ZCLI1

Zowe CLI e VSAM

Gravando registros com Zowe CLI e VSAM

15 passos

120 minutos

O DESAFIO

Você interage com o mainframe por meio de uma série de transações. Você emite um pedido para visualizar os trabalhos, outro para visualizar conjuntos de dados, outro para emitir um comando. Nos bastidores, a estrutura de código aberto, Zowe, está trabalhando para vincular o recursos do mainframe com APIs, comandos e bibliotecas fáceis de usar. Simplificando, você pode acessar um sistema Z de praticamente qualquer lugar, usando uma ampla variedade de ferramentas e plataformas.

ANTES DE VOCÊ COMEÇAR

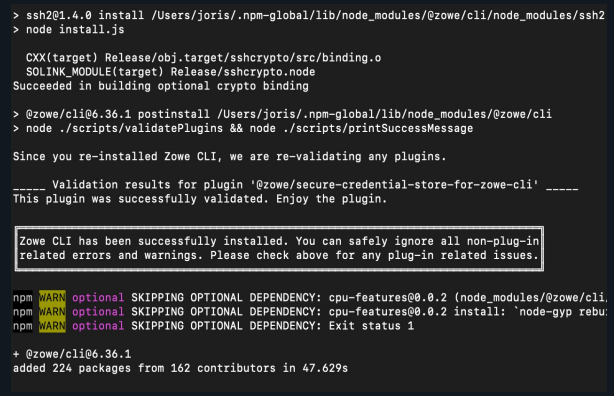
Este desafio fará mais sentido se você já completou todos os desafios de Fundamentos, como ele usa um pouco de tudo a partir daí. Nada é necessário, mas faremos suposições sobre o que você sabe neste momento.

1. Instalando o Zowe CLI

Estamos usando o plug-in Zowe Explorer para VS Code ao longo deste concurso, mas o Zowe faz muito mais e é responsável por trazer muito mais para o mainframe.

Para ser claro,**você está instalando o Zowe CLI em seu próprio computador, não no mainframe**. Você usará o Zowe CLI para fazer interface com o Zowe ez/OSMF que está sendo executado no mainframe, mas você estará conduzindo a maior parte desse desafio em seu próprio computador.

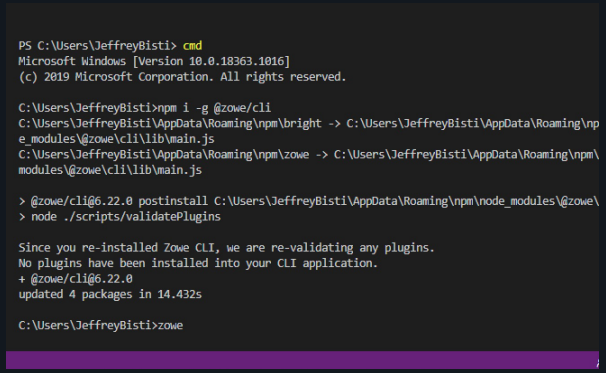
Os usuários do Linux podem precisar explorar um pouco para encontrar o que funciona em seu sistema específico, mas deve se aproximar das etapas do Mac, apenas substituindo o arquivo de perfil do shell correto.



2. Instalação do Zowe CLI para MAC

Para usar pacotes de nós no sistema operacional, precisamos para carregá-los em um diretório .npm-global que pode ser acessado por usuários regulares. Essas etapas configurarão isso, informarão ao npm (Node Package Manager) para usá-lo e incluirão isso no normal lista de locais onde procura programas para serem executados. Para usuários do MacOS, isso deve funcionar.

1:**mkdir ~/.npm-global**
2**: npm config set prefix '~/.npm-global'**
3:**echo "export PATH=~/.npm-global/bin/:$PATH" >> .zprofile**
4:**fonte .zprofile**
5:**npm i -g @zowe/cli**



3. Configuração do NPM para Windows

No Windows, primeiro vamos mudar para cmd do PowerShell e, em seguida, instalar o zowe zli usando npm, o Node Package Gerente. Isso deve funcionar para a maioria dos usuários, embora sua saída possa parecer um pouco diferente do que você vê na captura de tela.

