

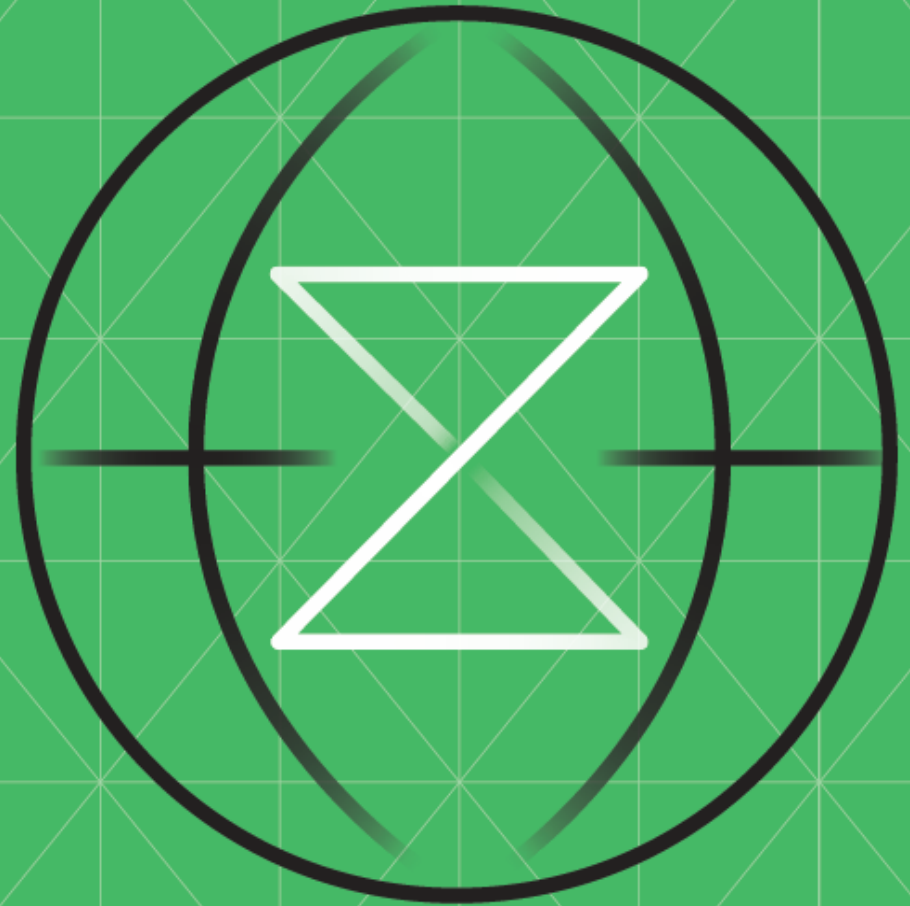
Extended

231021-0727

G02

Get Going on z/OS

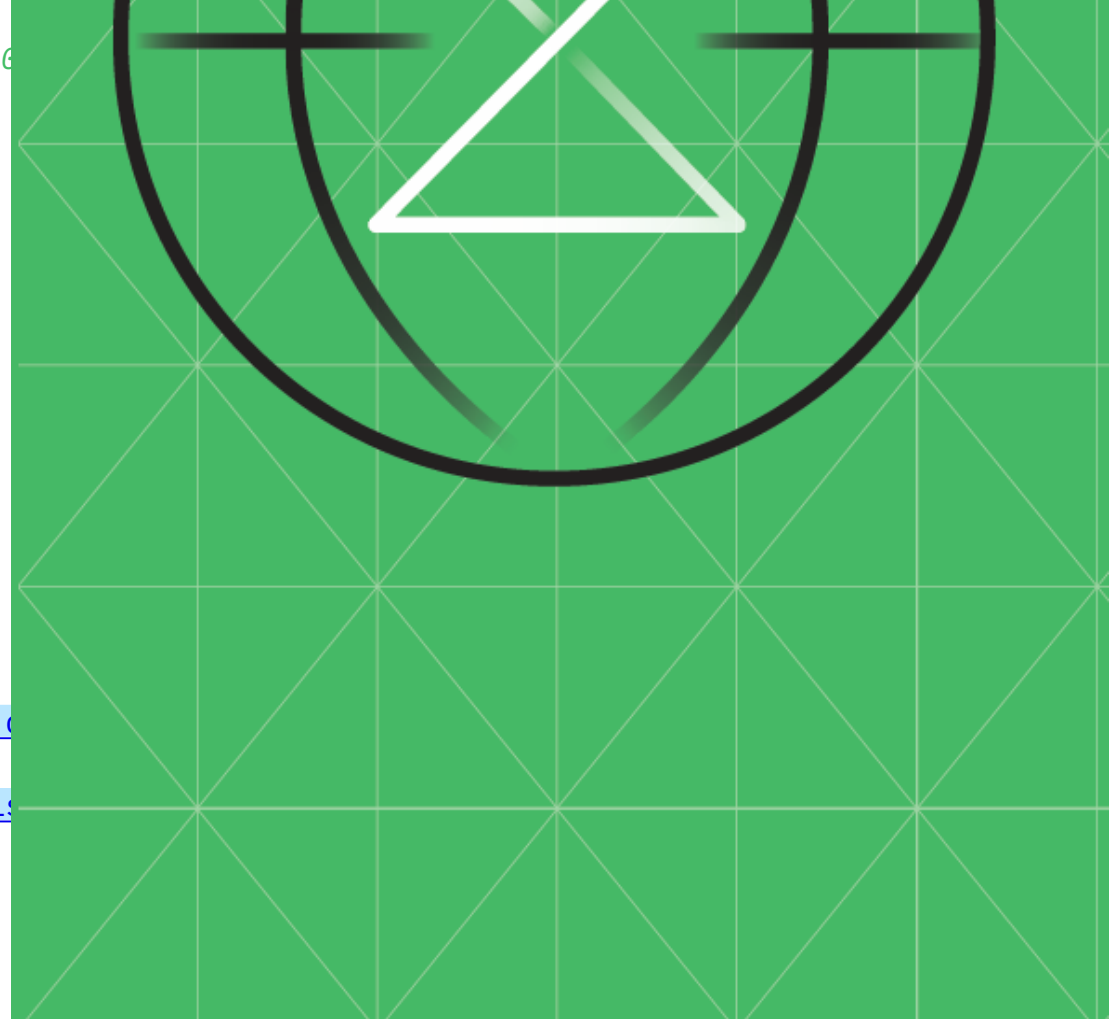
Automated Translation by Watson Language Translator



G02

Vá para o z/OS

- [Usando Go com VSAM](#)
 - [0 desafio](#)
 - [Antes de começar](#)
 - [Investimento](#)
- [1 Antes de começar](#)
 - [1.0.1 Ir Módulo de Gravação](#)
 - [1.0.2 Nosso Módulo de Pessoas](#)
- [2 Terminologia](#)
- [3 Acesse o pacote de ferramentas](#)
- [4 Parte 1: lendo e gravando um conjunto de c](#)
- [5 Parte 2: Estendendo a Funcionalidade](#)
- [6 Parte 3: Implementando a Exclusão de Regis](#)
- [7 Teste sua compreensão: Ajustes finais](#)
- [8 Vá, vá!](#)



USANDO GO COM VSAM

Mergulhando mais profundamente no Go trabalhando com registros do VSAM

0 Desafio

Neste desafio, você estará trabalhando com um projeto Go que envolve manipular registros VSAM (Virtual Storage Access Method) usando registros KSDS (Key-Sequenced Data Set). Neste desafio Go, você está encarregado de criar um banco de dados de pessoal para um departamento de Recursos Humanos.

Antes De Começar

Se você não tiver feito Go1 ou não tiver feito isso há algum tempo, por favor, volte e atualize suas habilidades.

