi "$jobid" 103
```


<https://www.npmjs.com/search?q=zowe>

npm promotes metadefinitions Products Pricing Documentation Community

npm Search Sign Up Sign In

46 packages found 1 2 3

Sort Packages
Optimal
Popularity
Quality
Maintenance

@zosconnect/zosconnect-zowe-cli p q m
Z/OS Connect EE Plugin for Zowe CLI
zowe z/os
crshnburn published 1.2.1 • 8 months ago

@zowe/zos-files-for-zowe-sdk p q m
Zowe SDK to interact with files and data sets on z/OS
zosmf mainframe CLI zos files datasets z/OSMF
mvs os390 z/OS zowe
zowerobot published 6.28.0 • 4 days ago

@zowe/zosmf-for-zowe-sdk p q m
Zowe SDK to interact with the z/OS Management Facility
zosmf mainframe CLI zos z/OSMF mvs os390
z/OS zowe
zowerobot published 6.28.0 • 4 days ago

@zowe/zos-tso-for-zowe-sdk p q m
Zowe SDK to interact with TSO on z/OS

<https://www.npmjs.com/search?q=zowe%20db2>

♥ Nocturnal Pajama Mutants

Products Pricing Documentation Community

npm

🔍 zowe db2

✕

Search

Sign Up

Sign In

1 packages found

Sort Packages

@zowe/db2-for-zowe-cli

Optimal

IBM® Db2® Plug-in for Zowe CLI

Popularity

 **zowerobot** published 4.1.0 • a month ago

Quality

Maintenance

p
q
m

>zowe plugins install @zowe/db2-for-zowe-cli

```
Joe -- bash -- 82x34
Joes-MBP-3:~ Joe$ zowe plugins install @zowe/db2-for-zowe-cli
Plugin-ins within the imperative CLI framework can legitimately gain
control of the zowe CLI application during the execution of every command.
Install 3rd party plug-ins at your own risk.

Imperative's plugin installation program handles @zowe peer dependencies.
You can safely ignore NPM warnings about missing @zowe peer dependencies.

Registry = https://registry.npmjs.org/

-----
npm WARN deprecated request@2.88.2: request has been deprecated, see https://github.com/request/request/issues/3142
npm WARN deprecated har-validator@5.1.5: this library is no longer supported
npm WARN @zowe/db2-for-zowe-cli@4.1.0 requires a peer of @zowe/imperative@^4.0.0 but none is installed. You must install peer dependencies yourself.
npm WARN @zowe/db2-for-zowe-cli@4.1.0 requires a peer of @zowe/cli@^6.0.0 but none is installed. You must install peer dependencies yourself.

Installed plugin name = '@zowe/db2-for-zowe-cli'
_____ Validation results for plugin '@zowe/db2-for-zowe-cli' _____
This plugin was successfully validated. Enjoy the plugin.

Joes-MBP-3:~ Joe$ █
```

>zowe db2

```
Joe — -bash — 82x34
Joes-MBP-3:~ Joe$ zowe db2
```

DESCRIPTION

Interact with IBM Db2 for z/OS

USAGE

zowe db2 <group>

Where <group> is one of the following:

GROUPS

```
call      Call a stored procedure
execute   Execute a SQL query
export    Export a table
```

GLOBAL OPTIONS

--response-format-json | --rfj (boolean)
Produce JSON formatted data from a command

--help | -h (boolean)
Display help text

--help-examples (boolean)

>zowe db2 execute sql --query

```
Joe — -bash — 82x34
Joes-MBP-3:~ Joe$ zowe db2 execute sql --query "select count(*) from sysibm.systables"
Result #1
-----
1: 364

Joes-MBP-3:~ Joe$ zowe db2 execute sql --query "select count(*) from sysibm.systables where TSNAME='SYSTSTAB'"
Result #1
-----
1: 57

Joes-MBP-3:~ Joe$ █
```

host connectivity endpoint location + authentication information



Where ?

Host:

Port:

Who am I ?

TSO User ID:

Credentials:

z/OS

zowe profiles



```
Joe$ zowe profiles

DESCRIPTION
-----

  Create and manage configuration profiles

USAGE
-----

  zowe profiles <group>

  Where <group> is one of the following:

GROUPS
-----

  create | cre      Create new configuration profiles
  delete | rm       Delete existing profiles
  list   | ls       List existing profiles
  set-default | set Set which profiles are loaded by default
  update | upd      Update existing profiles

GLOBAL OPTIONS
-----

  --response-format-json | --rfj (boolean)

  Produce JSON formatted data from a command

  --help | -h (boolean)

  Display help text
```

z/OS

zowe profiles create: list: --zosmf-profile



```
Joe — -bash — 82x34
Joes-MBP-3:~ Joe$ zowe profiles create zosmf-profile yoda --host tv5003.svl.ibm.com --port 443 --user WINCHJ --password *****
Profile created successfully! Path:
/Users/Joe/.zowe/profiles/zosmf/yoda.yaml

host:          tv5003.svl.ibm.com
port:          443
user:          WINCHJ
password:      *****
rejectUnauthorized: true

Review the created profile and edit if necessary using the profile update command.
Joes-MBP-3:~ Joe$ zowe profiles list zosmf
2e
3b
svl (default)
svlapi
ukzowe4
yoda
Joes-MBP-3:~ Joe$ zowe files list ds WINCHJ.* --zosmf-profile yoda
Joes-MBP-3:~ Joe$ zowe files list ds WINCHJ.* --zosmf-profile yoda
WINCHJ.ISPF.ISPPROF
WINCHJ.SPFL0G1.LIST
WINCHJ.SPFT0MP1.CNTL
WINCHJ.Z0WE.LOG
WINCHJ.ZWE118.SZWEAUTH
WINCHJ.ZWE118.SZWESAMP
WINCHJ.ZWE1181.SZWEAUTH
WINCHJ.ZWE1181.SZWESAMP
Joes-MBP-3:~ Joe$ █
```

A white box containing the text "z/OS" in a stylized font. A white arrow points from the terminal window to this box.

zowe profiles db2 create – look in *MSTR JES for details



```
DSNL004I #DI2E DDF START COMPLETE 025
LOCATION DSNV102E
LU GBIBMIYA.IYCYZDBE
GENERICLU -NONE
DOMAIN winmvs2e.hursley.ibm.com
TCPPOINT 41100
SECPOINT 30100
```



z/OS

A white arrow points from the right side of the central text box to a white-bordered box containing the text 'z/OS'.

zowe profiles db2 create name host port user password database

```
Joe — -bash — 82x13
[Joe@MBP-3:~ Joe$ zowe profiles create db2 2e --host winmvs2e.hursley.ibm.com --p
ort 41100 --user winchj --password ***** --database DSNV112E
Profile created successfully! Path:
/Users/Joe/.zowe/profiles/db2/2e.yaml

host:      winmvs2e.hursley.ibm.com
port:      41100
user:      winchj
password:  *****
database:  DSNV112E

Review the created profile and edit if necessary using the profile update command.
Joe@MBP-3:~ Joe$
```

Usage

zowe profiles create db2-profile <profileName> [options]

Positional Arguments

profileName (*string*)

Specifies the name of the new db2 profile.

Options

--host | -H (*string*)

The Db2 server host name

--port | -P (*number*)

The Db2 server port number

--user | -u (*string*)

The Db2 user ID (may be the same as the TSO login)

--password | --pass | --pw (*string*)

The Db2 password (may be the same as the TSO password)

--database | -d (*string*)

The name of the database

zowe --hw

```
Joe --bash — 80x6
Joes-MBP-3:~ Joe$ zowe --hw
Launching web help in browser...
Joes-MBP-3:~ Joe$
```

The screenshot shows a web browser window with the URL `File | /Users/Joe/.zowe/web-help/index.html?p=zowe_db2`. The page title is "Zowe CLI". The left sidebar is highlighted with a red box and contains a tree view of the documentation structure:

- zowe
 - auth
 - cics
 - config
 - db2
 - plugins
 - profiles
 - provisioning | pv
 - rse-api-for-zowe-cli | rse
 - zos-console | console
 - zos-files | files
 - zos-jobs | jobs
 - zos-tso | tso
 - zos-uss | uss
 - zos-workflows | wf
 - zosconnect
 - zosmf

The main content area is titled "zowe" and contains the following text:

Welcome to Zowe CLI!

Zowe CLI is a command line interface (CLI) that provides a simple and streamlined way to interact with IBM z/OS.

For additional Zowe CLI documentation, visit <https://docs.zowe.org>

For Zowe CLI support, visit <https://www.zowe.org>

Groups

- [auth](#) - Connect to Zowe API ML authentication service
- [cics](#) - Interact with IBM CICS programs and transactions.
- [config](#) - Manage configuration and overrides
- [db2](#) - Interact with IBM Db2 for z/OS
- [plugins](#) - Install and manage plug-ins
- [profiles](#) - Create and manage configuration profiles
- [provisioning | pv](#) - Perform z/OSMF provisioning tasks
- [rse-api-for-zowe-cli | rse](#) - IBM RSE API Plug-in for Zowe CLI
- [zos-console | console](#) - Issue z/OS console commands and collect responses
- [zos-files | files](#) - Manage z/OS data sets
- [zos-jobs | jobs](#) - Manage z/OS jobs
- [zos-tso | tso](#) - Interact with TSO
- [zos-uss | uss](#) - Issue z/OS USS commands and receive responses
- [zos-workflows | wf](#) - Create and manage z/OSMF workflows
- [zosconnect](#) - z/OS Connect EE plugin for Zowe CLI
- [zosmf](#) - Interact with z/OSMF

Global Options

@zowe/cli 6.23.0

<https://www.openmainframeproject.org/projects/zowe/conformance>



| | | | | | |
|---|--|--|---------------------------------------|---|---|
| PHOENIX Software International® (E)JES | BROADCOM® CA Endeavor® | BROADCOM® CA Endeavor® Bridge For Git | BROADCOM® CA File Master™ Plus | BROADCOM® CA IDMS | BROADCOM® CA JCLCheck™ Workload Automation |
| Phoenix Software International | Broadcom | Broadcom | Broadcom | Broadcom | Broadcom |
| BROADCOM® CA MAT Analyze | BROADCOM® CA MAT Detect | BROADCOM® CA OPS/MVS® | BROADCOM® CA Spool™ | BROADCOM® CA SYSVIEW® Performance Management | BROADCOM® CA View® |
| Broadcom | Broadcom | Broadcom | Broadcom | Broadcom | Broadcom |
| BROADCOM® CA Workload Automation CA 7® | BROADCOM® CA z/OS Extended Files | BROADCOM® CA z/OS Extended Jobs | IBM Remote System Explorer API | OPEN MAINFRAME PROJECT Zowe IBM® CICS® Plug-in for Zowe CLI | OPEN MAINFRAME PROJECT Zowe IBM® Db2® Database Plug-in for Zowe CLI |
| Broadcom | Broadcom | Broadcom | IBM | Open Mainframe Project | Open Mainframe Project |
| OPEN MAINFRAME PROJECT Zowe IBM® TMS™ Plug-in for Zowe CLI | OPEN MAINFRAME PROJECT Zowe IBM MQ for z/OS Plug-in for Zowe CLI | IBM IBM® Z Workload Scheduler | IBM z/OS Connect EE | OPEN MAINFRAME PROJECT Zowe z/OS FTP Plug-in for Zowe CLI | IBM Zowe CLI CICS Deploy |
| Open Mainframe Project | Open Mainframe Project | IBM | IBM | Open Mainframe Project | IBM |
| OPEN MAINFRAME PROJECT Zowe Secure Credential Store Plug-in | | | | | |

Introducing base profiles

```
Joe — -bash — 80x27
Joes-MBP-3:~ Joe$ zowe profiles list
```

DESCRIPTION

List profiles of the type {{type}}

USAGE

zowe profiles list <command>

Where <command> is one of the following:

COMMANDS

| | |
|----------------------------------|--------------------------------------|
| base-profiles base | List profiles of the type base |
| cics-profiles cics | List profiles of the type cics |
| db2-profiles db2 | List profiles of the type db2 |
| rse-profiles rse | List profiles of the type rse |
| ssh-profiles ssh | List profiles of the type ssh |
| tso-profiles tso | List profiles of the type tso |
| zosconnect-profiles zosconnect | List profiles of the type zosconnect |
| zosmf-profiles zosmf | List profiles of the type zosmf |

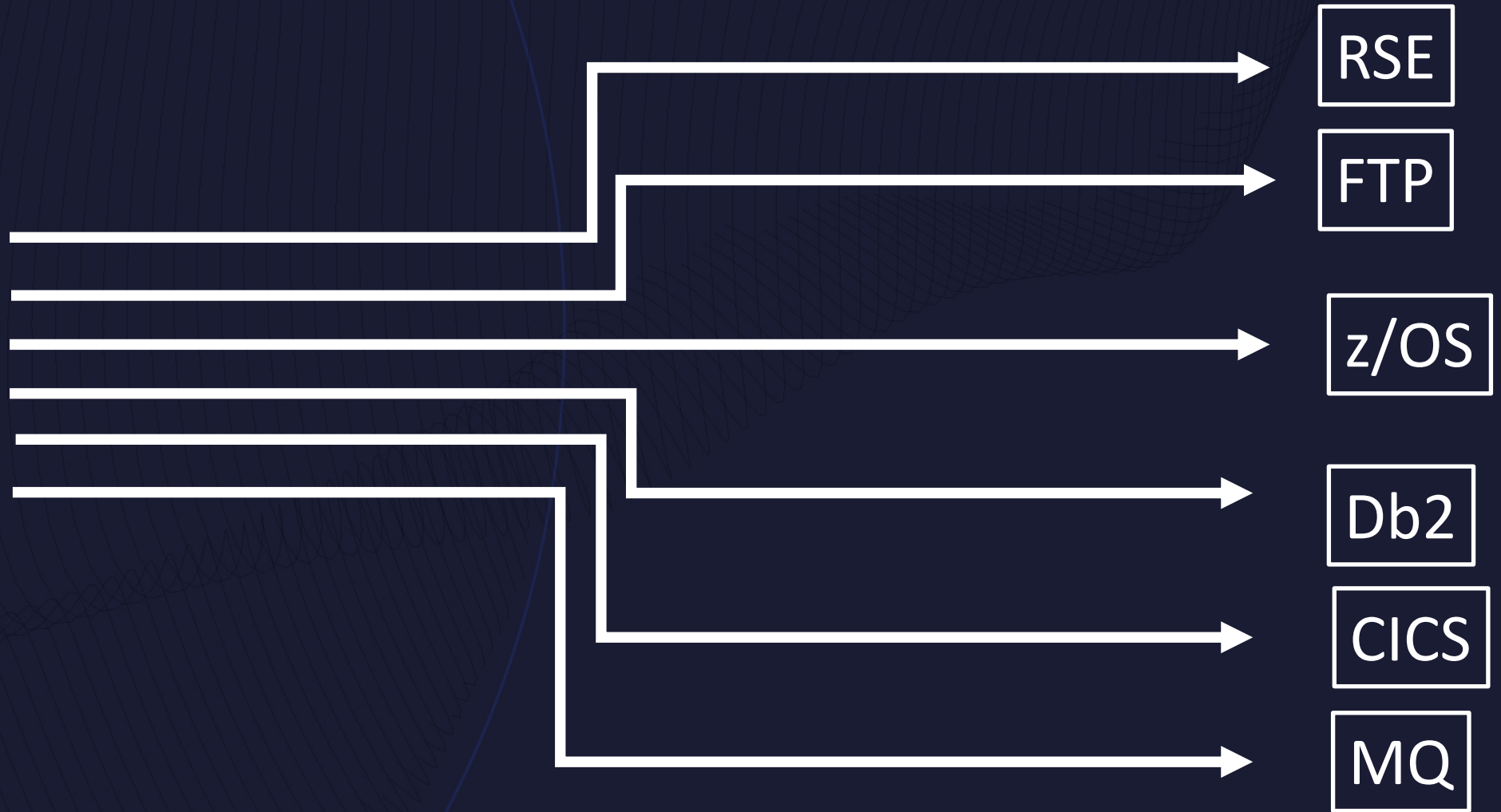
GLOBAL OPTIONS

Multiple profiles



z/OS

Multiple profiles



Base Profiles

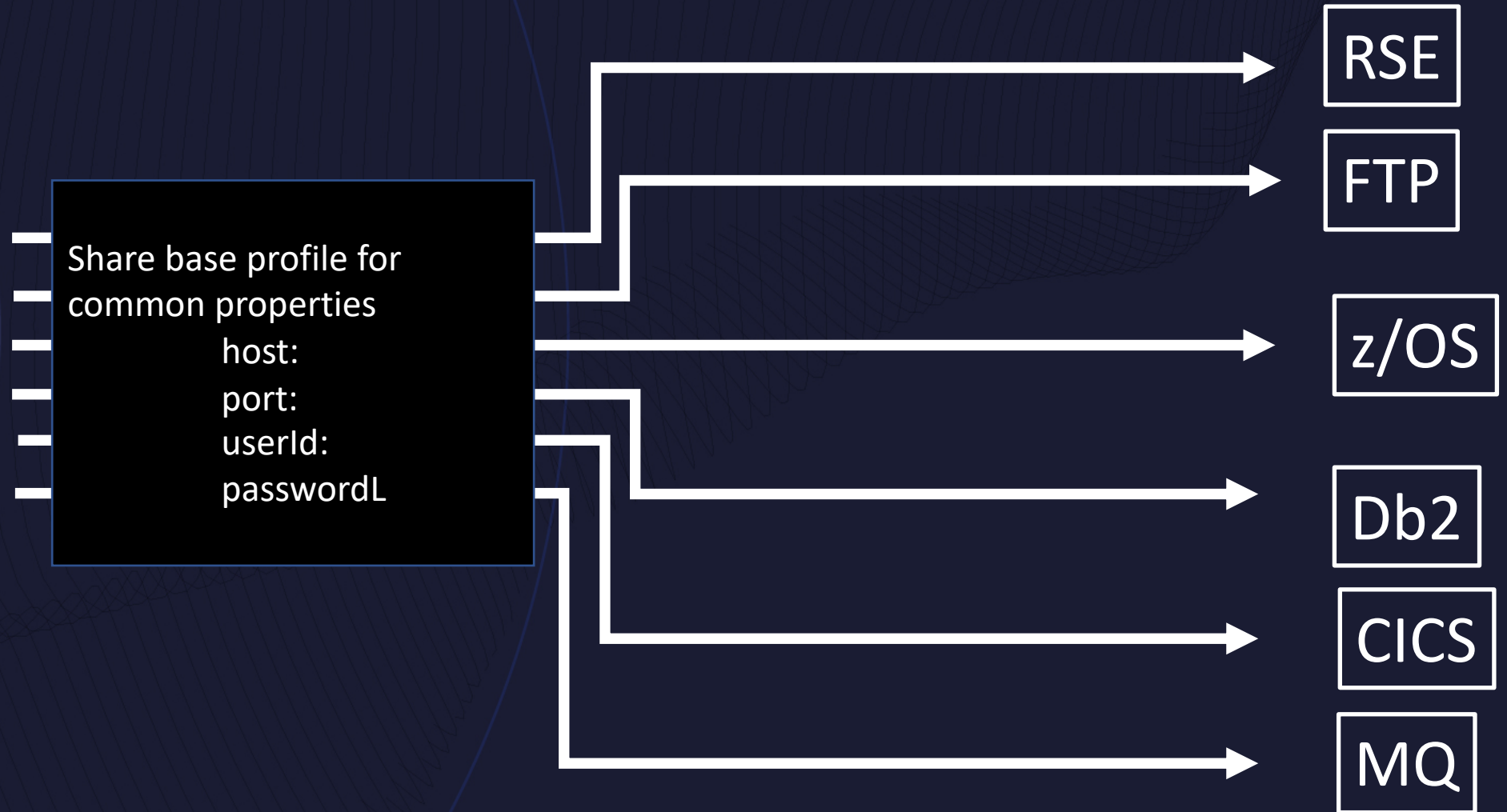
Base profile parent

```
name: default
host: tv5003.svl.ibm.com
port: 26502
rejectUnauthorized: false
userId: winchj
Password: *****
```

Individual Profile
Inherits where not
overridden

```
Name: myDb2
baseProfile: default
databaseName: DSNV112E
```

Multiple profiles with base profile



Token based authentication

```
Joe — node /usr/local/bin/zowe auth login apiml — 80x8
Joes-MBP-3:~ Joe$ zowe auth login apiml
Enter user name: winchj
Enter password : *****(masked)


```