1: tipocmd (isso mudará o shell para cmd do PowerShell)
1:**npm i -g @zowe/cli**
2:**zowe**

Ainda preso? Entre nos fóruns para obter orientação

```
DESCRIPTION
-----------

  Welcome to Zowe CLI!

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

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

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

USAGE
-----

  zowe <group>

  Where <group> is one of the following:
```

4. Um comando de quatro letras

Agora que estamos todos configurados, pegue um novo terminal e digite o comando**zowe**. Assim mesmo, por si só.

Certifique-se de seguir as instruções de configuração de perfil das etapas 5 a 6 do desafio REXX1, caso contrário, isso poderá falhar.

Você receberá uma descrição, uma lista de grupos de comandos e opções. Vamos gastar muito desse desafio passando por esses grupos de comando, e muitos deles devem soar um pouco familiares. Ir para zowe.org aprender mais.

```
USAGE
-----

  zowe zos-console <group>

  Where <group> is one of the following:

GROUPS
------

  collect Collect z/OS console command responses
  issue   Issue z/OS Console Commands

GLOBAL OPTIONS
--------------

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

     Produce JSON formatted data from a command

  --help  | -h (boolean)

     Display help text

  --help-examples  (boolean)
```

5. Construindo peça por peça

Você está usando a funcionalidade do Zowe para emitir comandos e fazer todo tipo de coisas através do VS Code. Neste desafio, estamos apenas usando o componente CLI autônomo para fazer as coisas de uma maneira diferente, o que pode ser útil em alguns situações.

Por exemplo, para ver o que mais pode ser feito no*console* grupo, digite o comando**console zowe**e depois aperte enter. Você pode ver que há uma opção para emitir comandos, bem como coletar respostas. Esses são mais dois grupos de comando dentro console.

```
EXAMPLES
--------

  - Submit the JCL in the data set "ibmuser.cntl(deploy)":

     $ zowe zos-jobs submit data-set "ibmuser.cntl(deploy)"

  - Submit the JCL in the data set "ibmuser.cntl(deploy)", wait
  for the job to complete and print all output from the job:

     $ zowe zos-jobs submit data-set "ibmuser.cntl(deploy)" --vasc

  - Submit the JCL in the file "iefbr14.txt":

     $ zowe zos-jobs submit local-file "iefbr14.txt"

  - Download all the output of the job with job ID JOB00234 to
  an automatically generated directory.:

     $ zowe zos-jobs download output JOB00234

  - View status and other details of the job with the job ID
  JOB00123:
```

6. Os exemplos são bons doc

Use o comando**zowe zos-jobs --help-examples**por um bom lista de comandos do zowe que você pode usar relacionados a tarefas do z/OS. A saída vai além do que é capturado na captura de tela acima, e há muitas variações disponibilizadas.

Estamos começando com o básico, não se preocupe, isso ficará um pouco mais empolgante em apenas mais algumas etapas.

"ME FALA MAIS SOBRE ZOWE. ISSO É COISA DA IBM OU...?"

Zowe é um projeto de código aberto para z/OS, destinado a tornar a plataforma mais acessível para usuários que não estão começando com anos e anos de experiência em mainframe. O projeto Zowe contém contribuições de indivíduos e empresas na comunidade de mainframe. Isso inclui o plug-in VS Code, várias APIs e o Zowe CLI que você está prestes a explorar.

Zowe é um projeto deOpen Projeto Mainframe , que é um projeto gerenciado pelaLiFundação Nux . Não é um produto IBM, embora a IBM seja um contribuidor e apoiador, e continue a defender Zowe como um modelo estratégico para trazer novos recursos e usuários para a plataforma de mainframe.

Uma das melhores maneiras de se conectar com empregadores e pessoas informadas é prestar atenção ao que está acontecendo nessas comunidades e ajudar sempre que encontrar uma oportunidade.

```
Monorail:~ jbisti$ zowe zos-tso issue command "status" --rfj
{
  "success": true,
  "exitCode": 0,
  "message": "",
  "stdout": "IKJ56455I Z99999 LOGON IN PROGRESS AT 08:41:30 ON JULY 14, 2020\nIKJ56951I NO BROADC
Y \n\nIKJ56216I NO JOBS FOUND+\nREADY \n\n",
  "stderr": "",
  "data": {
    "success": true,
    "startResponse": {
      "success": true,
      "zosmfTsoResponse": {
        "servletKey": "Z99999-78-aabeaaaq",
        "queueID": "1507336",
        "sessionID": "0x4E",
        "ver": "0100",
        "tsoData": [
          {
            "TSO MESSAGE": {
              "VERSION": "0100",
              "DATA": "IKJ56455I Z99999 LOGON IN PROGRESS AT 08:41:30 ON JULY 14, 2020"
            }
          },
          {
            "TSO MESSAGE": {
              "VERSION": "0100",
              "DATA": "IKJ56951I NO BROADCAST MESSAGES"
            }
```