INVESTIMENTO

Etapas	Duração
8	120 minutos

602|231 021-0727

1 ANTES DE COMEÇAR

Go é uma nova linguagem de programação popular e é usado devido à sua simplicidade e fácil de entender. O Go permite que desenvolvedores de aplicativos entreguem aplicativos em um tempo menor com menos linhas de código.

Para este desafio, você estará trabalhando com três arquivos: `one.go`, `two.go`, e `three.go` que finalmente constroem registros em um banco de dados de funcionários. Através destes programas você estará aprendendo sobre o `Go recordio module`. Você estará acessando registros VSAM enquanto entende como usar funções que estão no mesmo pacote ou em pacotes diferentes.

1.0.1 Ir Módulo De Gravação

Go-recordio é um módulo Go especializado em Virtual Access Storage Methods (VSAM) de entrada e saída (I/O). Ele fornece a capacidade de acessar esses registros diretamente do Go. Você usará os dois pacotes na pasta v2- `zosrecordio` para funções de E/S e `utils` para suportar funções usadas em casos como conversão de tipo de dados. É possível visualizar onde a v2 está `go.mod`.

1.0.2 Nosso Módulo De Pessoas

Usando o módulo go-recordio, você estará interagindo com um conjunto de dados VSAM KSDS. Ao trabalhar com KSDS, você estará trabalhando com `keys`. Essas chaves são exclusivas e são usadas para localizar um registro

1. Copie o arquivo `go2` diretório de `/z/public` em seu diretório local:

```
cp -R /z/public/go2 /z/ZID/go2
```

Neste projeto Go, você encontrará:

Um `go.mod` para gerenciar dependências em versões do Go e dependências externas

Você não precisará editar isso, mas você pode ver como nós o usamos para encontrar um módulo sem fazer download e como nós nomeamos nosso módulo Pessoas.

Um `main.go` que contém a função `main()`. Este é o seu ponto de entrada.

Este é o lugar onde você vai executar seus arquivos Go

Um `JCL` pasta contendo código JCL para criar seu conjunto de dados VSAM KSDS

O `Tools` fornece funções de utilitário para interagir com registros. Você verá o uso de `tools` dentro do seu `.go` arquivos

Este será o local onde você faz a maior parte do seu trabalho e altera o código quando solicitado a

2. Envie sua tarefa JCL executando

```
submit newksds.jcl
```

Certifique-se de estar no diretório correto que contém a JCL. O envio dessa tarefa está criando seu cluster VSAM.

1. Verifique se sua tarefa foi enviada com um código de condição 00.

Links úteis

[Conjuntos de Dados KSDS](#)

[Cabeçalhos de Go](#)

[Saída de Entrada Padrão.](#)

[Resumo de operações de E/S VSAM](#)

[Atualizar registros do VSAM](#)

2 TERMINOLOGIA

Alguma terminologia para referência em todo o desafio:

- `struct` : Uma estrutura (ou estrutura) é um tipo definido pelo usuário que permite agrupar itens de diferentes tipos em um único tipo. A estrutura contém uma lista de campos com um nome e um tipo..

Você pode ver no exemplo `one.go`, você está definindo uma estrutura que tem 5 nomes (Id, FN, LN, Posição, Departamento) com um tipo de tamanho fixo.

Lembre-se de que os registros KSDS são baseados em uma chave. Nossa chave aqui é Id. Você pode ver que a chave tem um tamanho fixo de 6.