```
Name: default
host: tv5003.svl.ibm.com
port: 26502
rejectUnauthorized: false
tokenType: apimlAuthenticationToken
tokenValue: eyJhbGciOiJSUzI1NiJ9.eyJzdWIiOiJ3aW5jaG9...

```

Multiple profiles with token based SSO



Share base profile for
common properties
host:
port:
authToken:

RSE

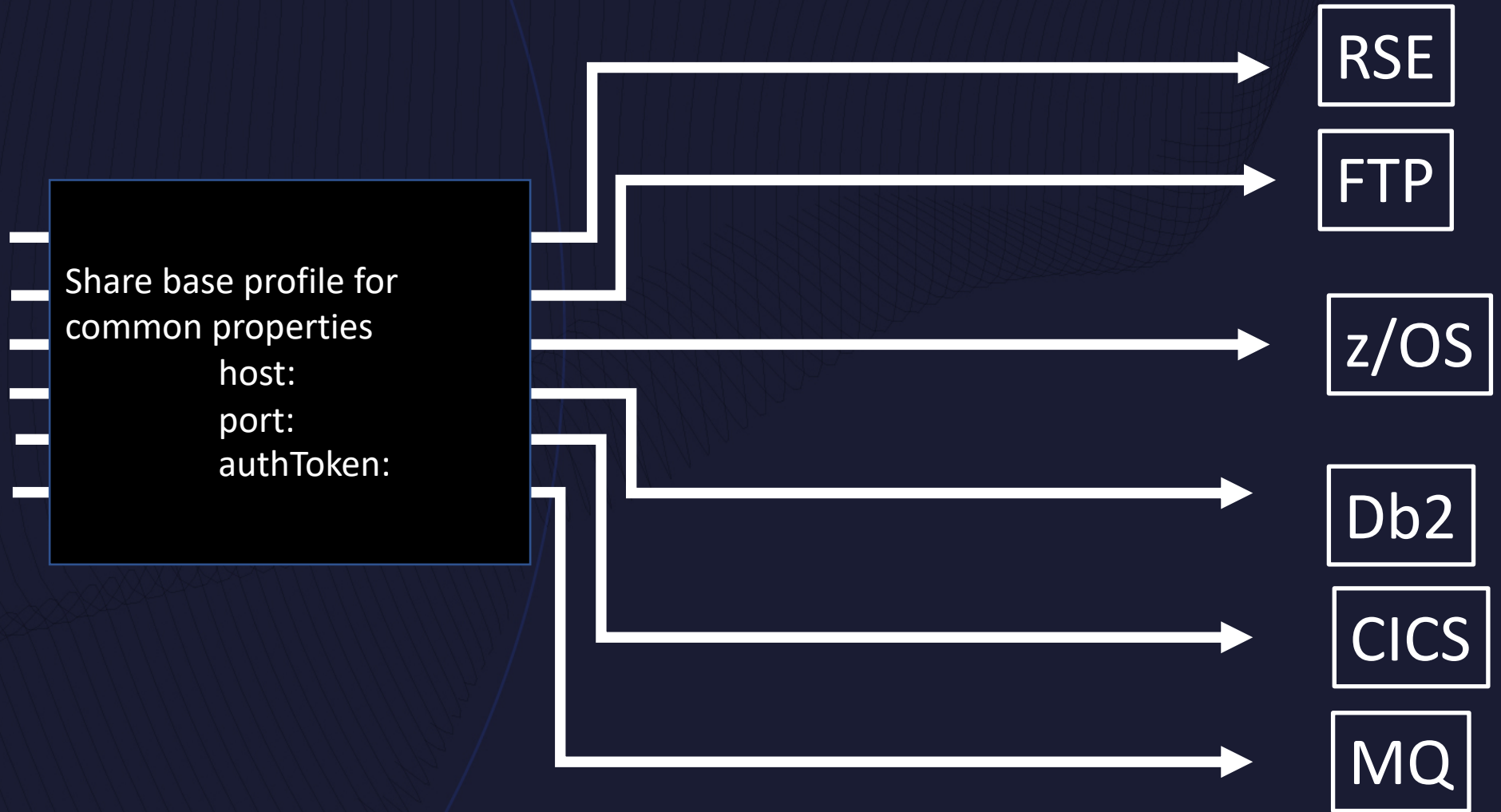
FTP

z/OS

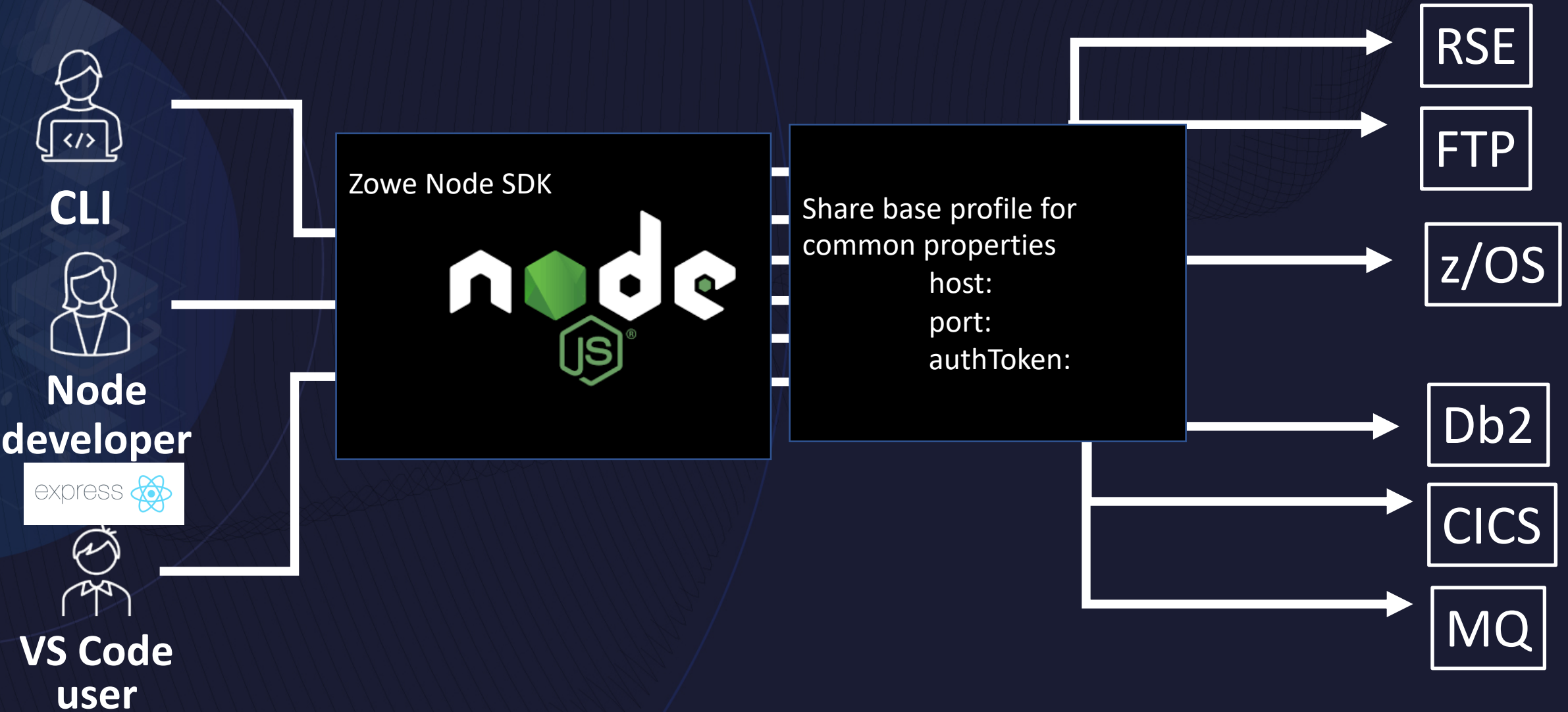
Db2

CICS

MQ



Zowe SDK



SDK Samples: <https://github.com/zowe/zowe-sdk-sample-scripts>

zowe / zowe-sdk-sample-scripts

<> Code 1 Issues 1 Pull requests ZenHub Actions Projects Wiki Security

master 1 branch 0 tags

Go to file Add file Code

dkelosky Merge pull request #3 from zowe/update-licenses ce23a9c on 10 Dec 2020 22 commits

- use-cases/node.js update licenses based on scan 3 months ago
- README.md Use the Node SDKs instead of the CLI 4 months ago

README.md

Zowe SDK Sample Scripts

This repository contains some sample scripts that utilize various components of the Zowe SDKs organized by use cases.

Use Cases

The `use-cases` directory has self-contained examples of various use cases that can utilize the Zowe SDKs.

Node.js:

- [SonarQube Data Set Analysis](#)
- [JCL Submission and Output Retrieval](#)
- [Assembling, Binding, and Linking Assembler Code](#)

```
33 const path = require("path");
34 const fs = require("fs");
35 const files = require("@zowe/zos-files-for-zowe-sdk");
36 const jobs = require("@zowe/zos-jobs-for-zowe-sdk");
37 const imperative = require("@zowe/imperative");
```

```
186 /*****
187 * Run JCL that was uploaded
188 *****/
189
190 let owner;
191 let jobid;
192 let jobname;
193
194 console.log(`Running JCL in file: ${properties.localFile}`);
195
196 const run = await jobs.SubmitJobs.submitJob(session, `${properties.dataset.dsn}${properties.dataset.member}`);
197 console.log("Run API response: ");
198 console.log(run);
199 console.log(`\n`);
200 owner = run.owner;
201 jobid = run.jobid;
202 jobname = run.jobname;
203 } catch (err) {
204 console.error(`Failed to run job: ${err.message}`)
205 process.exit(1);
206 }
207
208 /*****
209 * Let the job run for awhile
210 *****/
211
212 await delay(2000);
213
214 /*****
215 * Wait for the job to be out of input and execution
216 *****/
217
218 let status = "INPUT";
219 let checkNum = 0;
220 let response;
221 console.log(`Check job status`);
222
223 while (( status == "INPUT" || status == "ACTIVE" ) && checkNum < 100 ) {
224
225     checkNum = checkNum + 1;
226
227     response = await jobs.GetJobs.getJob(session, jobid);
228     status = response.status;
229     console.log( `Job ${jobid} status check #${checkNum}: ${status}` )
230 } catch (err) {
```

VS Code: <https://code.visualstudio.com/>

Visual Studio Code Docs Updates Blog API Extensions FAQ Learn

Search Docs Download

Version 1.54 is now available! Read about the new features and fixes from February.

Code editing. Redefined.

Free. Built on open source. Runs everywhere.

[Download Mac Universal Stable Build](#)

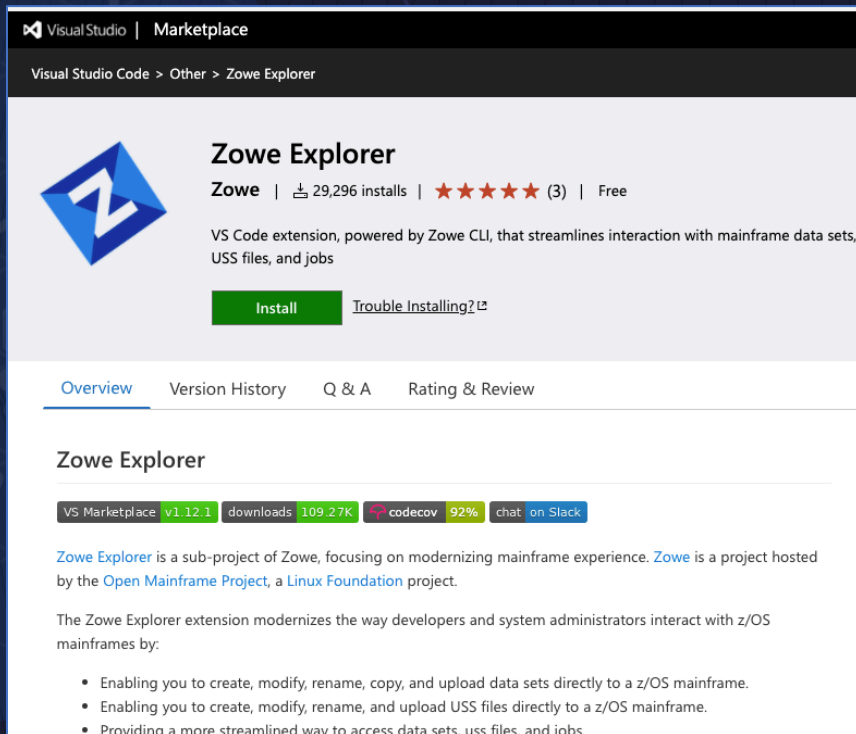
Other platforms and Insiders Edition

By using VS Code, you agree to its [license and privacy statement](#).

- Python 2019.6.24221 4.5★ 54.9M Microsoft [Install](#)
- GitLens — Git supercharge 9.8.5 23.1M 5★ Eric Amodio [Install](#)
- C/C++ 0.24.0 23M 3.5★ Microsoft [Install](#)
- ESLint 1.9.0 21.9M 4.5★ Dirk Baumer [Install](#)
- Debugger for C++ 4.11.6 20.6M 4★ Microsoft [Install](#)
- Language Support for Java 0.47.0 18.6M 4.5★ Red Hat [Install](#)
- vscod-icons 8.8.0 17.2M 5★ VSCoDe Icons Team [Install](#)
- Vetur 0.21.1 17M 4.5★ Pine Wu [Install](#)

IntelliSense Run and Debug Built-in Git Extensions

Visual Studio Marketplace Extension – Zowe Explorer



Visual Studio | Marketplace

Visual Studio Code > Other > Zowe Explorer

Zowe Explorer

Zowe | 29,296 installs | ★★★★★ (3) | Free

VS Code extension, powered by Zowe CLI, that streamlines interaction with mainframe data sets, USS files, and jobs

[Install](#) [Trouble Installing?](#)

[Overview](#) [Version History](#) [Q & A](#) [Rating & Review](#)

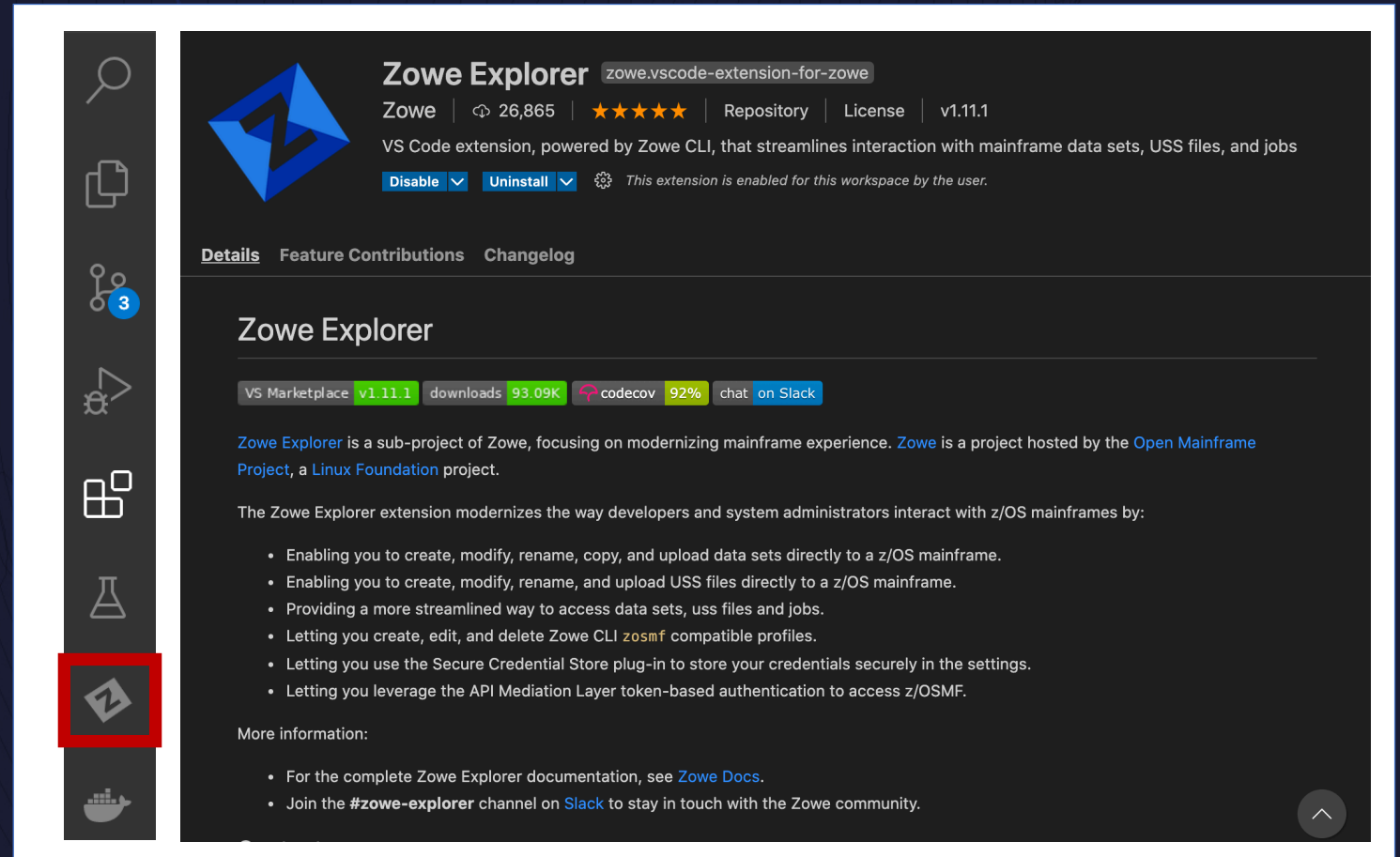
Zowe Explorer

VS Marketplace v1.12.1 | downloads 109.27K | codecov 92% | chat on Slack

Zowe Explorer is a sub-project of Zowe, focusing on modernizing mainframe experience. Zowe is a project hosted by the [Open Mainframe Project](#), a [Linux Foundation](#) project.

The Zowe Explorer extension modernizes the way developers and system administrators interact with z/OS mainframes by:

- Enabling you to create, modify, rename, copy, and upload data sets directly to a z/OS mainframe.
- Enabling you to create, modify, rename, and upload USS files directly to a z/OS mainframe.
- Providing a more streamlined way to access data sets, uss files, and jobs.



Zowe Explorer

zowe.vscode-extension-for-zowe

Zowe | 26,865 | ★★★★★ | Repository | License | v1.11.1

VS Code extension, powered by Zowe CLI, that streamlines interaction with mainframe data sets, USS files, and jobs

[Disable](#) [Uninstall](#) [Settings](#) This extension is enabled for this workspace by the user.

[Details](#) [Feature Contributions](#) [Changelog](#)

Zowe Explorer

VS Marketplace v1.11.1 | downloads 93.09K | codecov 92% | chat on Slack

Zowe Explorer is a sub-project of Zowe, focusing on modernizing mainframe experience. Zowe is a project hosted by the [Open Mainframe Project](#), a [Linux Foundation](#) project.

The Zowe Explorer extension modernizes the way developers and system administrators interact with z/OS mainframes by:

- Enabling you to create, modify, rename, copy, and upload data sets directly to a z/OS mainframe.
- Enabling you to create, modify, rename, and upload USS files directly to a z/OS mainframe.
- Providing a more streamlined way to access data sets, uss files and jobs.
- Letting you create, edit, and delete Zowe CLI `zosmf` compatible profiles.
- Letting you use the Secure Credential Store plug-in to store your credentials securely in the settings.
- Letting you leverage the API Mediation Layer token-based authentication to access z/OSMF.

More information:

- For the complete Zowe Explorer documentation, see [Zowe Docs](#).
- Join the [#zowe-explorer](#) channel on [Slack](#) to stay in touch with the Zowe community.

Sidebar: A vertical sidebar on the left contains several icons. The icon for Zowe Explorer, which is a blue square with a white 'Z', is highlighted with a red rectangular box.

70WF

- DATA SETS
 - Favorites
 - 3b
 - WINCHJ.BORK.SZWEAUTH
 - WINCHJ.BORK.SZWESAMP
 - WINCHJ.COBOL
 - FISH
 - SAMPLE
 - WINCHJ.JCL
 - BEER
 - BOBBY
 - CAT
 - COFFEE
 - COPYJOB
 - UNIX SYSTEM SERVICES (USS)
 - 1.0.0
 - 1.1.0
 - 1.13.0_1177
 - 1.13.0_1316
 - 1.14.0
 - runtime
 - bin
 - components
 - fingerprint
 - manifest.json
 - scripts
 - workflows
 - zowe-1.14.0
 - zowe-1.14.0.pax
 - 1.5.0
 - ims
 - zowe
 - JOBS
 - Favorites
 - 3b
 - WINCHJ(TSU13547) - ABEND S222
 - WINCHJ(TSU13696) - ABEND S222
 - IZUFPROC:SYSOUT(102)
 - WINCHJ(TSU13697) - ABEND S222
 - WINCHJ(TSU14609) - ABEND S222
 - WINCHJ(TSU14669) - ABEND S222
 - WINCHJ(TSU14665) - ABEND S222

- Full from mainframe
- Submit Job
- Add to Favorites
- Copy
- Edit
- Rename
- Delete Member