7. Formato para JSON

Certifique-se de ter seus trabalhos ativos e digite o comando **zowe zos-jobs lista de empregos.**

Você recebe de volta uma lista de tarefas do z/OS em execução ativa às quais tem acesso para consultar. Arrumado!

Agora, emita o mesmo comando com**--rfj**(Formato de resposta JSON) depois dele. Agora você obtém a saída COMPLETA, e a saída é no formato JSON, que pode ser muito mais facilmente interpretado por programas que lidam com o formato JSON.

E o JSON

Por que precisamos de JSON quando a saída original fez todo o sentido para nós?

JSON significa JavaScript Object Notation, e é apenas uma maneira de aninhar os atributos de algo em um objeto para que possa ser totalmente representado sempre que for acessado. Costuma ser um pouco mais leve e flexível do que outro formato de arquivo com um objetivo semelhante que você já deve ter ouvido falar, chamado XML.

Em muitas linguagens de programação, você pode simplesmente carregar um objeto JSON e usar*notação de ponto*para acessar os vários atributos desse objeto JSON, economizando tempo valioso na programação, em comparação com a tarefa manual de escrever analisadores para extrair informações da saída regular.

```
MTM> zowe zos-files list ds Z99999.ZOWEPS -a

dsname: Z99999.ZOWEPS
blksz:  6160
catnm:  MASTERV.CATALOG
cdate:  2020/08/28
dev:    3390
dsorg:  PS
edate:  ***None***
extx:   1
lrecl:  80
migr:   NO
mvol:   N
ovf:    NO
rdate:  ***None***
recfm:  FB
sizex:  15
spacu:  CYLINDERS
used:   0
vol:    VPWRKA
vols:   VPWRKA
```

8. Atribuir e listar

Vamos colocar isso em uso. Dê uma olhada no*arquivos*grupo de comandos e use-o para alocar (criar) um conjunto de dados sequencial chamado **Zxxxxx.ZOWEPS***(com seu próprio ID de usuário, é claro)*

Em seguida, use outro comando zowe cli para mostrar os atributos do conjunto de dados que você acabou de criar. Eles devem ser semelhantes ao acima.

Se você receber uma mensagem de tempo limite, tente adicionar **- - tempo limite de resposta 30**até o final do comando, para permitir atrasos na resposta.

```
dsname: Z99999.ZOWEPS
blksz:  9600
catnm:  CATALOG.ZOS1
cdate:  2020/07/14
dev:    3390
dsorg:  PS
edate:  ***None***
extx:   1
lrecl:  120
migr:   NO
mvol:   N
ovf:    NO
rdate:  ***None***
recfm:  FB
sizex:  15
spacu:  CYLINDERS
used:   0
vol:    VPWRKB
vols:   VPWRKB
```

9. Totalmente personalizado

Então agora você conhece outra maneira de criar e ver conjuntos de dados. O problema é que criamos esse conjunto de dados usando um conjunto padrão de valores, e uma das grandes vantagens dos conjuntos de dados é como eles são personalizáveis. Excluir esse conjunto de dados (com outro**arquivos**comando) e use a ajuda (ou oonline documentação em zowe.org) para**recriar esse conjunto de dados sequenciais**com alguns atributos personalizados.

Primeiro, queremos que o comprimento do registro seja 120 em vez do padrão 80 (temos alguns registros longos) e queremos um tamanho de bloco de 9600.