- `buffbytes`: Um ponteiro para um espaço de dados que pode ser tratado como uma fatia de byte de tipo.. Optamos por dar este nome a uma fatia de bytes (notada por `[] byte`) que usamos como um buffer. As funções de gravação que estamos usando requerem que os dados sejam enviados e recebidos como uma fatia de byte. Você verá a que `buffbytes` é atribuído em `one.go`.
- `ConvertTypetoSlice`: Usado para converter qualquer tipo que ele é fornecido e cria uma fatia de byte A conversão permite usar os dados de estrutura como uma fatia de byte. A estrutura e a fatia de bytes compartilham a mesma memória
- `PeopleHeader`: Por conveniência, usamos essa estrutura para organizar nossos dados antes de convertê-los em uma fatia de byte e enviá-los para serem gravados ou lidos. O cabeçalho da palavra é incluído como uma confirmação para o conceito de que um cabeçalho, o primeiro registro em um cluster KSDS, define o layout para cada registro subsequente
- `stream`: Um fluxo é usado para identificar um registro Um fluxo gerencia o fluxo de dados e é usado para ler e gravar um registro É uma conexão aberta com o banco de dados que você criou em seu JCL (conjunto de dados KSDS).

602|231.021-0727

3 ACESSE O PACOTE DE FERRAMENTAS

Neste desafio, você estará fingindo usar uma ferramenta de RH para registrar informações sobre funcionários, cargos, departamentos e muito mais. Você estará incluindo em registros com esses detalhes em `one.go`, `two.go`, e `three.go`.

Antes de mergulhar nisso, você vai querer olhar para `tools.go` pacote. Este pacote é referenciado pelos diferentes programas Go neste desafio para ler, gravar e excluir uma estruturação.

Você também pode atualizar um registro se você gostaria de tentar isso após o desafio ser concluído. criar uma estrutura, atualizar o registro e excluir o registro.

Você pode notar que algumas funções e variáveis começam com nomes / identificadores que são capitalizados e outros que não. Isso ocorre porque identificadores que são capitalizados representam entidades que são exportáveis para outros pacotes.

1. Olhe para `tools.go` pacote
2. Leia o código e os comentários para entender o que está acontecendo dentro do pacote
3. Examine o pacote e tente entender o propósito e como as funções estão trabalhando juntas. Tudo bem se você ainda não entender como tudo se conecta. Você estará referenciando este pacote durante todo o desafio e o entenderá melhor à medida que você for adiante. Esteja preparado para alterar partes do código dentro do pacote quando solicitado.

4 PARTE 1: LENDO E GRAVANDO UM CONJUNTO DE DADOS

1. Comece examinando `one.go` para entender sua funcionalidade.

`one.go` é sua introdução ao go-recordio. Neste programa, você está incluindo um registro de um funcionário usando uma estrutura.

2. Observe que este pacote está sendo importado de `/people/tools`. Este é o mesmo pacote que você viu um passo antes. As funções que são extraídas do pacote `util` serão iniciadas com `tools`.
3. Vamos executar este pacote e ver se algum bug aparece. Inclua o `one.go` função em `main.go`.

Dica: Encontre o arquivo `main.go` e vá até a parte inferior. Você não precisará mudar muito

4. Depois de ter integrado a parte 1 no `main.go`, no seu terminal execute o comando `go build` para compilar o projeto..
5. Execute o programa inserindo `./people "'zID.VSAMDS'"` no terminal. Você está usando `./people` como o nome executável devido ao que é especificado no `go.mod` para o módulo.

Nota: Se você atualizar sua visualização USS, você verá as pessoas executáveis aparecerem no diretório `go2`. Se você clicar nisso, você verá uma mensagem de erro aparecer. Isso não significa que o seu executável está errado, a mensagem de erro está correta. Continue com o desafio.

6. O que você vê? Se o programa foi executado, você verá que ele foi executado corretamente, mas algumas das saídas não parecem legíveis. Sua fatia impressa legível, mas sua estrutura não. Ele está retornando valores decimais (ASCII) e você deseja que ele emita uma sequência dos dados do funcionário inseridos em `one.go`. Você terá que fazer mudanças no **1 linha de código** para `tools.go`. São pequenas alterações, que envolvem certificar-se de que cada TYPE na função de estrutura de impressão retorne um STRING.

Dica: Olhe para a sua função de estrutura de impressão em `tools.go`.