```

Users > Joe > .vscode > extensions > zowe.vscode-extension-for-zowe-1.8.0 > resources > temp > _D_ > 3b > WINCHJ.COBOL(SAMPLE).cbl > {} PROGRAM: AWIXMP
1 *****
2 * This program demonstrates the following Language *
3 * Environment callable *
4 * services : CEEMOUT, CEELCOT, CEEDATE *
5 *****
6 ** I D D I V I S I O N **
7 Identification Division.
8 Program-id. AWIXMP.
9 *****
10 ** D A T A D I V I S I O N **
11 *****
12 Data Division.
13 Working-Storage Section.
14 *****
15 ** Declarations for the local date/time service.
16 *****
17 01 Feedback.
18 COPY CEEIGZCT
19 02 Fb-severity PIC 9(4) Binary.
20 02 Fb-detail PIC X(10).
21 77 Dest-output PIC S9(9) Binary.
22 77 Lildate PIC S9(9) Binary.
23 77 Lilsecs COMP-2.
24 77 Greg PIC X(17).
25 *****
26 ** Declarations for messages and pattern for date formatting.
27 *****
28 01 Pattern.
29 02 PIC 9(4) Binary Value 45.
30 02 PIC X(45) Value
31 "Today is Wwwwwwwwwwz, Mmmmmmmmmz ZD, YYYY.".
32 77 Start-Msg PIC X(80) Value
33 "Callable Service example starting.".
34 77 Ending-Msg PIC X(80) Value
35 "Callable Service example ending.".
36
37
38
39

```

PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL 2: bash

```

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
Joes-MBP-3:docs-site Joe$ pwd
/Users/Joe/Zowe/repos/docs-site
Joes-MBP-3:docs-site Joe$ git pull
Already up to date.
Joes-MBP-3:docs-site Joe$ git fetch
Joes-MBP-3:docs-site Joe$

```

Data Sets

Select a filter

- + Create a new filter. Comma separate multiple entries (pattern 1, pattern 2, ...)
- WINCHJ.*
- TSTRADM.*
- USER.PROCLIB
- ADCD.Z23B.PROCLIB
- FEU.Z23B.PROCLIB

Job submitted JOB15530

ZOWE

DATA SETS

- Favorites
 - [3b]: WINCHJ.JCL
 - [ukzowe]: TSTRADM.SOW1.ISPF.ISPPROF
- 3b
 - WINCHJ.BORK.SZWEAUTH
 - WINCHJ.BORK.SZWESAMP
 - WINCHJ.COBOLE
 - WINCHJ.JCL**
 - BEER
 - BOBBY
 - CAT
 - COFFEE
 - COPYJOB
 - CRICKET
 - FISH
 - FRIDAY
 - FROM
 - JIM2020

- Create New Member
- Upload Member...
- Show Data Set Attributes
- Migrate Data Set
- Add to Favorites
- Paste
- Rename
- Delete PDS

WINCHJ.JCL

- BEER
- BOBBY
- CAT
- COFFEE
- COPYJOB
- CRICKET
- FISH
- FRIDAY
- FROM
- JIM2020

- Pull from Mainframe
- Submit Job**
- Add to Favorites
- Copy
- Edit
- Rename
- Delete Member

Jobs

WIN

+ Owner/Prefix Job Search

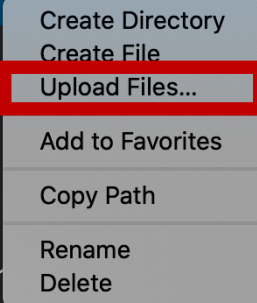
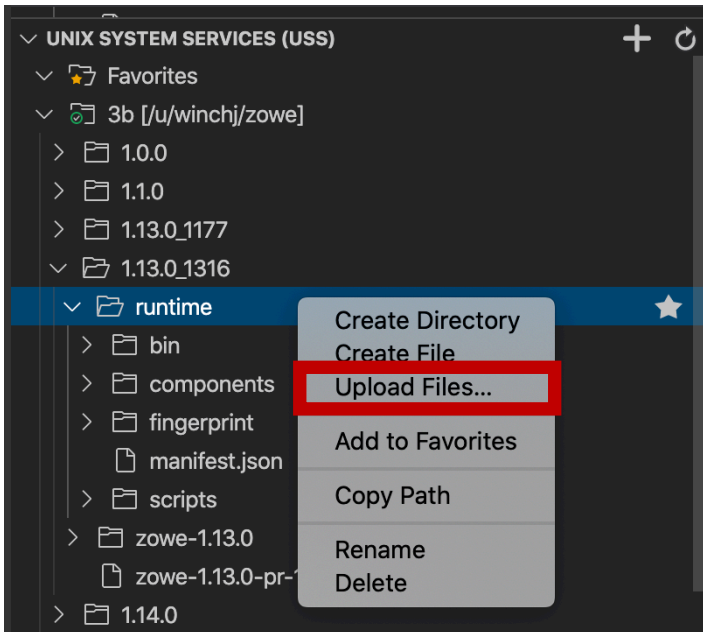
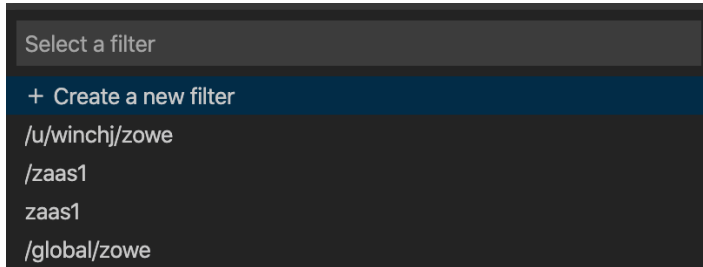
+ Job Id search

JOBS

- Favorites
- 3b
 - ZWEJ1SV(STC13644) - SYS FAIL
 - ZWES1SV(STC14640) - SYS FAIL
 - ZWES1STC(STC14660) - SYS FAIL
 - ZWES2SV(STC15109) - SYS FAIL
 - ZWEJ1SV(STC15483) - PENDING
 - ZWEJ1SV:STDOUT(101)** (Context Menu)
 - Get JCL
 - Issue Modify command
 - Issue Stop command
 - ZWEJ1SV:STDERR(102)
 - JES2:JESMSG LG(103)
 - JES2:JESYSMSG(104)
 - ZWES1STC(STC15502) - ACTIVE
 - ZWEJ1SV(STC15506) - ACTIVE
 - iovana

```
ZWEJ1SV:STC15483.STDOUT x
rc=0
1 .ng JAVA_HOME/bin to the PATH...
2 .ng NODE_HOME/bin to the PATH...
3 2.18.0 is supported.
4 .ng if node bin is functional...
5 n is functional
6 .fully checked z/OS MF is available on 'https://winmvs3b.hursley.ibm.com:32070/zosmf/info'
7 rersion 1.8.0_251 is supported
8 .fully checked z/OS MF is available on 'https://winmvs3b.hursley.ibm.com:32070/zosmf/info'
9 rersion 1.8.0_251 is supported
10 .fully checked z/OS MF is available on 'https://winmvs3b.hursley.ibm.com:32070/zosmf/info'
11 rersion 1.8.0_251 is supported
12 2.18.0 is supported.
13 .ng if node bin is functional...
14 n is functional
15 2.18.0 is supported.
16 .ng if node bin is functional...
17 n is functional
18 2.18.0 is supported.
19 .ng if node bin is functional...
20 n is functional
21 .ng node exists
22 p/IBM/cnj/v12r0/IBM/node-latest-os390-s390x/bin/node is /usr/lpp/IBM/cnj/v12r0/IBM/node-latest-os390-s390x/bin/n
23 | app-server plugin installer. Log=/u/winchj/zowe-instance-dir/logs/install-app.log
24 path=/u/winchj/zowe/1.14.0/runtime/components/app-server/share/zlux-server-framework/utis
25 :h=/u/winchj/zowe-instance-dir/workspace/api-catalog
26 :th=/u/winchj/zowe-instance-dir/workspace/app-server/serverConfig/server.json
27 |-08 12:19:30.968 <ZWED:65939> ZWESVUSR INFO (_zsf.utils,util.js:288) ZWED0051I /u/winchj/zowe-instance-dir/works
28 |-08 12:19:30.969 <ZWED:65939> ZWESVUSR INFO (_zsf.utils,util.js:288) ZWED0051I /u/winchj/zowe-instance-dir/works
29 |-08 12:19:30.970 <ZWED:65939> ZWESVUSR INFO (_zsf.utils,util.js:288) ZWED0051I /u/winchj/zowe-instance-dir/works
30 |-08 12:19:30.971 ZWED0109I - Registering App (ID=org.zowe.api.catalog) with App Server
31 |-08 12:19:30.974 ZWED0110I - App org.zowe.api.catalog installed to /u/winchj/zowe-instance-dir/workspace/api-ca
32 |with rc=0
33 .ng node exists
34 p/IBM/cnj/v12r0/IBM/node-latest-os390-s390x/bin/node is /usr/lpp/IBM/cnj/v12r0/IBM/node-latest-os390-s390x/bin/n
35 | app-server plugin installer. Log=/u/winchj/zowe-instance-dir/logs/install-app.log
36 path=/u/winchj/zowe/1.14.0/runtime/components/app-server/share/zlux-server-framework/utis
37 :h=/u/winchj/zowe-instance-dir/workspace/explorer-jes
38 :th=/u/winchj/zowe-instance-dir/workspace/app-server/serverConfig/server.json
39 |-08 12:19:32.582 <ZWED:65942> ZWESVUSR INFO (_zsf.utils,util.js:288) ZWED0051I /u/winchj/zowe-instance-dir/works
```

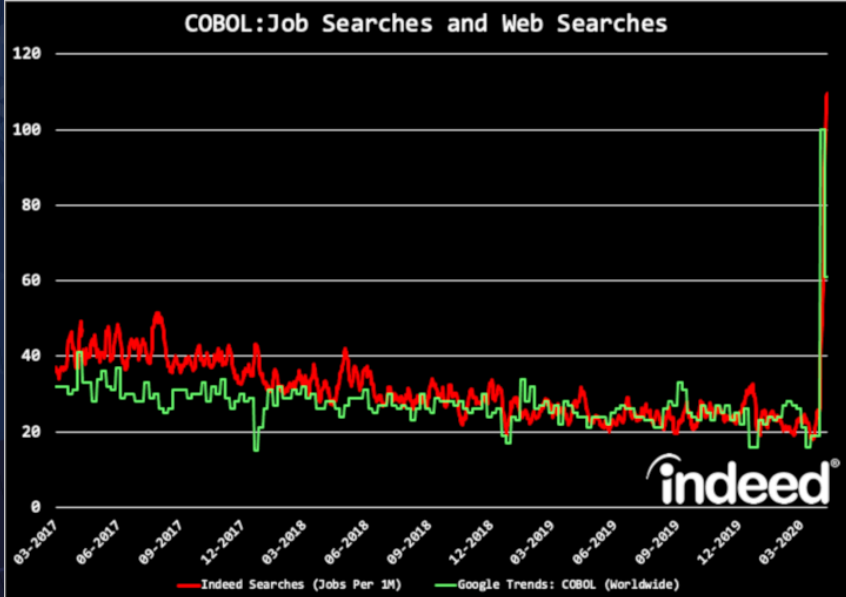
Unix System Services



```
zowe-install.sh x
Users > Joe > .vscode > extensions > zowe.vscode-extension-for-zowe-1.11.1 > resources > temp > _U_ > 3b > u > winchj > zowe-1.19.0 > install > zowe-ins
23 # -f: log file. This is optional. It provides option to direct all installation
24 #   logs into one file. This option is deprecated in favor of "-l" option.
25 #####
26
27 #####
28 # Functions
29 separator() {
30 |   echo "-----"
31 | }
32
33 usage() {
34 |   if [ "${RUN_ON_ZOS}" = "true" ]; then
35 |     echo "Usage: $0 -i <zowe_install_path> -h <zowe_dsn_prefix> [-l <log_directory>]"
36 |   else
37 |     echo "Usage: $0 -i <zowe_install_path> [-l <log_directory>]"
38 |   fi
39 |   exit 1
40 | }
41
42 show_usage_error_and_exit() {
43 |   message=$1
44 |
45 |   echo "Error: ${message}" >&2
46 |   usage
47 | }
48
49 prepare_temp_dir() {
50 |   # Create a temp directory to be a working directory for sed replacements and logs, if install_dir is read-on
51 |   if [[ -w "${INSTALL_DIR}" ]]
52 |   then
53 |     export TEMP_DIR=${INSTALL_DIR}/temp_`date +%Y-%m-%d`"
54 |   else
55 |     export TEMP_DIR=${TMPDIR:-/tmp}/zowe_`date +%Y-%m-%d`"
56 |   fi
57 |   mkdir -p $TEMP_DIR
58 |   chmod a+rxw $TEMP_DIR
59 | }
60
61 prepare_log_file() {
62 |   if [[ -z "${LOG_FILE}" ]]
63 |   then
64 |     set_install_log_directory "${LOG_DIRECTORY}"
65 |     validate_log_file_not_in_root_dir "${LOG_DIRECTORY}" "${ZOWE_ROOT_DIR}"
66 |     set_install_log_file "zowe-install"
67 |   else
68 |     set_install_log_file_from_full_path "${LOG_FILE}"
69 |     validate_log_file_not_in_root_dir "${LOG_FILE}" "${ZOWE_ROOT_DIR}"
70 |   fi

```

COBOL programmers are in demand to fight the coronavirus pandemic

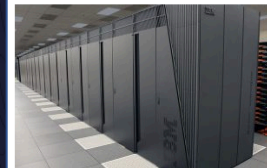


Home > Latest News > Business > COBOL given a new life by COVID-19

Latest News Business Public Sector

COBOL given a new life by COVID-19

By Ian Murphy · April 29, 2020



COBOL, the language that refuses to die, has been given another lease of life by COVID-19. Research from Job Site, [Indeed](#), shows a significant surge in demand for COBOL programmers. Driving this is the need for mainframe programmers to work on older

systems that run on mainframes, used by organisations in both the public and private sector.

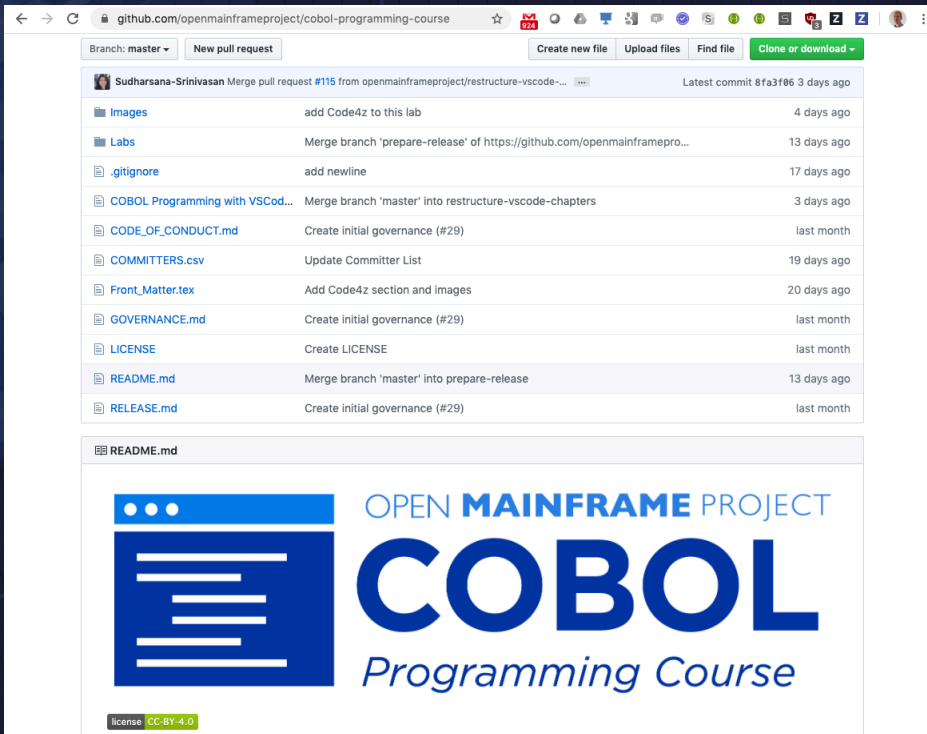
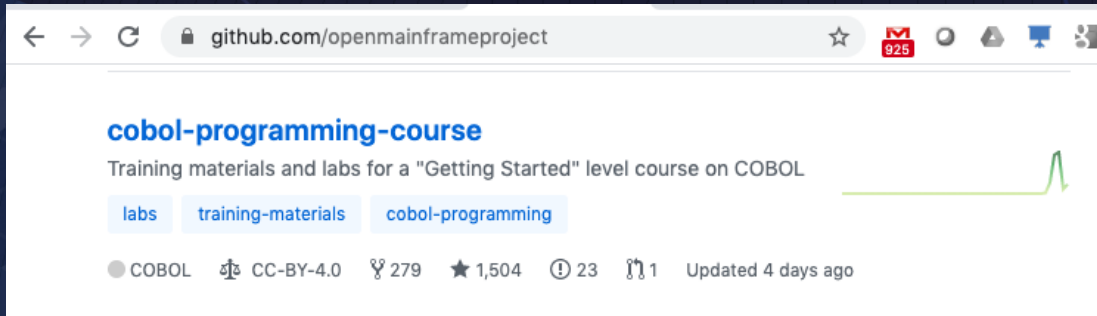
Indeed passed its data to NTT DATA UK to analyse. It compared that data with similar data from Google Trends. [Simon Williams](#), CEO of [NTT DATA UK](#), said: "It comes as no surprise that there has been a surge in interest in COBOL. There are a lot of legacy systems in operation, reliant on legacy code. This has to be a wakeup call – if organisations don't have the skills to maintain their code, they need to modernise."



Simon Williams, CEO of NTT DATA UK

Working with the Linux Foundation's Open Mainframe Project, IBM stood up an employment and volunteer forum and a technical forum on Thursday to match COBOL programmers with state agencies that are looking for additional help. The "[Calling all COBOL Programmers](#)" forum asks volunteers or those looking for a job to list their name, location availability and resume so that government agencies can reach out if they fit a specific need. The [technical forum](#) is available for government agencies to get "extra reassurance" on the 61-year-old programming language.

"[IBM] knew which states were going to be hotspots, and we proactively reached out to each of those states," said Barry Baker, vice president of IBM Z Software, a mainframe platform that many states run unemployment systems on.



Syntax Highlighting and Coloring

The Code4z package enables coloring of keywords, paragraphs, and variables in different colors to make the code easier to navigate.

Syntax and Semantic Check

The COBOL Language Support extension checks for mistakes and errors in COBOL code. The syntax check feature reviews the whole content of the code, highlights errors and suggests fixes.

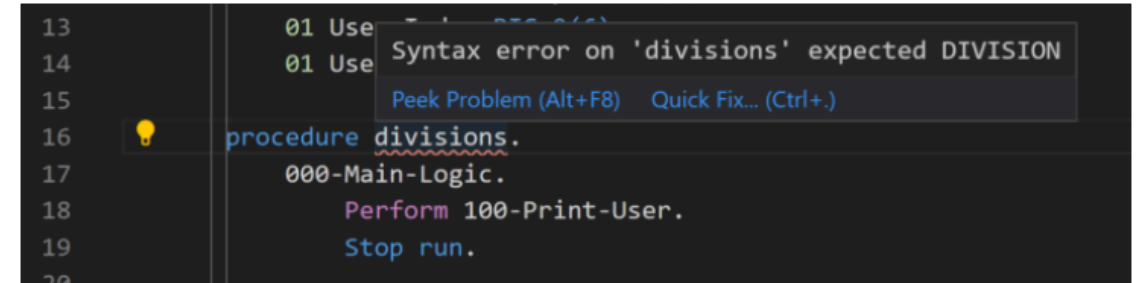


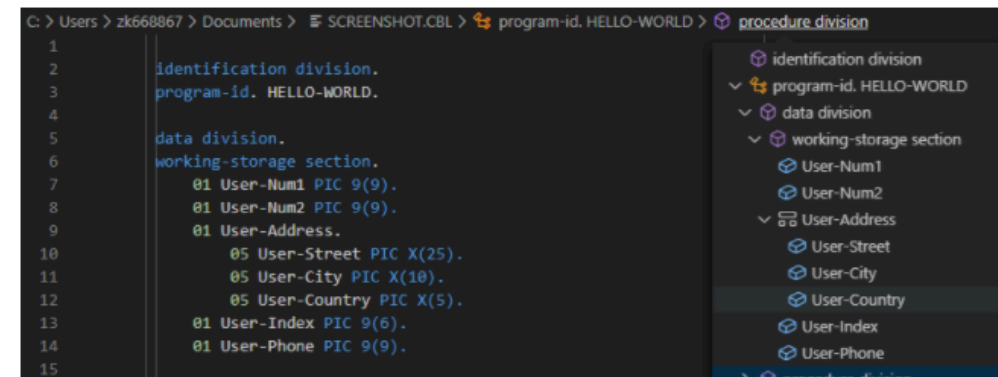
Figure 1. The syntax and semantic check feature highlights an error.

Navigation of Code

The Code4z package enables several features for ease of navigation through code.

Breadcrumb View

The breadcrumb view across the top of the editor shows where the current line of code exists within the structure of the COBOL source code. You can click each element on the bar to see that section of the code highlighted, or to select a code element within the section and navigate to it.



COBOL

Articles
Code Patterns
Series
Tutorials
Videos

Community
Blog Posts
Events

Related topics
Find an expert
COBOL technical questions forum
Enterprise COBOL for Z/OS documentation library
Enterprise COBOL for Z/OS knowledge center
IBM Z and LinuxONE
Community COBOL topic group

COBOL Fridays: Get hands-on - Run a COBOL program

Global Student Hub

Site feedback

WEBINAR:
Get Hands-on!
Run a COBOL
program

IBM
Z Ecosystem Team

COBOL Fridays 2: Get hands on! Run a COBOL program