Ao obtê-lo, você verá uma leitura diferente para o Bloco Tamanho e Comprimento do Registro (LRECL), como na captura de tela acima.

```
EXAMPLES
--------

  - Create a VSAM data set named "SOME.DATA.SET.NAME" using
  default values of INDEXED, 840 KB primary storage and 84 KB secor

     $ zowe zos-files create data-set-vsam SOME.DATA.SET.NAME

  - Create a 5 MB LINEAR VSAM data set named
  "SOME.DATA.SET.NAME" with 1 MB of secondary space. Show the prope
  data set when it is created:

     $ zowe zos-files create data-set-vsam SOME.DATA.SET.NAME --dat
secondary-space 1MB --show-attributes

  - Create a VSAM data set named "SOME.DATA.SET.NAME", which is
  retained for 100 days:

     $ zowe zos-files create data-set-vsam SOME.DATA.SET.NAME --ret
```

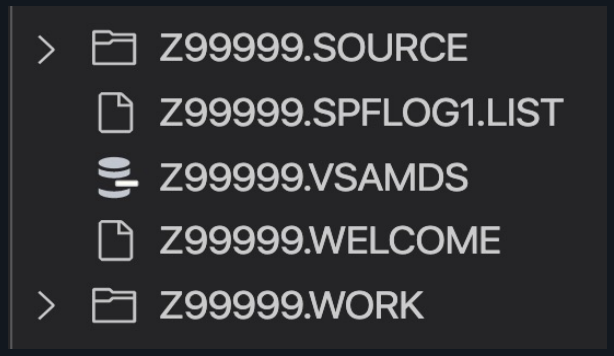
10. As chaves para nossos dados

Um tipo de conjunto de dados que você viu nos menus do Zowe é **VSAM**, e merece atenção especial. O VSAM não é usado para armazenar mensagens JCL ou "Bem-vindo ao Mainframe". Sua hora de brilhar é quando um aplicativo precisa

acessar os registros da forma mais rápida e eficiente possível. Na verdade, sem um software especial para interpretar arquivos VSAM, você não pode abri-los em um editor normal, mas os aplicativos comem alegremente

esses arquivos para cima.

É tudo uma questão de eficiência no acesso aos dados. Leia mais abaixo.



11. Crie um conjunto de dados VSAM

Você está ficando bom em alocar conjuntos de dados. Faça um dado VSAM conjunto chamado Zxxxxx.VSAMDS . Consulte o Zdevo ajuda online para um guia para o comando.

Quando terminar, olhe para seus atributos (você sabe como) e você notará algo bem interessante; parece que existem TRÊS conjuntos de dados aqui. Além disso, se você visualizá-lo em sua lista de conjuntos de dados

no VS Code , você verá um novo ícone elegante. Curioso ainda? Vamos prosseguir.

```
1IDCAMS  SYSTEM SERVICES
0
      REPRO -
            INFILE(INPUT) -
            OUTDATASET(Z99999.VSAMDS)-
            ERRORLIMIT(6)
0IDC0005I NUMBER OF RECORDS PROCESSED WAS 1000
0IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0
0
0IDC0002I IDCAMS PROCESSING COMPLETE. MAXIMUM CONDITION CODE WAS 0
```

12. Carregue-o com registros

Em seguida, vamos adicionar alguns registros. Você pode usar o existente dados de amostra em**'ZXP.PUBLIC.SAMPDATA'**,ou você pode se divertir e fazer o seu próprio. Mockaroo.com tem um bom gerador de dados que você pode experimentar, embora com algumas notas:

1) A primeira coluna (as "chaves") deve estar em ordem

2) Omita quaisquer registros/linhas em branco

3) Você precisará de zeros à esquerda para chaves, caso contrário, o VSAM pode não vê-los como estando em ordem quando você tenta importar

4) Certifique-se de que esses novos dados de entrada estejam armazenados em um conjunto de dados zOS.