Dica: Olhe a linha 62 em `one.go`

7. Compile e execute seu programa quando tiver corrigido a saída.
8. Você deve ter uma saída semelhante à imagem abaixo.

```
Struct: 000001 - Fname - Lname - TestPosition - SampleDepartment  
Flocate succesful  
slice 000001FnameLnameTestPositionSampleDepartment  
  
End: Part one.
```

Se você tem uma saída que se parece com a imagem abaixo, então continue tentando! Certifique-se ao fazer uma mudança que você execute o comando `go build` para recompilar o código.

```
/z/z28604/go2-new > go build  
/z/z28604/go2-new > ./people "'/'Z28604.VSAMDS'"  
Struct: [48 48 48 48 48 49] - [70 110 97 109 101 0 0 0 0 0 0 0 0 0 0] - [76 110 97 109 101 0 0 0 0 0 0 0 0 0  
0 0] - [84 101 115 116 80 111 115 105 116 105 111 110 0 0 0 0] - [83 97 109 112 108 101 68 101 112 97 114 116  
109 101 110 116]  
Flocate succesful  
slice 000001FnameLnameTestPositionSampleDepartment  
  
End: Part one.
```

5 PARTE 2: ESTENDENDO A FUNCIONALIDADE

1. Localizar `two.go` e ler o programa

Em `two.go` você está lidando com uma fatia de estruturas em vez de apenas uma única entrada.

2. Semelhante ao que você fez para `one.go`, incorpore a função de `two.go` em `main.go` e construir, em seguida, executá-lo
3. Você ainda não terminou! Inclua informações de um membro da equipe adicional na matriz de estruturas.

```
ID: 123452
First Name: Melvin
Last Name: Mainframe
Position: Librarian
Department: Library
```

4. Compile e execute o seu programa novamente Certifique-se de que sua nova estrutura tenha sido incluída em sua saída antes de avançar.

602|231 021-0727

```
Writing struct: 0
Writing struct: 1
Writing struct: 2
Struct: 000001 - Fname - Lname - TestPosition - SampleDepartment
Struct: 123451 - Alice - Sand - Actor - Film
Struct: 123452 - Melvin - Mainframe - Librarian - Library
Struct: 123456 - Eric - Waters - Artist - Anthropology

End: Part two.
```

6 PARTE 3: IMPLEMENTANDO A EXCLUSÃO DE REGISTRO.

Agora que você entende como construir registros a partir de estruturas, tanto singulares quanto uma matriz, você vai praticar a exclusão de um registro.

1. Você conhece o exercício. Incluir a função de `three.go` em `main.go` e executar o programa.
2. Observe em sua saída que a funcionalidade de exclusão está incompleta Isso é o que você estará adicionando em seguida.
3. Consulte as etapas descritas no link fornecido: [KSDS Excluindo Registro](#)

Isso ajudará a entender o processo de exclusão de registros usando operações VSAM. Consulte a coluna KSDS e localize a linha de exclusão (registro) Você estará usando essas informações a seguir:

4. Atualize a função de exclusão de registro no `tools.go` para excluir com êxito o registro pretendido

Você precisará remover comentário **FOUR** linhas na função excluir

Em duas das quatro linhas, você precisará incluir o restante do procedimento de exclusão. Consulte as etapas no KSDS que excluem etapas de registro e veja quais já foram feitas antes de chamar a função e quais permanecem.

Se precisar de mais ajuda, utilize as funções disponíveis no go-recordio módulo. Isso pode ser encontrado dentro do pacote zosrecordio na v2. [Go-recordio](#)

Sugestão: As duas primeiras linhas que você descomenta fornecem uma dica sobre qual variável passar quando você adiciona comandos de leitura e exclusão.