```

1  *A B
2  IDENTIFICATION DIVISION.
3  PROGRAM-ID. PAYROL00
4  DATA DIVISION.
5  WORKING-STORAGE SECTION.
6  ***** Variables for the report
7  77 WHO      PIC X(15).
8  77 WHERE    PIC X(20).
9  77 WHY      PIC X(20).
10 77 RATE     PIC 9(3).
11 77 HOURS    PIC 9(3).
12 77 GROSS-PAY PIC 9(5).
13
14 PROCEDURE DIVISION.
15 ***** COBOL MOVE statements - Literal Text to Variables
16 MOVE "Captain COBOL" TO WHO.
17 MOVE "San Jose, California" TO WHERE.
18 MOVE "Learn to be a COBOL expert" TO WHY.
19 MOVE 19 TO HOURS.
20 MOVE 23 TO RATE.
21 ***** Calculation using COMPUTE reserved word verb
22 COMPUTE GROSS-PAY = HOURS * RATE.
23 ***** DISPLAY statements
24 DISPLAY "Name: " WHO.
25 DISPLAY "Location: " WHERE.
26 DISPLAY "Reason: " WHY.
27 DISPLAY "Hours Worked: " HOURS.
28 DISPLAY "Hourly Rate: " RATE.
29 DISPLAY "Gross Pay: " GROSS-PAY.
30 DISPLAY WHY " from " WHO.
  
```

Visual Studio Code interface showing the file explorer on the left with a tree view of project files including 'LearnCOBOL', 'Z80573', and 'PAYROL00'. The main editor window shows the COBOL source code for 'PAYROL00.CBL'. A small video inset in the bottom right corner shows a presenter.

IBM
COBOL FRIDAYS

How to Connect COBOL Applications with APIs?

09.04.2020

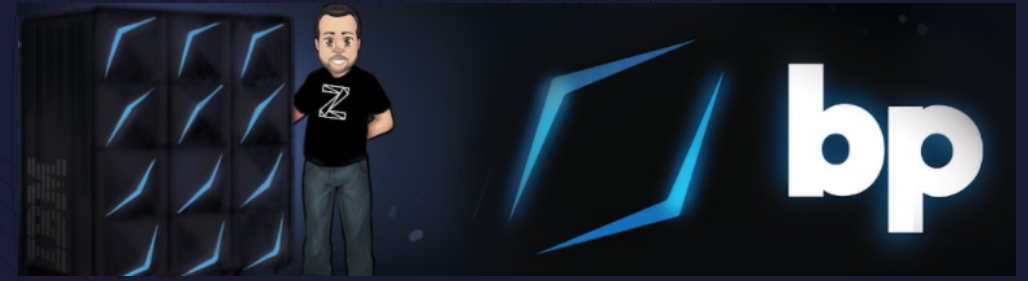
Video | Tech Talks

COBOL Fridays: Connect COBOL applications with APIs

September 4, 2020

→

9:01



```
//JES1JOB2 JOB 1
//*
//* JCL line 5 needs a SECRET= value to execute
//*
//SETVAR SET SECRET=MYSECRET
//SECRET EXEC PGM=IEBGENER
//SYSPRINT DD DUMMY
//SYSUT1 DD DSN=MTM200.PUBLIC.WORK(&SECRET),
// DISP=SHR
//SYSUT2 DD DSN=&SYSUID..OUTPUT(JES1JOB2),
// DISP=SHR
//SYSIN DD DUMMY
```

YouTube master the mainframe

right corner here this new area this output terminal area pops up and it says

#ibmzday #MastertheMainframe
VS Code and Plugins for Master the Mainframe 2020

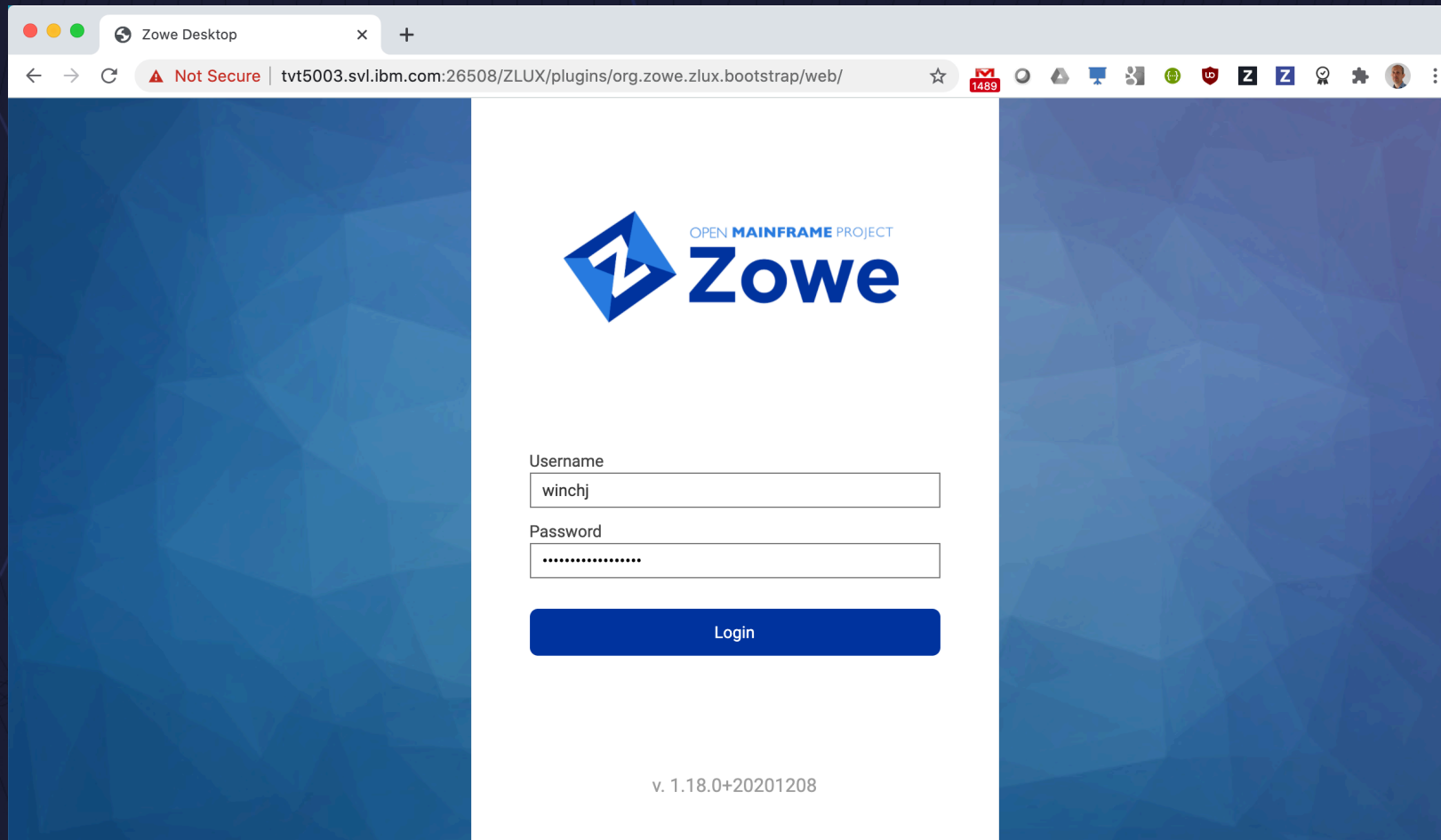
Bill Pereira
453 subscribers

HOME VIDEOS PLAYLISTS CHANNELS DISCUSSION

Created playlists

- Master the Mainframe 2020 (6 videos)
- Ansible (3 videos)
- Linux on Z! (3 videos)
- MIM 2019 (9 videos)
- Lives (7 videos)
- Zowe - Demos (10 videos)

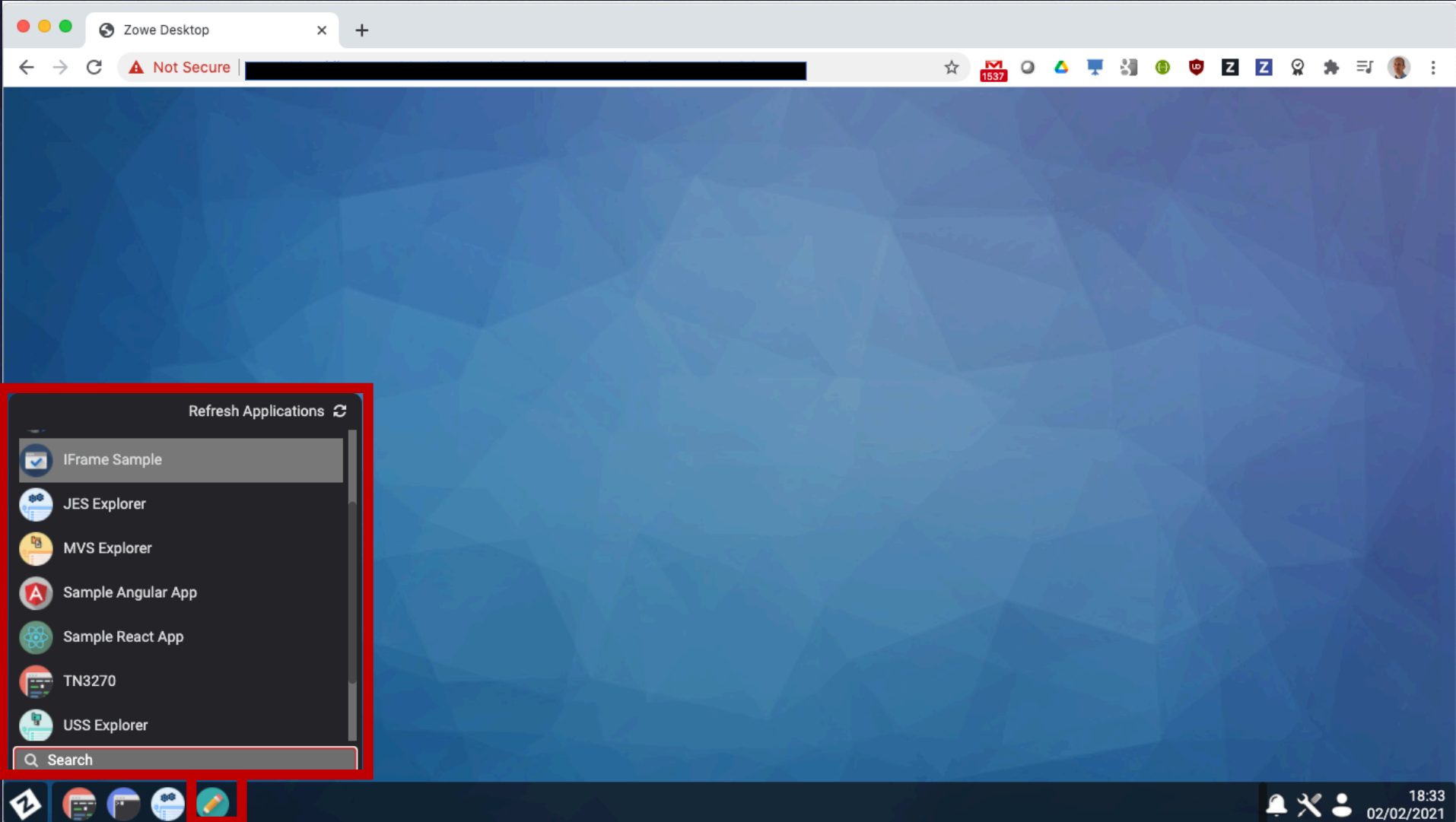
Zowe Desktop



The screenshot shows a web browser window with the following details:

- Browser Tab:** Zowe Desktop
- Address Bar:** Not Secure | tvt5003.svl.ibm.com:26508/ZLUX/plugins/org.zowe.zlux.bootstrap/web/
- Page Content:**
 - Logo:** Zowe logo with the text "OPEN MAINFRAME PROJECT" above it.
 - Form Fields:**
 - Username:** Input field containing "winchj".
 - Password:** Input field with masked characters ".....".
 - Login Button:** A blue button labeled "Login".
 - Version:** v. 1.18.0+20201208

App launch bar



File Editor

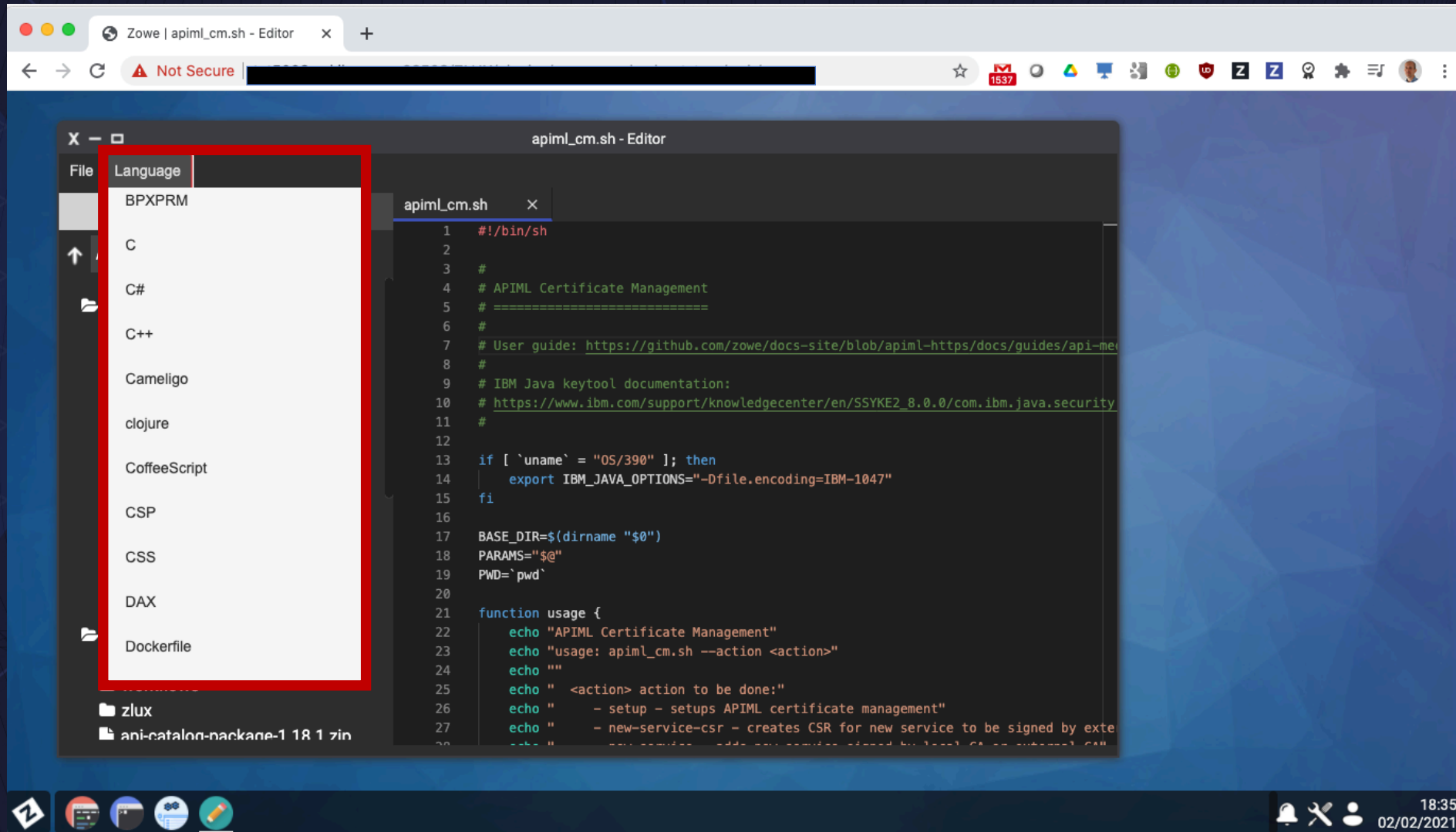
The screenshot displays a web browser window with a file editor interface. The browser's address bar shows "Zowe | apiml_cm.sh - Editor" and a "Not Secure" warning. The editor interface is divided into two main sections:

- File Explorer (Left):** Shows a directory tree for "/u/winchj/zowe-1.18.0". The "bin" directory is expanded, and "apiml_cm.sh" is selected and highlighted with a red box. Other files in the "bin" directory include "zowe-configure-instance.sh", "zowe-generate-checksum.sh", "zowe-init.sh", "zowe-setup-certificates.env", "zowe-setup-certificates.sh", "zowe-setup-keyring-certificates.env", "zowe-support.sh", and "zowe-verify-authenticity.sh".
- Code Editor (Right):** Displays the content of "apiml_cm.sh". The code is as follows:

```
1 #!/bin/sh
2
3 #
4 # APIML Certificate Management
5 # =====
6 #
7 # User guide: https://github.com/zowe/docs-site/blob/apiml-https/docs/guides/api-me
8 #
9 # IBM Java keytool documentation:
10 # https://www.ibm.com/support/knowledgecenter/en/SSYKE2\_8.0.0/com.ibm.java.security
11 #
12
13 if [ `uname` = "OS/390" ]; then
14     export IBM_JAVA_OPTIONS="-Dfile.encoding=IBM-1047"
15 fi
16
17 BASE_DIR=$(dirname "$0")
18 PARAMS="$@"
19 PWD=`pwd`
20
21 function usage {
22     echo "APIML Certificate Management"
23     echo "usage: apiml_cm.sh --action <action>"
24     echo ""
25     echo "  <action> action to be done:"
26     echo "    - setup - setups APIML certificate management"
27     echo "    - new-service-csr - creates CSR for new service to be signed by exte
28     echo "    - new-service - adds new service signed by local CA or external CA"
```

The system tray at the bottom right shows the time as 18:34 on 02/02/2021.

File Content types



The image shows a web browser window with a code editor. The browser's address bar shows "Zowe | apiml_cm.sh - Editor" and "Not Secure". The code editor has a "Language" dropdown menu open, listing various programming languages. The code in the editor is a shell script for "apiml_cm.sh".

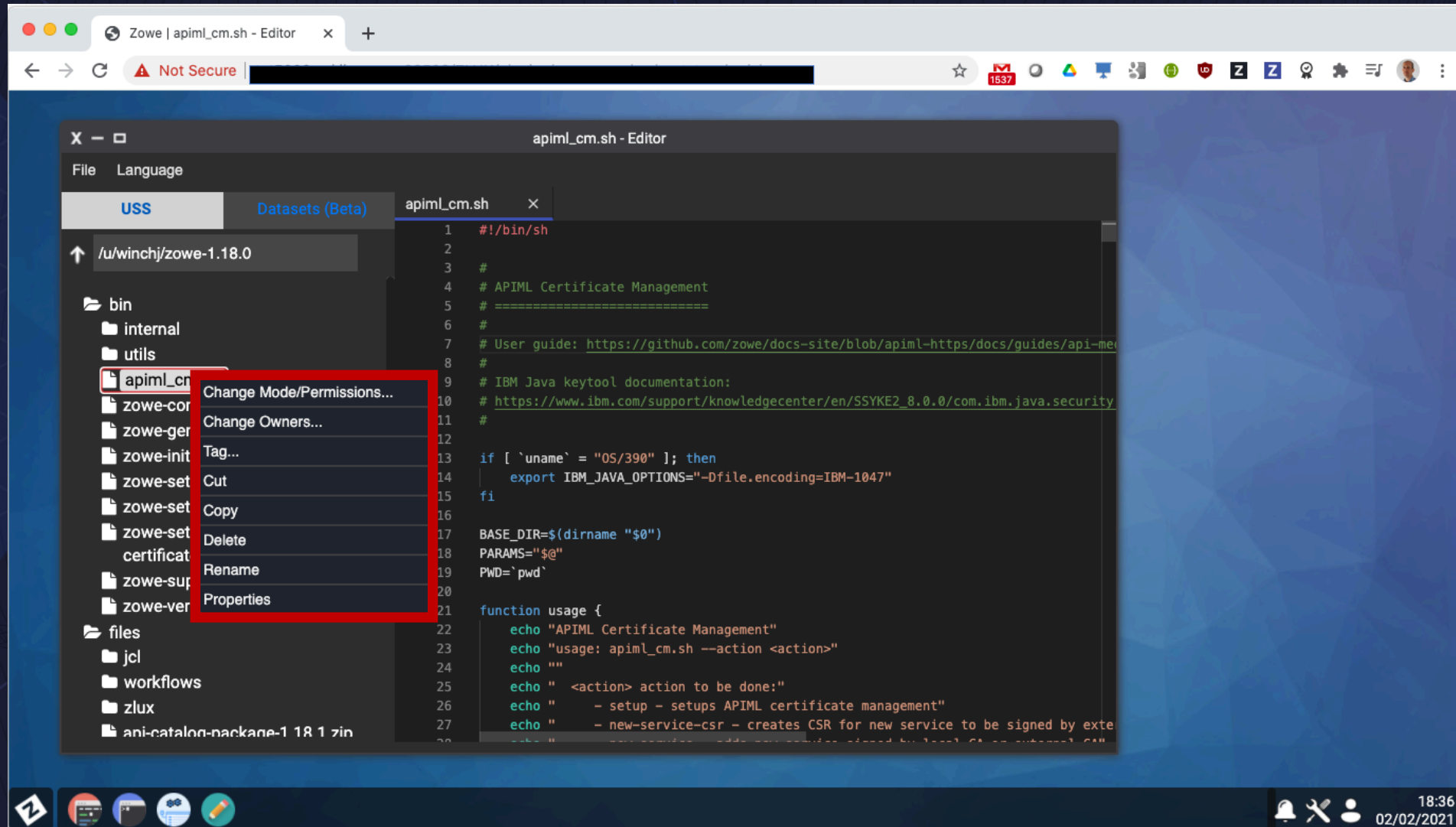
Language Selection Menu:

- BPXPRM
- C
- C#
- C++
- Cameligo
- clojure
- CoffeeScript
- CSP
- CSS
- DAX
- Dockerfile

Code Editor Content (apiml_cm.sh):

```
1 #!/bin/sh
2
3 #
4 # APIML Certificate Management
5 # =====
6 #
7 # User guide: https://github.com/zowe/docs-site/blob/apiml-https/docs/guides/api-me
8 #
9 # IBM Java keytool documentation:
10 # https://www.ibm.com/support/knowledgecenter/en/SSYKE2_8.0.0/com.ibm.java.security
11 #
12
13 if [ `uname` = "OS/390" ]; then
14     export IBM_JAVA_OPTIONS="-Dfile.encoding=IBM-1047"
15 fi
16
17 BASE_DIR=$(dirname "$0")
18 PARAMS="$@"
19 PWD=`pwd`
20
21 function usage {
22     echo "APIML Certificate Management"
23     echo "usage: apiml_cm.sh --action <action>"
24     echo ""
25     echo "  <action> action to be done:"
26     echo "    - setup - setups APIML certificate management"
27     echo "    - new-service-csr - creates CSR for new service to be signed by exte
28     echo "    - add-service-csr - adds new service signed by local CA or external CA"
```

Pop-up menu Cut/Copy/Delete/In place rename



Set file attributes for code page content tagging

The screenshot displays a web-based editor interface for a file named `apiml_cm.sh`. The interface includes a browser window at the top with the address bar showing "Zowe | apiml_cm.sh - Editor" and a "Not Secure" warning. Below the browser is the editor's workspace, which has a dark theme. On the left, a file explorer shows the directory structure, with `apiml_cm.sh` selected. The main editor area shows the contents of the shell script, including comments and code lines. A red-bordered dialog box titled "apiml_cm.sh - File Tagging" is overlaid on the editor. The dialog has a "Tag file as" label and a dropdown menu currently showing "Untagged". Other options in the dropdown include "Binary", "IBM-1047", "ISO8859-1", and "UTF-8". The system tray at the bottom right shows the time as 18:57 and the date as 02/02/2021.

USS navigator as well as Data Sets

The screenshot displays the Zowe USS Navigator interface. On the left, a file browser shows a directory structure under 'WINCHJ.*'. The 'ZWEKRING' directory is selected. On the right, the 'ZWEKRING - Editor' window shows a JCL script with the following content:

```
71 /**
72 /*******
73 /**      EXPORT SYMLIST=*
74 /**
75 /**      SET  PRODUCT=RACF          * RACF, ACF2, or TSS
76 /**                               12345678
77 /**      SET  ZOWEUSER=ZWESVUSR    * userid for Zowe started task
78 /**                               12345678
79 /**
80 /**      * Hostname of the system where Zowe is to run
81 /**      SET  HOSTNAME=''
82 /**      * IP address of the system where Zowe is to run
83 /**      SET  IPADDRES=''
84 /**      * Keyring for the Zowe userid
85 /**      SET  ZOWERING='ZoweKeyring'
86 /**      * Zowe's certificate label
87 /**      SET  LABEL='localhost'
88 /**      * Name of the data set containing Zowe's certificate (PKCS12)
89 /**      SET  DSNAME=
90 /**      * Password for the PKCS12 data set
91 /**      SET  PKCSPASS=''
92 /**      * Name/Label of the intermediate CA of the Zowe certificate
93 /**      * Ignore if not applicable
94 /**      SET  ITRMZCA=
95 /**      * Name/Label of the root CA of the Zowe certificate
96 /**      * Ignore if not applicable
97 /**      SET  ROOTZCA=
98 /**      * Name/Label of the root CA of the z/OSMF certificate
```

Zowe Desktop includes 3270 emulator

The screenshot displays the Zowe Desktop interface. On the left, a file browser shows a directory structure under 'WINCHJ.*'. The main area is a terminal window titled 'ZWEKRING - Edito' with a file named 'apiml_cm.sh'. The terminal content includes comments and shell commands for setting environment variables like 'HOSTNAME', 'IPADDRESS', and 'PKCSPASS'. Overlaid on the terminal is a '0S/390 Primary Option Menu' with a red border. This menu lists various options such as 'Settings', 'View', 'Edit', 'Utilities', 'Foreground', 'Batch', 'Command', 'Dialog Test', 'LM Facility', 'Programs', 'SCLM', 'Workplace', and 'SDSF'. It also displays system information like 'User ID : WINCHJ', 'Time : 13:38', and 'Terminal : 3278'. At the bottom of the menu, it lists function key shortcuts: 'F1=Help', 'F2=Split', 'F3=Exit', 'F7=Backward', 'F8=Forward', and 'F9=Swap'. The system tray at the bottom right shows the time '18:39' and date '02/02/2021'.

App to app launching – Whole > Sum of parts

The screenshot displays the JES Explorer web interface. On the left is a file explorer showing a directory structure for user WINCHJ, including folders like ISPPROF, SPFLOG1, SPFTMP1, ZWE118, and ZWE1181, and files like ZWECSVSM, ZWEKRING, ZWENOKYR, ZWENOSEC, ZWESASTC, ZWESECUR, ZWESIPRG, ZWESIP00, ZWESISCH, ZWESISTC, and ZWESVSTC. The ZWEKRING file is selected. In the center is a code editor showing the contents of apimL_cm.sh, which includes comments and an EXPORT SYMLIST statement. On the right, a terminal window displays the OS/390 Primary Option Menu with options for Settings, View, and Edit, along with user and time information. Below the terminal is a job details panel for STC05976-JESYSMSG, showing job filters (Owner: *, Prefix: *MSTR, Job ID: *, Status: ACTIVE) and a list of job-related messages. The messages include job expansion information and allocation details for various datasets and SMS.

Menu Utilities Compilers Options Status Help

OS/390 Primary Option Menu

0 Settings Terminal and user parameters User ID : WINCHJ
1 View Display source data or listings Time : 13:38
2 Edit Create or change source data Terminal : 3278

JES Explorer

Owner= * Prefix= *MSTR JobId= * Status= ACTIVE

Job Filters

Owner * Prefix *MSTR
Job ID * Status ACTIVE

APPLY RESET

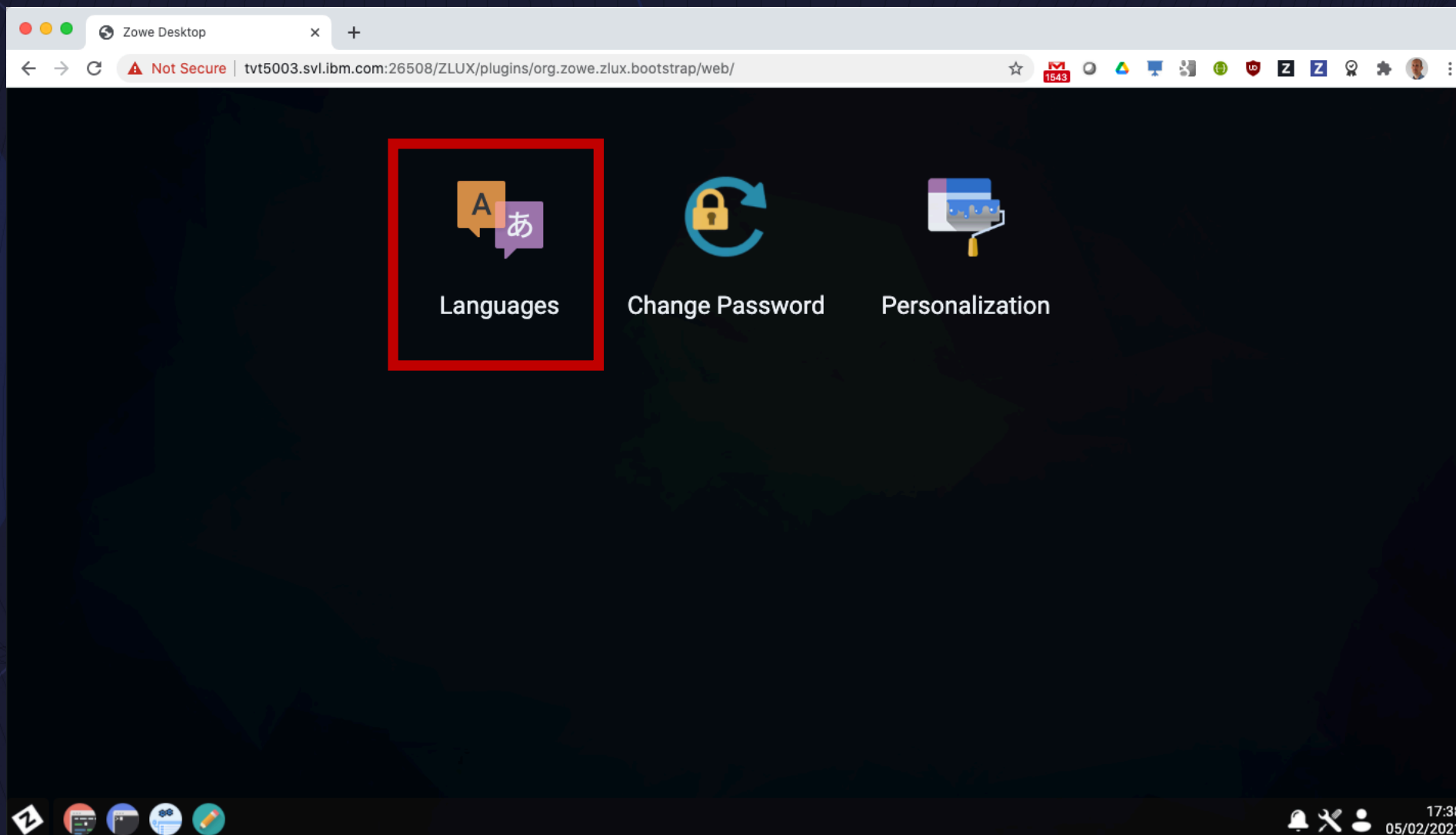
DBC1MSTR:STC05976 [ACTIVE]

JESMSG LG
JESJCL
JESYSMSG

STC05976-JESYSMSG X