Baixe a amostra**REPRO**membro JCL do conjunto de dados ZXP.PUBLIC.JCL para sua estação de trabalho pessoal, colocando-o na pasta ou diretório em que você está trabalhando no momento. Observe que

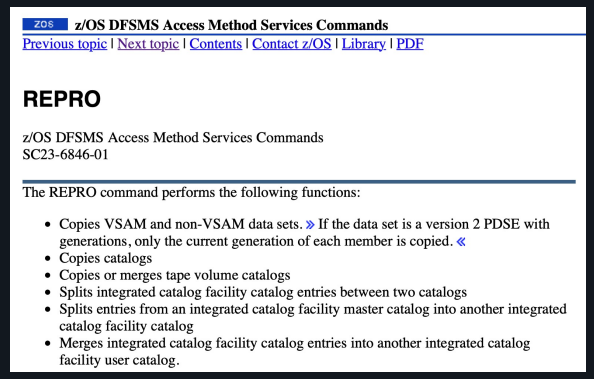
se você criou seu próprio conjunto de dados de entrada, precisará editar a JCL para apontar para seu conjunto de dados de origem. Nomeie seu arquivo REPRO como **repro.txt**

Usar**zowe jobs enviam o arquivo local "repro.txt"**para enviar o JCL diretamente de sua máquina, através do Zowe CLI. Você verá uma pequena animação e um número de trabalho. Verifique esse número de trabalho e certifique-se de que funcionou sem problemas.

Aqui estou. Alocar-me como VSAM

VSAM é complicado, e esta pequena caixa cinza não lhe dará anos de experiência trabalhando com conjuntos de dados VSAM, mas lhe dirá que se você quiser esse trabalho de mainframe, faça toda a leitura e prática com conjuntos de dados VSAM que você puder . Eles são um componente central de qualquer grande empresa de mainframe.

Por enquanto, saiba que existem quatro tipos principais de conjuntos de dados VSAM, KSDS (sequência de chave), ESDS (sequência de entrada), RRDS (registro relativo) e Linear (LDS). KSDS e ESDS são os mais comuns, e a diferença se resume a como cada registro é armazenado e acessado. KSDS significa que você faz referência a uma chave (como procurar um número de conta) e obtém as informações dessa conta como registro. O ESDS armazena dados em uma ordem sequencial, para dados que provavelmente serão lidos um após o outro em uma ordem específica. Isso é o suficiente por enquanto, mas se você ainda estiver com fome,hhá mais alguns a considerar .



z/OS z/OS DFSMS Access Method Services Commands

Previous topic | Next topic | Contents | Contact z/OS | Library | PDF

REPRO

z/OS DFSMS Access Method Services Commands
SC23-6846-01

The REPRO command performs the following functions:

- Copies VSAM and non-VSAM data sets. » If the data set is a version 2 PDSE with generations, only the current generation of each member is copied. «
- Copies catalogs
- Copies or merges tape volume catalogs
- Splits integrated catalog facility catalog entries between two catalogs
- Splits entries from an integrated catalog facility master catalog into another integrated catalog facility catalog
- Merges integrated catalog facility catalog entries into another integrated catalog facility user catalog.

13. Vamos fazer um inventário

Vamos falar sobre o que acabamos de fazer. O JCL executa IDCAM, que é usado principalmente para gerenciar conjuntos de dados VSAM. Dentro do IDCAM, estamos usando oREPRO comando para carregar um conjunto de dados sequencial em um conjunto de dados formatado em VSAM. (Há MUITA complexidade acontecendo aqui que não vemos, mas assim como antes, há muitas oportunidades para controlar exatamente como você deseja que essa cópia aconteça, incluindo parâmetros criptográficos.

Os dados são os mesmos, mas agora são estruturados fundamentalmente diferentes, indexados por chave e podem ser referenciados muito mais eficientemente por programas (incluindo aqueles escritos em REXX)