Dica: Use a linha 64 para saber como formatar seus comandos

5. Verifique a exclusão usando o loop de impressão.. Você deve ver algo semelhante à captura de tela abaixo.

```
Good floccate test
Struct: 000001 - Fname - Lname - TestPosition - SampleDepartment
Struct: 123451 - Alice - Sand - Actor - Film
Struct: 123452 - Melvin - Mainframe - Librarian - Library
Struct: 123456 - Eric - Waters - Artist - Anthropology
Good floccate ready for delete

Attempting delete

Struct: 123451 - Alice - Sand - Actor - Film
Struct: 123452 - Melvin - Mainframe - Librarian - Library
Struct: 123456 - Eric - Waters - Artist - Anthropology
Confirmed record deleted

End: Part three.
```

7 TESTE SUA COMPREENSÃO: AJUSTES FINAIS

Agora, você deve se sentir familiarizado com os três arquivos Go e como eles estão usando tools.go para executar.

No início do desafio, você adicionou no `newksds.jcl` e você estiver usando o banco de dados criado (conjunto de dados KSDS) para executar seu executável..

Como o banco de dados obtém seus atributos quando ele é criado?

1. Consulte a JCL e descubra o que está acontecendo na tarefa.
2. Observe o que está acontecendo no

`SYSIN DDNAME`. É possível ver isso se o conjunto de dados `&SYSUID..VSAMDS` existe, ele será excluído. Em seguida, um novo conjunto de dados é criado com os parâmetros especificados

Você aprendeu no início do desafio que um conjunto de dados do KSDS possui cada um de seus registros identificados por uma chave. A chave de cada registro é exclusiva e predefinida no registro. No seu caso, a chave é o ID e tem seis dígitos.

3. Modifique as chaves de registro encurtando o número da chave principal (ID) permitindo apenas 3 caracteres. Você terá que alterar quaisquer valores de chave em `one.go`, `two.go`, `three.go`. Também será necessário certificar-se de que o valor da chave seja 3 ao criar o conjunto de dados e no `tools.go` onde você lista o valor da chave.

No geral, haverá 8 mudanças totais dentro de 5 arquivos diferentes.

Quando você estiver alterando as chaves no `.go` arquivos, **fique com os três últimos** números.

Certifique-se de reenviar seu JCL se você mudou alguma coisa nesse arquivo

4. Construa e execute o programa novamente Sua saída contém apenas os registros com 3 chaves de caracteres? Se não, continue tentando. Você deve ter um total de 3 registros.

```
Struct: 451 - Alice - Sand - Actor - Film  
Struct: 452 - Melvin - Mainframe - Librarian - Library  
Struct: 456 - Eric - Waters - Artist - Anthropology  
Confirmed record deleted
```


8 VÁ, VÁ!

Agora você aprendeu como usar go-recordio com conjuntos de dados VSAM. Você está familiarizado com estruturas de construção, incluindo novos registros e excluindo registros.

Certifique-se de que você tenha:

- Criado o executável para one.go, two.go e three.go
- Adicionado em seu registro do Melvin Mainframe
- ID da chave alterado para 3
- Excluído o registro 001 em three.go
- Saída com um total de 3 registros, com os 3 espaços de caracteres atualizados

Enviar `CHKG02` para assegurar que você tenha feito tudo corretamente antes de concluir o desafio no IBM Z Xplore

Bom trabalho-vamos recapitular	Em seguida ...
<p>Parabéns!</p> <p>Você acabou de aprender uma introdução básica aos registros do VSAM usando o Go. Através destes exercícios, você aprendeu:</p> <ul style="list-style-type: none"> • Como usar pacotes do Go • Como um módulo usa pacotes • Como usar o go-recordio • Como criar e acessar registros VSAM usando o Go • Sobre arquivos go.mod • Como criar estruturas no Go • Como criar, excluir e atualizar registros • Como usar funções que estão em pacotes iguais ou diferentes 	<p>Há muito mais recursos Go para aproveitar no z/OS. Você pode tentar usar o UpdateRecord em tools.go para experimentar a atualização de um registro existente. Você também pode visitar https://go.dev/learn/ para saber mais sobre o Go e fazer uma visita turística.</p>

GO2/231 021-0727