```
2 IEF001I PROCEDURE DBC1MSTR WAS EXPANDED USING
3 IEF695I START DBC1MSTR WITH JOBNAME DBC1MSTR IS ASSIGNED
4 IEFA111I DBC1MSTR IS USING THE FOLLOWING JOB RELATED SETT
5 SWA=ABOVE, TIOT SIZE=32K, DSENQSHR=DISALLOW, GDGBIA
6 IEF236I ALLOC. FOR DBC1MSTR DBC1MSTR
7 IGD103I SMS ALLOCATED TO DDNAME STEPLIB
8 IEF237I 0499 ALLOCATED TO
9 IEF237I 0499 ALLOCATED TO
10 IGD103I SMS ALLOCATED TO DDNAME BSDS1
11 IGD103I SMS ALLOCATED TO DDNAME BSDS2
12 IEC161I 056-084, DBC1MSTR, DBC1MSTR, BSDS1,,, DSN121.BSDS01,
13 IEC161I DSN121.BSDS01.DATA, CATALOG.T50032
14 IEC161I 056-084, DBC1MSTR, DBC1MSTR, BSDS1,,, DSN121.BSDS01,
15 IEC161I DSN121.BSDS01.INDEX, CATALOG.T50032
16 IEC161I 062-086, DBC1MSTR, DBC1MSTR, BSDS1,,, DSN121.BSDS01,
17 IEC161I DSN121.BSDS01.DATA, CATALOG.T50032
18 IEC161I 056-084, DBC1MSTR, DBC1MSTR, BSDS2,,, DSN121.BSDS02,
19 IEC161I DSN121.BSDS02.DATA, CATALOG.T50032
20 IEC161I 056-084, DBC1MSTR, DBC1MSTR, BSDS2,,, DSN121.BSDS02,
21 IEC161I DSN121.BSDS02.INDEX, CATALOG.T50032
22 IEC161I 062-086, DBC1MSTR, DBC1MSTR, BSDS2,,, DSN121.BSDS02,
23 IEC161I DSN121.BSDS02.DATA, CATALOG.T50032
24 IGD101I SMS ALLOCATED TO DDNAME (SYS00007)
25 DSN (DSN121.ARCHLOG1.B0000282
26 STORCLAS (CLASS2) MGMTCLAS ( ) DATACLAS (
27 VOL SER NOS= T50036
28 IGD101I SMS ALLOCATED TO DDNAME (SYS00008)
29 DSN (DSN121.ARCHLOG1.A0000282
```

Desktop can have language changed



Set to Chinese

Zowe | Languages

Not Secure | tvt5003.svl.ibm.com:26508/ZLUX/plugins/org.zowe.zlux.bootstrap/web/

1543

Languages

Language Selected: Chinese

Apply

A あ

English

中文

Français

日本語

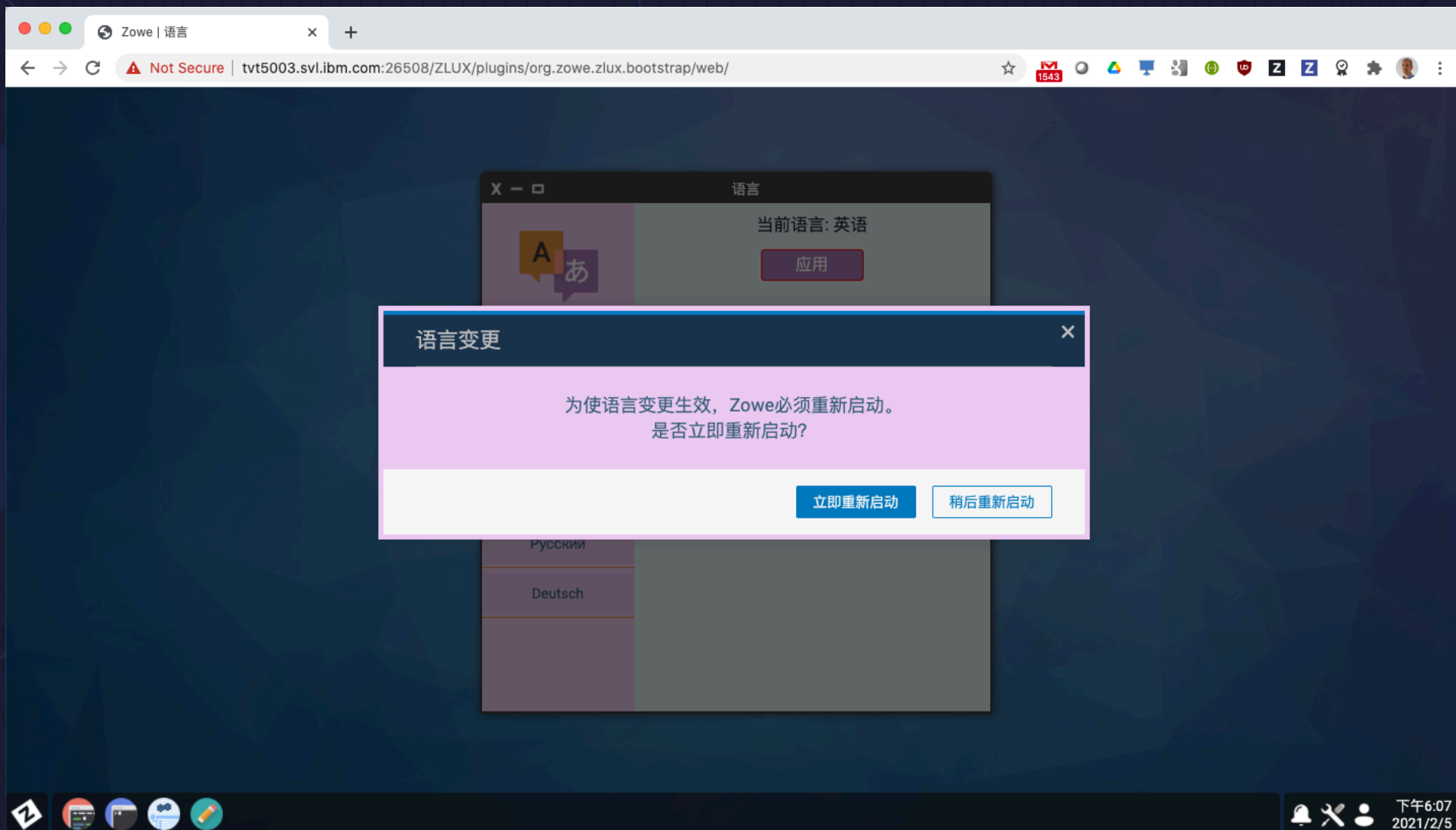
Русский

Deutsch

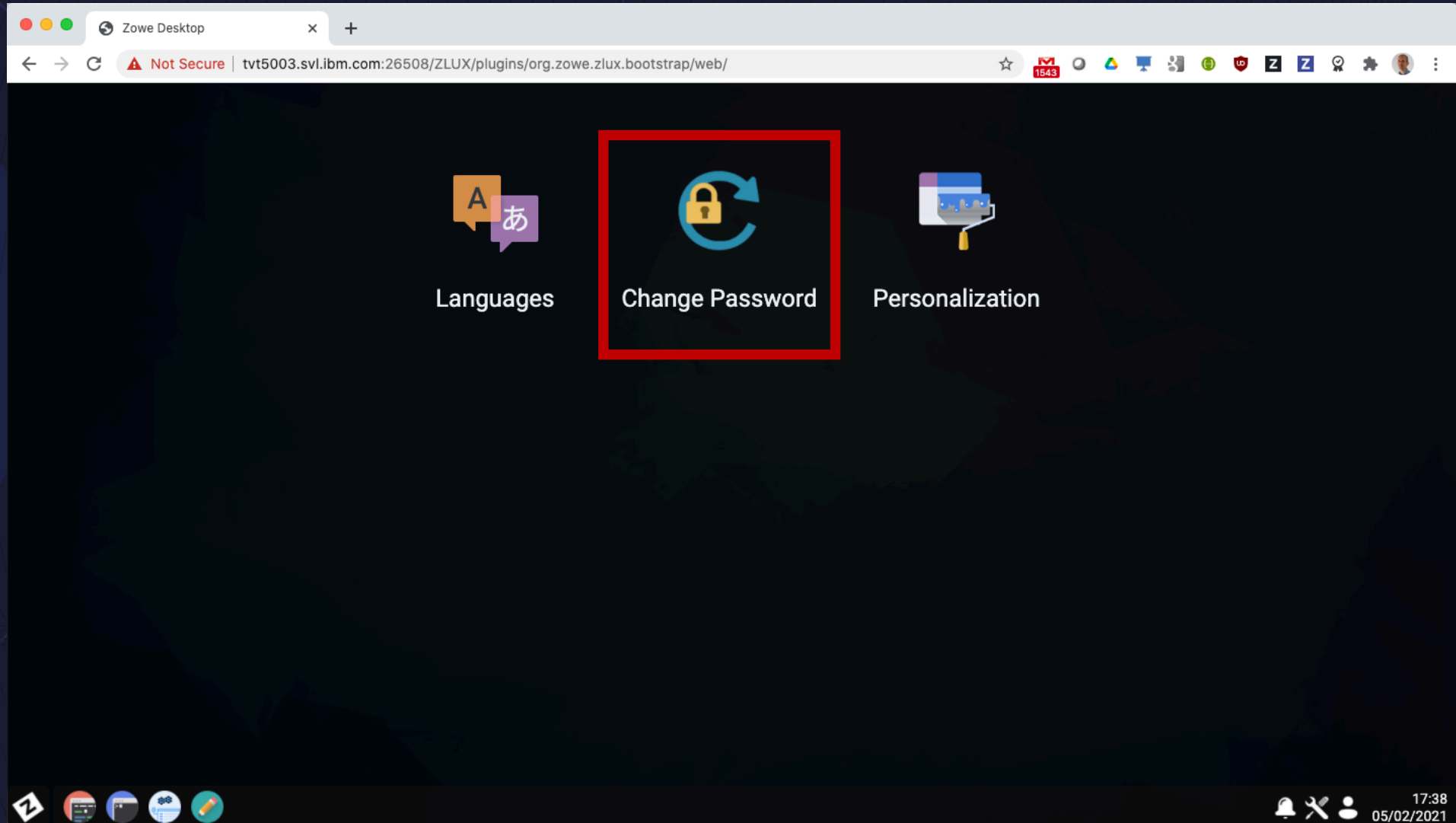
17:39

05/02/2021

Which works across all apps supporting the locale



Change TSO password




The screenshot shows a web browser window titled "Zowe Desktop" with the address bar displaying "tvt5003.svl.ibm.com:26508/ZLUX/plugins/org.zowe.zlux.bootstrap/web/". The page content features three main menu items: "Languages" (with an icon of two speech bubbles, one orange with 'A' and one purple with 'あ'), "Change Password" (with an icon of a padlock and a circular arrow), and "Personalization" (with an icon of a paintbrush). The "Change Password" option is highlighted with a red rectangular border. At the bottom of the browser window, there is a taskbar with several application icons and a system tray showing the time "17:38" and date "05/02/2021".

Password change without needing to log into TSO

Zowe Desktop x +

Not Secure | tvt5003.svl.ibm.com:26508/ZLUX/plugins/org.zowe.zlux.bootstrap/web/

[Back](#)



Password
.....

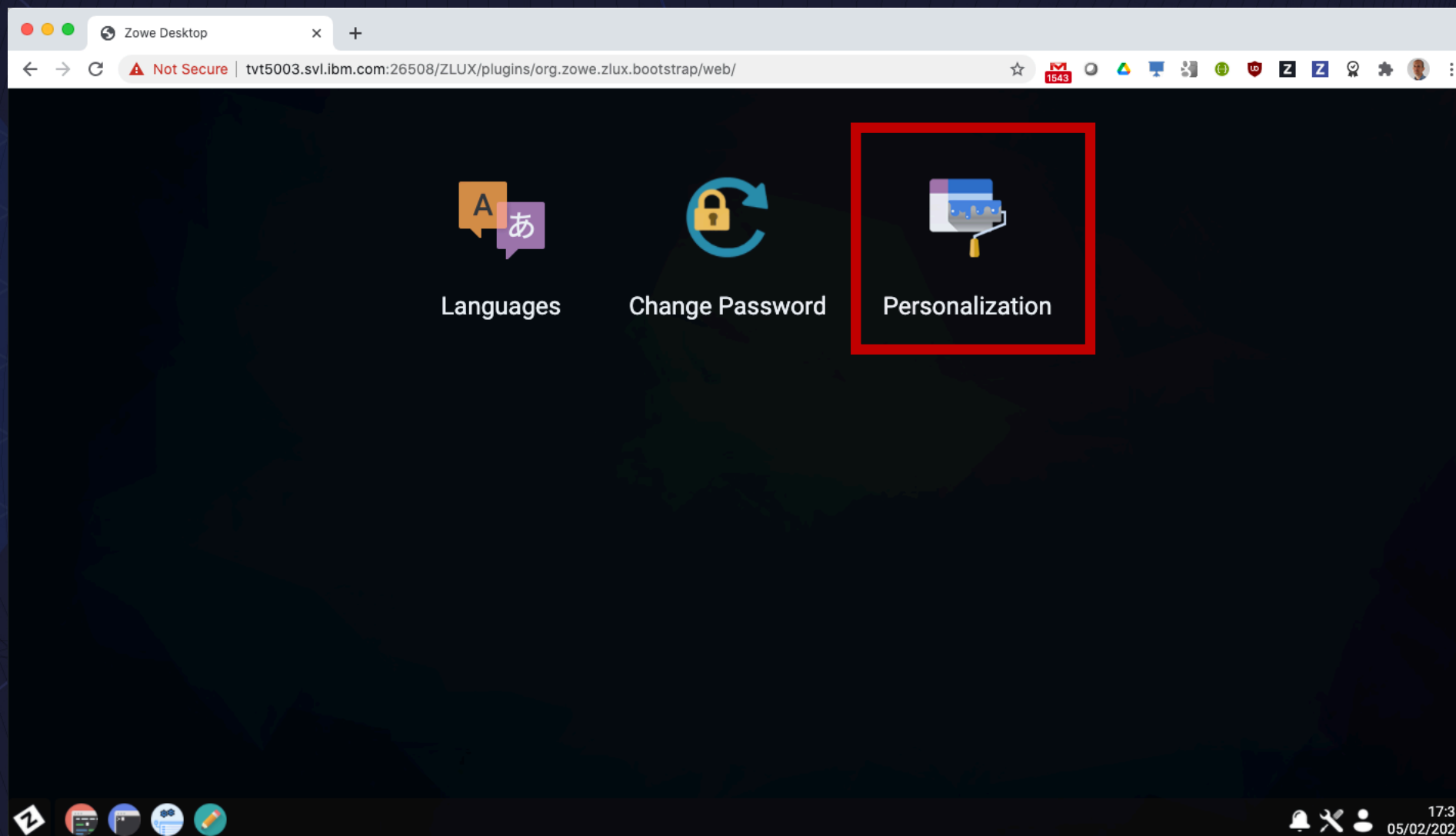
New Password
.....

Confirm New Password
.....

Save and return to desktop

v. 1.18.0+20201208

Desktop theme can be personalized



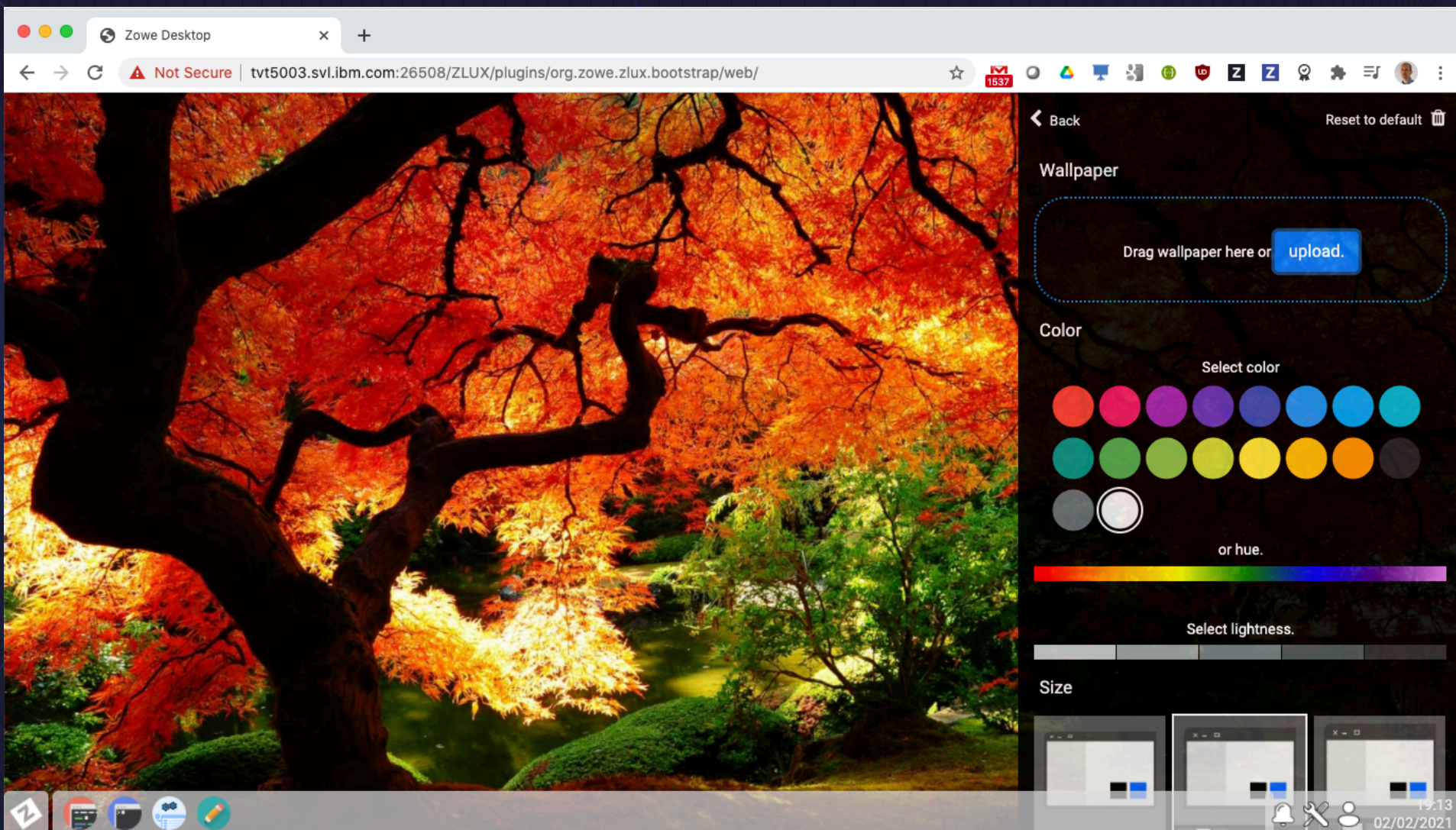
Icon colors and sizes

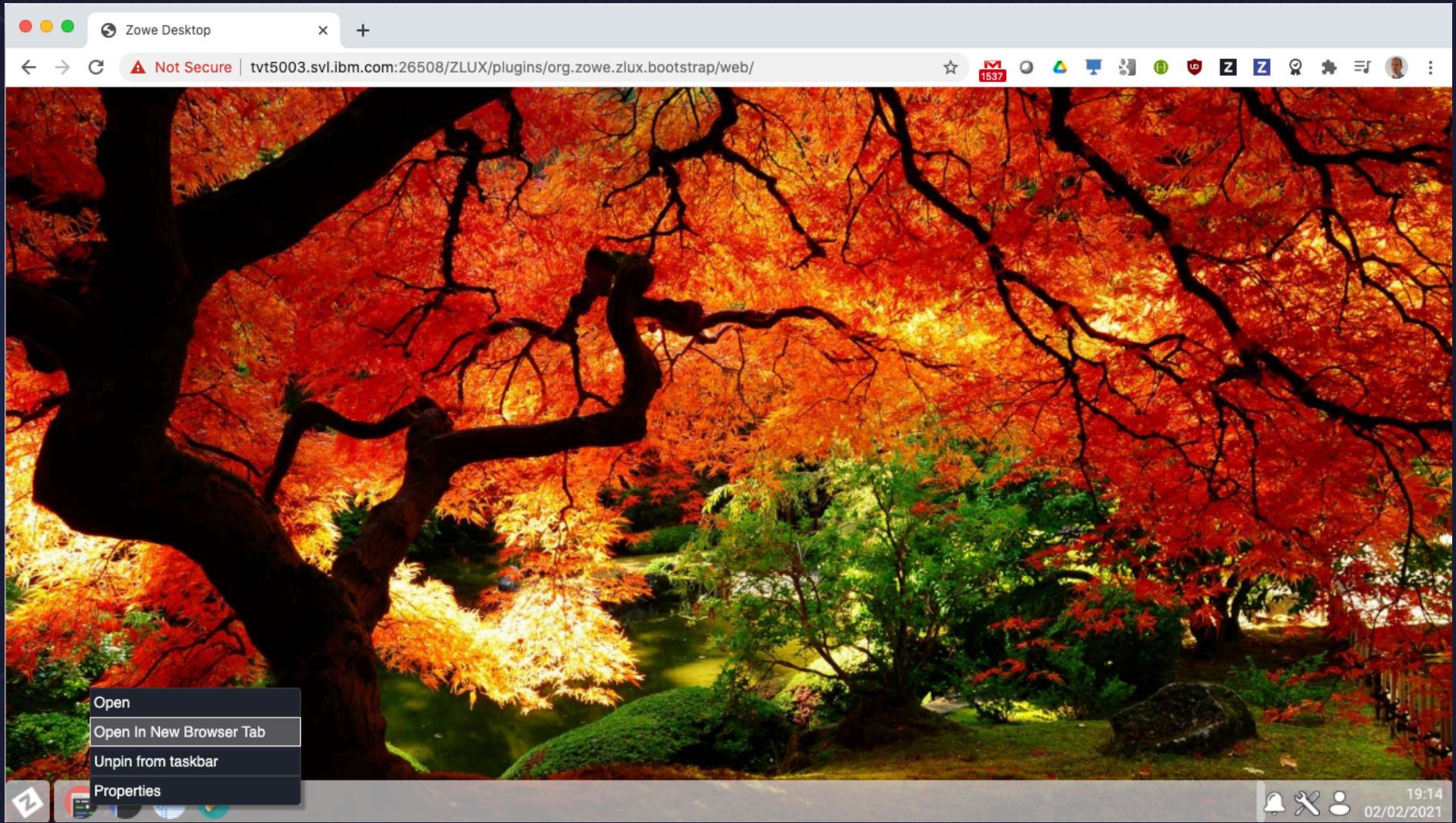
The image shows a web browser window titled "Zowe Desktop" with the address bar displaying "tvt5003.svl.ibm.com:26508/ZLUX/plugins/org.zowe.zlux.bootstrap/web/". The main content area features a blue geometric pattern. A settings panel is open on the right side, containing the following sections:

- Wallpaper:** A dashed box with the text "Drag wallpaper here or [upload.](#)"
- Color:** A "Select color" section with a grid of 18 color swatches (red, pink, purple, blue, cyan, teal, green, yellow, orange, grey) and a "or hue." section with a rainbow color bar.
- Select lightness:** A horizontal slider with a white circle indicating the current position.
- Size:** Three window icons showing different sizes, with a white circle indicating the selected size.

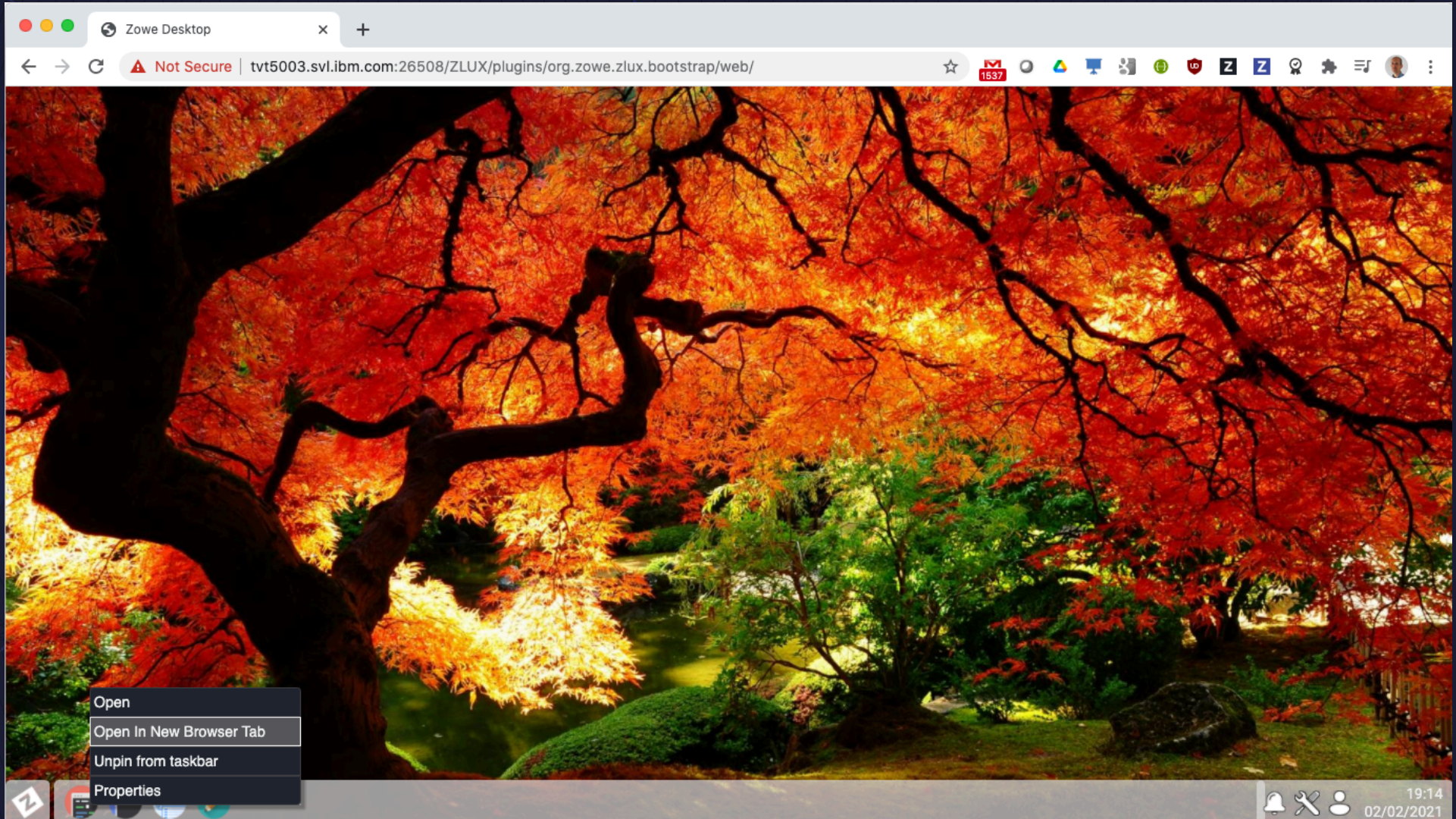
The browser's taskbar at the bottom shows several application icons and a system tray with a clock displaying "15:13" and the date "02/02/2021".

And background can be customized



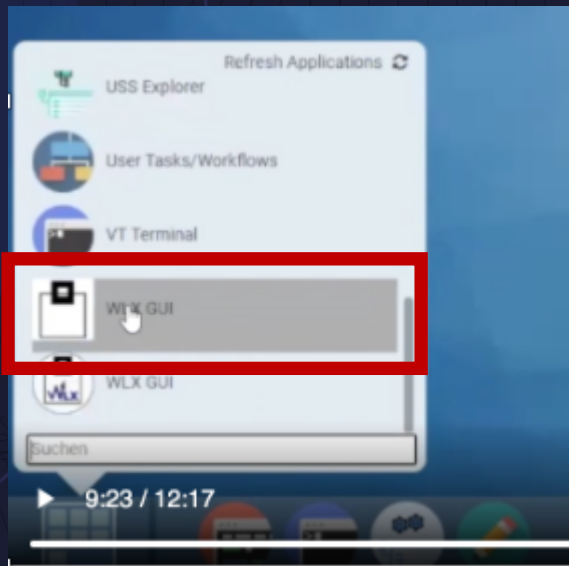
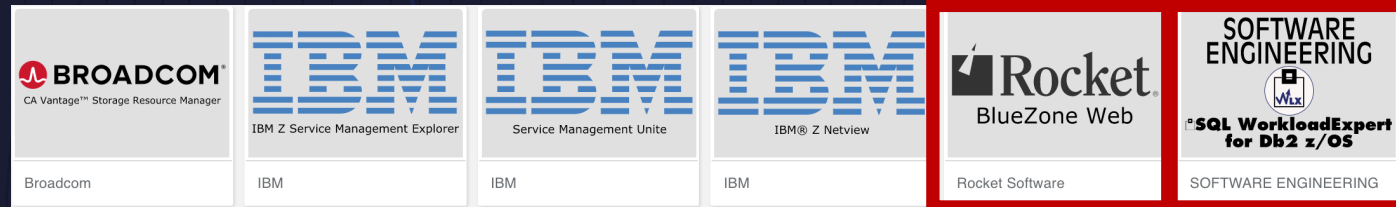


Apps can be launched tile in desktop or new browser tab



Desktop apps have a conformance program

71



Wlx GUI
SQL WorkloadExpert for Db2 z/OS

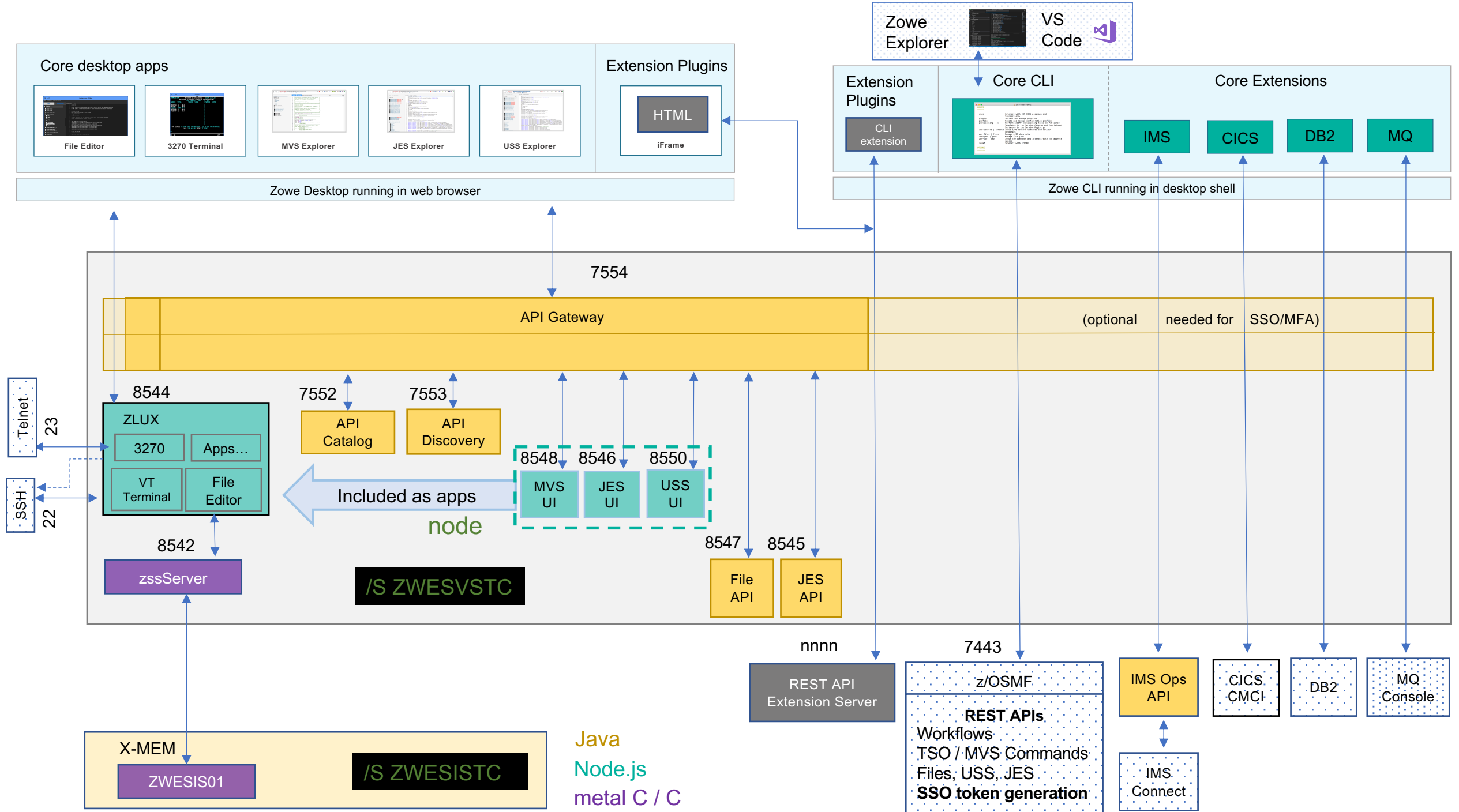
Wlx KPIs and summaries

| Wlx Key | Wlx Type | Highest CPU Time | Sum of Elapsed Time | Average Elapsed Time |
|----------------------------|----------|------------------|---------------------|----------------------|
| 2019-06-17-14.33.47.045752 | X | 19.954375 | 2741 | 0.000264 |
| 2019-07-15-12.45.58.961701 | X | 1.175413 | 193 | 0.000359 |
| 2019-07-26-10.38.21.885695 | X | 1.748068 | 107 | 0.002267 |
| 2019-07-29-14.10.48.190713 | X | 18.371832 | 1994 | 0.000462 |
| 2019-09-06-10.28.26.343640 | X | 18.007326 | 4103 | 0.000223 |
| 2019-10-23-11.39.10.031095 | X | 1.820334 | 172 | 0.000249 |
| 2019-10-23-14.00.41.321452 | X | 1.819609 | 94 | 0.000227 |
| 2019-10-23-16.34.54.926956 | X | 1.851513 | 178 | 0.000257 |
| 2019-10-25-10.21.48.395670 | X | 0.000734 | 0 | 0.000250 |
| 2019-10-25-10.23.04.801566 | X | 0.000021 | 1.864769 | 0.000178 |
| 2019-10-25-12.16.13.813214 | X | 59.458901 | 7176 | 0.000128 |

Create Report dialog:

- Category Column: Wlx Key
- Value Columns: 8 Items selected
- Chart Type: Radar

10:13 / 12:17



API Routing – Reverse Proxy Edge Server – Netflix Zuul

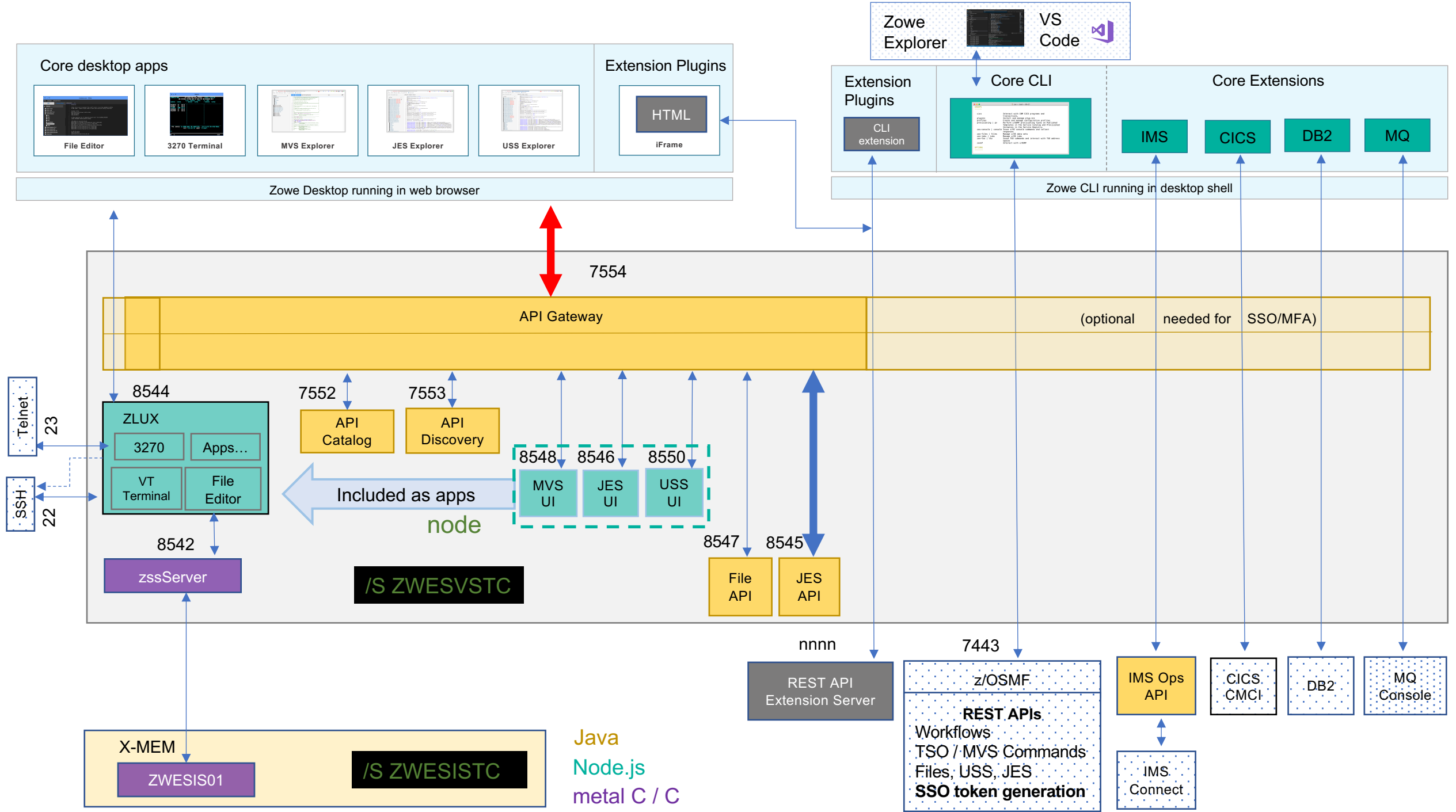
Single sign-on (SSO), Multi Factor Authentication (MFA),
x509 client cert auth

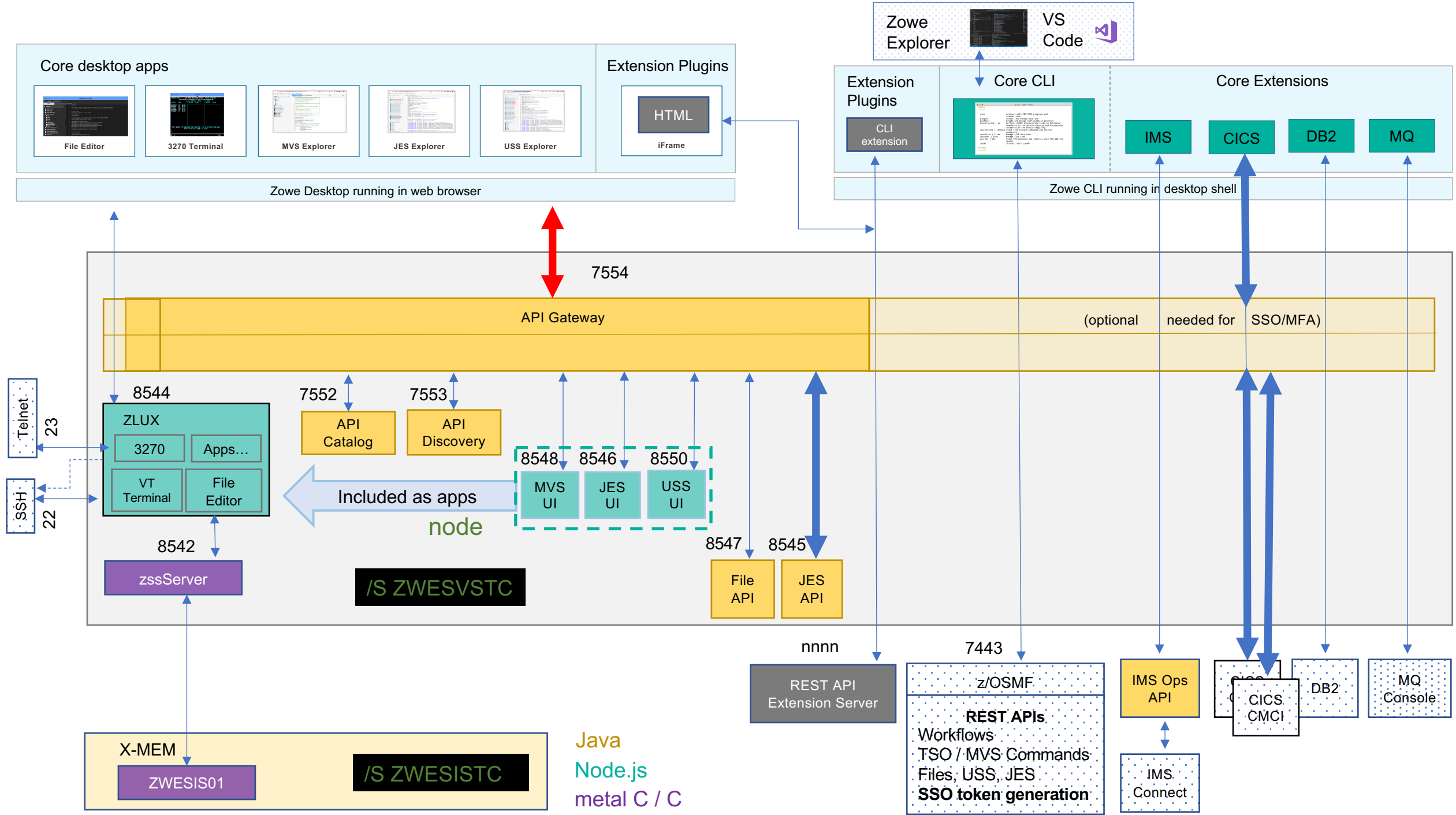
API Catalog – Swagger / Open Api

API Balancing – Workload Management of requests

Single Certificate and external IP

High Availability





API Mediation Layer

API Mediation Layer

- ✓ The API Catalog is running
- ✓ The Discovery Service is running
- ✓ The Authentication service is running

Version 1.18.1 build # 141



Available API services

API Mediation Layer API

The API Mediation Layer for z/OS internal API services. The API Mediation Layer provides a single point of access to mainframe REST APIs and offers enterprise cloud-like feature...

● All services are running

IBM Remote System Explorer (RSE)

The IBM RSE collection of REST APIs allows working with MVS data sets, z/OS UNIX files and commands, JES jobs, TSO commands, and other z/OS components.

● All services are running

z/OS Datasets and Unix Files services

IBM z/OS Datasets and Unix Files REST services

● All services are running

z/OS Jobs services

IBM z/OS Jobs REST services

● All services are running

z/OSMF services

IBM z/OS Management Facility REST services

● All services are running

Zowe System Services (ZSS)

Zowe System Services is used for enabling low-level microservices and other privileged mainframe services, like USS, MVS, authentication, and security management.

● All services are running

API Catalog

Not Secure | tvf5003.svl.ibm.com:26502/apicatalog/ui/v1/#/tile/datasetsAndUnixFiles/unixfiles

Data Sets V2 APIs Data Sets Controller V 2

- GET** `/api/v2/datasets/{filter}/list` Get a list of data sets without attributes matching the filter
- GET** `/api/v2/datasets/{dataSetName}/members` Get a list of members for a partitioned data set
- DELETE** `/api/v2/datasets/{dataSetName}` Delete a data set or member
- GET** `/api/v2/datasets/{filter}` Get a list of data sets matching the filter
- PUT** `/api/v2/datasets/{oldDataSetName}/rename` Rename of a sequential data set, or PDS member
- POST** `/api/v2/datasets` Create a data set
- GET** `/api/v2/datasets/{dataSetName}/content` Get the content of a sequential data set, or PDS member
- PUT** `/api/v2/datasets/{dataSetName}/content` Sets the content of a sequential data set, or PDS member

Unix Files APIs V1 Unix Files Controller V 1

- GET** `/unixfiles/api/v1/{path}/**` Get the contents of a Unix file
- POST** `/unixfiles/api/v1/{path}/**` Create a new Unix File or Directory
- PUT** `/unixfiles/api/v1/{path}/**` Update the contents of a Unix file



Data Sets V2 APIs

Data Sets Controller V 2



GET /api/v2/datasets/{filter}/list Get a list of data sets without attributes matching the filter

GET /api/v2/datasets/{dataSetName}/members Get a list of members for a partitioned data set

DELETE /api/v2/datasets/{dataSetName} Delete a data set or member

GET /api/v2/datasets/{filter} Get a list of data sets matching the filter

This API returns the attributes of data sets matching the filter

Parameters

Try it out

filter * required
string
(path)

Dataset filter string, e.g. HLQ.**, **.SUF, etc.

filter - Dataset filter string, e.g. HLQ.***, **.*

Responses

Response content type application/json

200

OK

Example Value Model

Workload automation in Zowe™ API

Full set of workload automation REST API available for modelling, monitoring and administration

Servers: /wsz [Authorize]

[Model] Calendar

[Model] Event Triggered Tracking

[Model] JCL

[Model] Job Stream

[Model] Operator/Instruction

[Model] Period

[Model] RCGPeriod

[Model] Resource

[Model] RunCycle

[Model] Variable Table

[Model] Workstation

[Model] WorkstationClosedInterval

[Model] ZDestination

[Plan] Critical Job

[Plan] Job

[Plan] Job Stream

[Plan] Resource

[Plan] Virtual Workstation

[Plan] Workstation

Engine

WAPL

[Model] Workstation

| | | | |
|--------|--|---|-------------------------|
| GET | /v1/{engine}/model/workstation | GET operation for Workstation using a TWSKey | getWorkstationByName |
| POST | /v1/{engine}/model/workstation | ADD operation for Workstation | addWorkstation |
| DELETE | /v1/{engine}/model/workstation | DELETE operation for Workstation using a TWSKey | removeWorkstationByName |
| GET | /v1/{engine}/model/workstation/{workstationId} | GET operation for Workstation using a TWSId | getWorkstationById |

[Model] Job Stream

| | | | |
|--------|--|---|----------------------|
| GET | /v1/{engine}/model/jobstream | GET operation for JobStream using a TWSKey | getJobStreamByKey |
| POST | /v1/{engine}/model/jobstream | ADD operation for JobStream | addJobStream |
| DELETE | /v1/{engine}/model/jobstream | DELETE operation for JobStream using a TWSKey | removeJobStreamByKey |
| GET | /v1/{engine}/model/jobstream/{jobstreamId} | GET operation for JobStream using a TWSId | getJobStreamById |

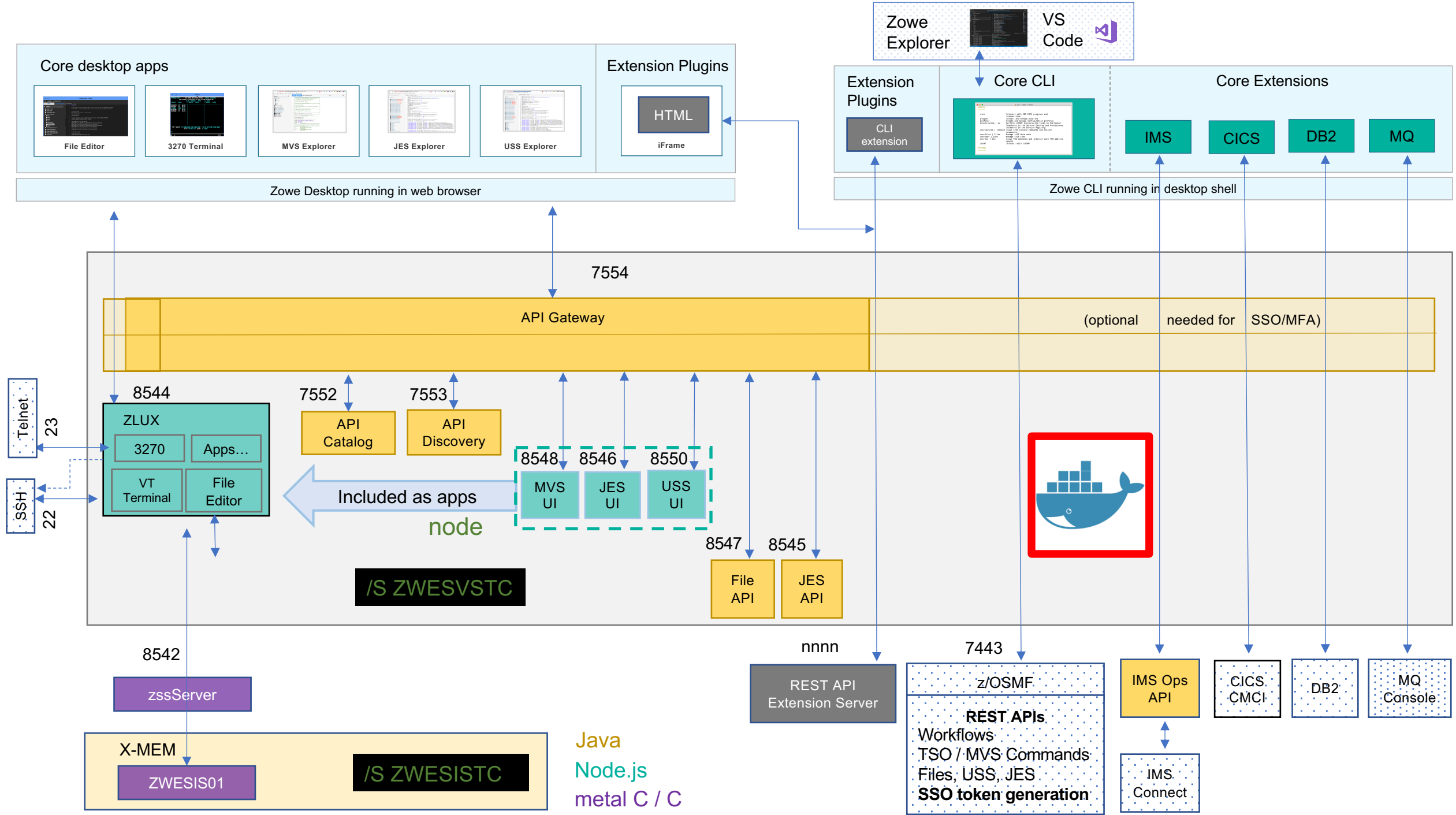
[Plan] Job

| | | | |
|-----|--|-----------------------------------|--------------------------|
| GET | /v1/{engine}/plan/{planId}/job | GET Job In Plan by key | getJobInPlanByKey |
| GET | /v1/{engine}/plan/{planId}/job/{jobId} | GET Job In Plan by Id | getJobInPlan |
| PUT | /v1/{engine}/plan/{planId}/job/{jobId} | Update Job in plan | updateJobInPlanProps |
| PUT | /v1/{engine}/plan/{planId}/job/{jobId}/action/add_dependencies | Add dependencies to a job in plan | addJobInPlanDependencies |
| PUT | /v1/{engine}/plan/{planId}/job/{jobId}/action/add_successors | Adding a successor to the job | addJobInstanceSuccessors |

[Plan] Critical Job

| | | | |
|------|---|--|----------------------------|
| POST | /v1/{engine}/plan/{planId}/criticaljob/{criticalJobId}/query_predecessors_chain | Predecessors for Critical Job in plan | queryPredecessorsChain |
| POST | /v1/{engine}/plan/{planId}/criticaljob/query | Query operation for Critical Job in Plan | queryCriticalJobInPlan |
| POST | /v1/{engine}/plan/{planId}/criticaljob/query_next | Query operation for Critical Job in Plan | queryNextCriticalJobInPlan |
| POST | /v1/{engine}/plan/{planId}/criticaljob/query_next_predecessors_chain | Next predecessors for Critical Job in plan | queryNextPredecessorsChain |
| POST | /v1/{engine}/plan/{planId}/criticaljob/summary_info | Summary info for Critical Job in plan | querySummaryInfo |