Na realidade, esses dados não estão muito bem indexados, pois cada linha é sua própria chave, mas se mergulharmos no particulares da construção de um cluster VSAM , você pode ver como as chaves e o tamanho do registro podem ser especificados.

```
-LISTING OF DATA SET -Z99999.VSAMDS
0KEY OF RECORD - 001354719770    HUBERT    DEMONGEOT    87-8997183    #D230E7
 001354719770    HUBERT    DEMONGEOT    87-8997183    #D230E7    1GKMCCE34AR94
0KEY OF RECORD - 001359581404    AGNESE    FARRANCE    56-4110060    #9A9D61
 001359581404    AGNESE    FARRANCE    56-4110060    #9A9D61    SCFAB01A76G165
0KEY OF RECORD - 001362199763    STEPHEN    TODHUNTER    79-5179893    #906724
 001362199763    STEPHEN    TODHUNTER    79-5179893    #906724    WBAUT9C57BA3
0KEY OF RECORD - 001369213008    REAGEN    MCILWRICK    01-1405738    #619A1B
 001369213008    REAGEN    MCILWRICK    01-1405738    #619A1B    JTEBU5JRXF518
0KEY OF RECORD - 001384380151    BENNY    LAMBIS    83-6731093    #586AEC    2
 001384380151    BENNY    LAMBIS    83-6731093    #586AEC    2C3CCAEG5FH726459
0KEY OF RECORD - 001398310239    RAHAL    PENNYCORD    14-4881973    #03E5B3
 001398310239    RAHAL    PENNYCORD    14-4881973    #03E5B3    WBAVC73508A963
0KEY OF RECORD - 001399406486    ARIDATHA    TOSELAND    01-9975566    #69E914
 001399406486    ARIDATHA    TOSELAND    01-9975566    #69E914    JM1GJ1T68E11
0KEY OF RECORD - 001401405570    ORALLE    KIMMINS    66-8864767    #6198F7
 001401405570    ORALLE    KIMMINS    66-8864767    #6198F7    SAJWA4GB7EL9541
0KEY OF RECORD - 001409084356    LYNNE    COLLCOTT    94-7549269    #5B0003
 001409084356    LYNNE    COLLCOTT    94-7549269    #5B0003    1B3CC5FB8AN6801
0KEY OF RECORD - 001413763279    ROUVIN    APNALDI    05-6394813    #99A54E
```

14. Imprimindo registros

Vamos usar mais um comando IDCAMS para examinar nossa saída, o P apropriadamente chamadoRINT comando e cdane-se isso exemplo (dica) e preste atenção no CParâmetro HARACTER (dica) para que sua saída se pareça com a captura de tela acima. Você reunirá informações de várias fontes aqui, então pense no que você tem e no que deseja. Você desejará imprimir esse conjunto de dados VSAM em formato de caractere. Isso ajuda? Não tenha medo de parar pelos fóruns para alguma ajuda.

```
1  [illegible]                                                    TIME: 15:13:21
2  0
3       PRINT -
4            [illegible]
5            [illegible]
6            [illegible]
7  [illegible]                                                    TIME: 15:13:21
8  -LISTING OF DATA SET - [illegible].VSAMDS
9  0KEY OF RECORD - [illegible]
10  000003644195    [illegible]
11 0KEY OF RECORD - [illegible]
12  000004292952    [illegible]
13 0KEY OF RECORD - [illegible]
14  000006738499    [illegible]
15 0KEY OF RECORD - [illegible]
16  000011843626    [illegible]
17 0KEY OF RECORD - [illegible]
18  000016940643    [illegible]
19 0KEY OF RECORD - [illegible]
20  000019874197    [illegible]
```

15. Faça valer a pena

Venha para o Zowe CLI, fique para o VSAM e IDCAMS. Para completar isso, estamos procurando por 3 coisas:

1) Seu**Zxxxxx.ZOWEPS**conjunto de dados sequenciais
2) Seu**Zxxxxx.VSAMDS**Conjunto de dados VSAM
3) As primeiras 20 linhas de saída do seu comando IDCAMS PRINT, copiadas/coladas em um **Zxxxxx.OUTPUT.VSAMPRNT**conjunto de dados. Não escreva este conjunto de dados diretamente do seu JCL, use SYSPRINT e copie/cole as linhas 1-20 do seu SYSPRINT. Precisamos do cabeçalho. Consulte a captura de tela acima como um exemplo (levemente editado).

Quando tiver concluído a tarefa, envie**CHKAZCLI**

BOM TRABALHO! VAMOS RECAPITULAR

Você veio para este desafio*provavelmente*sem saber muito sobre ZCLI, e estão saindo sabendo não só como se locomover no Zowe CLI, mas um pouco sobre VSAM e IDCAMS.

Você provavelmente também notou o nível de instrução começando a mudar de "aqui está um comando" para "descobrir isso". Bem-vindo às grandes ligas, é assim que rolamos agora.

PRÓXIMO...

Agora que o Zowe CLI está em seu kit de ferramentas e você tem algumas experiência VSAM, vamos dar mais alguns detalhes sobre como o VSAM funciona.