Gofish – A Golang Client Library for Swordfish

Sean McGinnis
Dell EMC
Features

- Hides internal details from consumer
- Presents basic object model
- Attempts to handle implementation specific differences so users don’t need to
Background Information
What the Heck is Go?

- Statically typed language from Google
- Fast, efficient, and portable
  - Native concurrency
  - Target different platforms at compile time

```go
package main

import (  
    "fmt"
)

func main() {  
    message := "Hello, SDC!!"
    fmt.Println(message)
}
```
Gofish History

- Interest in support in OpenSDS project
- Open Source Liaison group efforts
- Open source way – saw a need and wanted to “scratch the itch”
Redfish Overview

- Build off of capabilities of IMPI, SMASH, etc.
- Use modern, interoperable technology
  - REST (HTTP)
  - JSON data representation
  - Clearly defined object schema
Swordfish Overview

- Built on top of Redfish specification
- Adds objects for storage management
- Close collaboration with DMTF Redfish team
Object Model
Swordfish Objects

- /redfish/v1
  - ServiceRoot
    - /StorageServices/1
      - Name
      - Description
      - Status
      - /Drives/1
      - /Endpoints/1
      - /Enclosure/1
      - /Volumes/1
      - /StoragePools/1
      - /FileSystem/1
Demo
https://github.com/stmcginnis/gofish-demo
Contributing

- Typical GitHub workflow:

  $ git clone https://github.com/stmcginnis/gofish
  $ git checkout -b [branch_name]
  $ git push [your_personal_github_account]
Thanks to Contributors!

Juliano Martinez
ncode

Mike Pittelko
mike-pittelko
Call to Action

- Try it out!
- Report any issues via the GitHub issue tracker
- Submit pull requests for any updates and fixes
- Spread the word!
Links and References

https://github.com/stmcginnis/gofish
https://redfish.dmtf.org/
https://www.snia.org/forums/smi/swordfish
Please take a moment to rate this session.

Your feedback matters to us.