docs.zowe.org/stable/



Download Zowe

Zowe has both server and client components, which you can install independently. Download the latest installer to install Zowe on the z/OS server, on your computer, or both. Start your journey with Zowe today!

Want to build Zowe on your own? Access [Zowe GitHub repositories](#) to download the source code.

Server-side component installer

Install Zowe z/OS components from the **convenience build** or the **SMP/E build** depending on your need.

You can also optionally use the Docker build technical preview to run a subset of the Zowe server-side components outside z/OS. Download and learn more about the build in the technical preview section.

Convenience build

PAX archive format installed on the z/OS server

[Zowe 1.18.0 z/OS Convenience build](#)

[Read installation docs](#)

SMP/E build

SMP/E format installed on the z/OS server

Download the base FMID AZWE001 (based on v1.9.0) first and then apply the PTFs to get the latest version.

[Zowe 1.9.0 FMID AZWE001](#)

[Zowe 1.18.0 PTF UO01965 and UO01966](#)

[Read installation docs](#)

Client-side component installer

Install **Zowe CLI** or **Zowe Explorer**, a Visual Studio Code extension powered by Zowe CLI.

Zowe CLI

Install Zowe CLI from the local package or from an npm registry if your computer is connected to the Internet.

Download the Zowe CLI core package and optionally download the plug-ins (CICS, Db2, IMS, MQ, z/OS FTP, and so on) to gain more capabilities.

[Zowe 1.18.0 CLI Core](#)

[Zowe 1.18.0 CLI Plug-ins](#)

[Read installation docs](#)

Zowe Client SDKs

Download the Zowe Software Development Kits (SDKs) for use in development and automation.

[Zowe 1.18.0 Node.js Client SDK](#)

[Zowe 1.18.0 Python Client SDK](#)

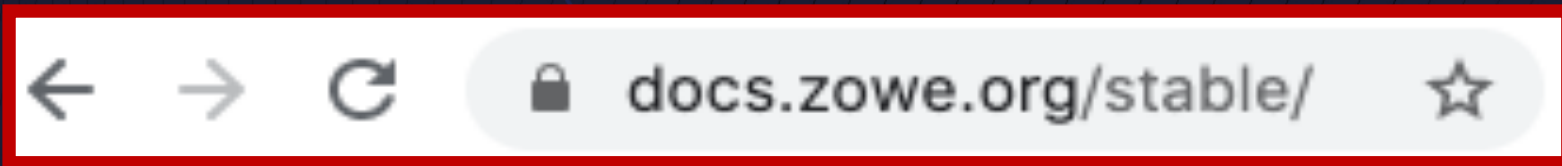
[Read installation docs](#)

Zowe Explorer

Installed directly to VSCode within the GUI

[Visual Studio Code Marketplace](#)

[Read installation docs](#)



User Guide

Planning and preparing the installation ▾

Introduction

Planning the installation of Zowe server components

Planning z/OS installation

Topology of the Zowe z/OS launch process

RUNTIME_DIR

INSTANCE_DIR

KEYSTORE_DIRECTORY

System requirements

Installing Node.js on z/OS

Configuring z/OSMF

Configuring z/OSMF Lite (for non-production use)

UNIX System Services considerations for Zowe

Installing Zowe z/OS components ▶

Installing Zowe Docker Bundle ▶

Installing Zowe CLI ▶

Advanced Zowe configuration ▶

Introduction

The installation of Zowe™ consists of two independent processes: installing the Zowe server components either entirely on z/OS or a combination of z/OS and Docker, and then installing Zowe CLI on a desktop computer.

The Zowe server components provide a web desktop that runs in a web browser providing a number of applications for z/OS users, together with an API mediation layer provides single-sign on (SSO), organization of the multiple zowe servers under a single website, and other capabilities useful for z/OS developers.

Zowe CLI can connect to z/OS servers and allows tasks to be performed through a command line interface.

Because Zowe is a set of components, before installing Zowe, first determine which components you want to install and where you want to install them. This guide provides documentation for all of the components and it is split into sections so you can install as much as you need.

Here are some scenarios to consider:

- If you will only be accessing the Zowe server components through a web browser or REST API client, then you do not need to install the Zowe CLI.
- If you will only be using the Zowe CLI, depending on the plugins used you may not need to install the Zowe server components.
- If you intend to use Docker for the server components, less components need to be installed on z/OS. If you are not using the Desktop or ZSS, then it's possible run the other Zowe components without installing any of Zowe onto z/OS.

Before you start the installation, review the information on system requirements and other considerations.

Planning the installation of Zowe server components

All Zowe server components can be installed on z/OS, but some have the alternative option of being run inside of a Docker image on a Linux host. Which option you choose effects the prerequisites, where they are installed, and the installation steps needed.

Planning z/OS installation

If you are installing one or more server components onto z/OS, the following information is required during the installation process. Software and hardware prerequisites are covered in the next section.



Search or jump to... Pull requests Issues Marketplace Explore

Zowe

Zowe, a top level project of Open Mainframe Project

<https://www.zowe.org> zowe-user@lists.openmainframepr...

Repositories 113 Packages People 132 Teams 36 Projects

Pinned repositories

- community**
Zowe Community - Sub-projects, Squads, Contribution Guidelines, Meeting Minutes, and more
☆ 27 👤 33
- zlc**
Zowe Leadership Committee collaboration
● Batchfile ☆ 13 👤 14
- docs-site**
Documentation for the Zowe project
● Groovy ☆ 45 👤 77

Find a repository... Type: All Language: All

explorer-mvs

● JavaScript 📄 EPL-2.0 👤 9 ☆ 0 ⚠️ 1 🔄 7 Updated 10 minutes ago

zowe-install-packaging

● Shell 📄 EPL-2.0 👤 34 ☆ 18 ⚠️ 246 (1 issue needs help) 🔄 5 Updated 18 minutes ago

explorer-jes

Top languages

- TypeScript ● JavaScript ● Java ● Shell ● Groovy

Most used topics

Manage

- zowe zowe-cli mainframe java rest-api

<https://slack.openmainframeproject.org/>

The screenshot displays the Slack interface for the Open Mainframe Project workspace. On the left sidebar, a list of channels is visible, including #zowe-dev, #zowe-doc, #zowe-explorer, and #zowe-user. The main area shows a message in the #zowe-dev channel from Tom Zhang, dated 4:22 AM, announcing the promotion of Build #145 to Zowe v1.20.0-RC2. The message includes several download links for local releases, SMP/e builds, and Docker images for amd64 and s390x architectures. It also provides links for the CLI Core Package and CLI Plugins Package. The interface includes a search bar at the top, a channel list on the left, and a message input area at the bottom.

#zowe-dev ☆
1 | General Development and architecture discussions

data from multiple lpar of multiple sysplexes with unrelated z/OSM within the same UI. There's still much to do and anyone interested in being on Tuesday, March 9th at the architecture and work involved should reply on this message so we... [Show more](#)

Labels
enhancement

zowe/community | Mar 9th | Added by GitHub

3

Thursday, March 11th

James Schlosser 10:24 PM
Does anyone have system symbols setup for the zowe root directory? If so how did you get Zowe to find the MVS, JES, and OMVS explorers?

Friday, March 12th

Tom Zhang 4:22 AM
Build #145 is promoted as Zowe v1.20.0-RC2, you can download from:

<https://zowe.jfrog.io/zowe/libs-release-local/org/zowe/1.20.0-RC2/zowe-1.20.0-RC2.pax>

SMP/e build:
<https://zowe.jfrog.io/zowe/libs-release-local/org/zowe/1.20.0-RC2/zowe-smpe-package-1.20.0-RC2.zip>

docker amd64 image:
<https://zowe.jfrog.io/zowe/libs-release-local/org/zowe/1.20.0-RC2/server-bundle.amd64-1.20.0-RC2.tar>

docker s390x image:
<https://zowe.jfrog.io/zowe/libs-release-local/org/zowe/1.20.0-RC2/server-bundle.s390x-1.20.0-RC2.tar>

The CLI Core Package is published here:
<https://zowe.jfrog.io/zowe/libs-release-local/org/zowe/1.20.0-RC2/zowe-cli-package-1.20.0-RC2.zip>

The CLI Plugins Package is published here:
<https://zowe.jfrog.io/zowe/libs-release-local/org/zowe/1.20.0-RC2/zowe-cli-plugins-1.20.0-RC2.zip>

Message #zowe-dev

B I Aa @ 😊 🗑️ ➡️



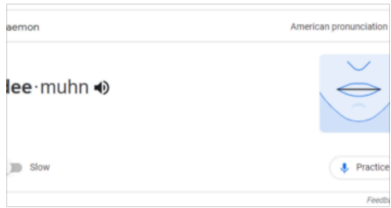
medium.com/zowe



Top 5 Most Read Zowe Stories

{Core} 2020 was a year of growth for the Zowe open source framework, highlighted by the 41 articles contributed by the community.

David McNierney
Jan 5 · 2 min read



Zowe CLI—A Faster Experience

{Core} Here we'll show a feature planned for the next major version of Zowe CLI—the ability to run it in "daemon" mode.

Dan Kelosky
Dec 29, 2020 · 3 min read



CA Vantage and Zowe

{Ecosystem} Reducing the strain on storage administrators with CA Vantage and Zowe

James Branam
Dec 17, 2020 · 4 min read



Deep interaction with zOS in Zowe

{Core} Integrate new functions to ZSS to enable new zOS functionality in Zowe UI.

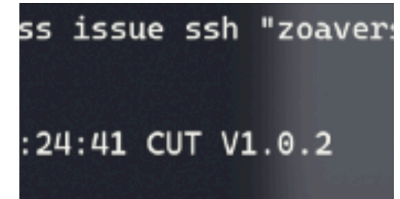
Jakub Balhar
Nov 25, 2020 · 6 min read



Access z/OS Performance and Management Data using CA SYSVIEW and Zowe CLI

{Ecosystem} The Zowe CLI and CA System plugin allows access to z/OS Performance Management Daten from JES, CICS, MQ, DB2, IMS and more ...

Michael Heuzey
Nov 24, 2020 · 5 min read



How to Call Z Open Automation Utilities (ZOA Utilities) from Zowe CLI

{Core} Zowe CLI working alongside ZOA Utilities. Background, samples, opening up the Zowe CLI to access and manipulate your z/OS system

Dan Kelosky
Oct 20, 2020 · 3 min read



CA File Master Plus Zowe CLI Plugin: Enhanced to support Single Sign-on

{Ecosystem} Using CA File Master Plus with the Zowe CLI and API alongside multi-factor authentication and single sign-on.

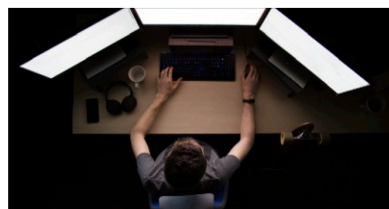
Sudeep Chaurasia
Dec 14, 2020 · 4 min read



The ZAAS Client: a library for the API Mediation Layer

{Core} The ZAAS Java client library allows interaction with Zowe's SSO, JWT, MFA and PassTicket mechanisms provided by Zowe. Here's how !

Andrea Tabone
Dec 3, 2020 · 5 min read



Zowe SDKs: Build z/OS Connected Applications Faster

{Core} For some time, the Open Mainframe Project's Zowe has been opening up the mainframe ecosystem, allowing end users to leverage the...

Andrew Harn
Nov 30, 2020 · 4 min read



Keep certificates stored in the SAF keyrings

{Core} Zowe now supports the possibility to keep the certificates for communication safely stored within a SAF keyring. Here is how ...

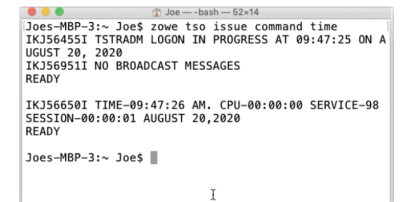
Jakub Balhar
Sep 18, 2020 · 11 min read



Streamline your mainframe development with Zowe Explorer & Zowe CLI (a series...)

{Core} Zowe Explorer and Zowe CLI—streamlining mainframe development with REXX.

Alex Dumitru
Aug 25, 2020 · 4 min read



Zowe CLI and TSO commands

{Core} Zowe CLI allows you to issue TSO command, as well as start, work with, and stop a session all from your desktop PC

Joe Winchester
Aug 18, 2020 · 9 min read

Configuring Zowe Db2 Command Line Interface



Joe Winchester
Jan 19 · 8 min read



{Core} The [Open Mainframe Project's Zowe](#) command line interface (CLI) provides the ability for a developer to manipulate z/OS by issuing commands on their laptop. It does this by connecting to z/OS and issuing z/OSMF REST APIs. As well as providing base function, the CLI is also extensible allowing it to connect to other endpoints on z/OS including our favourite enterprise database Db2. This blog describes how to obtain, configure, troubleshoot, and use the Zowe Db2 CLI plugin. I'm fortunate enough to use a Mac as my go-to laptop so the article includes idiosyncrasies peculiar to MacOS, however the Zowe Db2 CLI works equally well on Windows, Linux, or any operating system that can run Node.js.

Pre-requisites

The Command Line Interface has some pre-requisites that need to be installed beforehand, see [Software requirements for Zowe CLI plug-ins](#).

SQL0805N: Database BIND

To be able to run remote client SQL commands against a database it needs to have had a `BIND` command invoked against it. If the database hasn't been bound then you'll see the error `SQL0805N` .

```
Command Error:
DB2 ODBC Driver Error: [node-ibm_db] Error in
ODBCConnection::QuerySync while executing query.Error Details:
Error: [IBM][CLI Driver][DB2] SQL0805N Package
"DSNV112E.NULLID.SYSSH200.5359534C564C3031" was not found.
SQLSTATE=51002
SQLCODE: -805
SQLSTATE: 51002
```

The TSO user ID that does this needs to have `BINDADD` authority (*strictly speaking this depends on the value of the subsystem parameter `BINDND` whose default value is `BINDADD` which could be different between sites*). More information on the `BIND` command can be found in the [DB2 knowledge centre documentation](#), as well as [The Bind process](#). It's likely that you'll need to ask your friendly z/OS Db2 sysprog to do the `BIND` for you, and as they might be getting tired of you pinging them with requests be sure to promise you'll buy them a cup of tea and pastry next time you're together in their favorite neighborhood cafe.

Why the mainframe is alive and thriving

Mainframes are still going strong after 70 years.

Recommended Content:

White Papers: Get IBM Cloud Free Tier

Experiment and build with 40+ services on IBM Cloud for free, including Watson APIs, DevOps tools and more. No credit card required. No time limits.

Get Started



By R "Ray" Wang for Constellation Research | March 5, 2021 -- 20:33 GMT (20:33 GMT) | Topic: Cloud

Mainframes entered the market in the early 1950's when IBM and the seven dwarfs (Burroughs, Unisys, NCR, Control Data, Honeywell, GE, and RCA) created the computing age and competed for critical applications, sophisticated modeling, and large-scale transactions and workloads among the largest of organizations. Over the past seven decades, compute power, storage, and networking have seen various waves of centralization and decentralization amidst each wave of disruptive technology adoption.

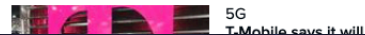
RECOMMENDED FOR YOU

Enjoy full access with a Pay-As-You-Go account

White Papers provided by IBM

DOWNLOAD NOW

MORE FROM STEPHANIE CONDON



5G

Mobile says it will

Moreover, investments by organizations such as Broadcom and IBM's in **Zowe**, the open mainframe project, enable all users to access an integrated and extensible open source framework for z/OS. Users can use one language to access open systems and open software through a set of common APIs and OS capabilities. Consequently, developers can use the same tools for the cloud and mainframe enabling a broader set of developers and resources to work on both. The numbers show this as well with 94 of the top 100 IBM Z Enterprises are running Linux on Z and able to modernize with the mainframe. In addition, a 2019 survey of mainframe professionals conducted by Forrester for IBM showed that nearly 90% of professionals in enterprise computing careers agree the job market for their skill set is growing, and 75% believe there is a high demand for their skills in mainframe.



WHERE THE WORLD MEETS DEVOPS

LATEST • DEVOPS CHATS WEBINARS • LIBRARY EVENTS • SPONSORED COMMUNITIES • RELATED SITES • MEDIA KIT

AI CLOUD CONTINUOUS DELIVERY CONTINUOUS TESTING DEVSECOPS LEADERSHIP SUITE PRACTICES ROELBOB LOW-C

Home » Blogs » DevOps Practice » Open Mainframe Project: Zowe Ready for Prime Time



Open Mainframe Project: Zowe Ready for Prime Time



BY: DON MACVITTIE ON FEBRUARY 14, 2019 — 0 COMMENTS

There is a lot of interest in updating mainframe technology/interfaces across traditional enterprises. As development environments and toolsets have evolved outside the mainframe, there is a struggle to keep up—partially because backward compatibility requirements make wild changes difficult and partly because the very architecture of mainframes is different.

QUESTIONS?



THE END

Tobias Leicher

**zClient IT Architect
& zChampion for Modernization**



IBM Allee 1
D-71139 Ehningen
☎ 0151 – 15 16 24 89
✉ tobias.leicher@de.ibm.com

