

Redfish Schema Demo

Przemyslaw Czarnowski, Maciej Lawniczak

February 1, 2020

Question

- Is it difficult?

Question

- Is it difficult?
 - no, but...

Today's show

What is done?

- simple OEM Schema "FortuneGenerator"
- backend, serving 2 kinds of Fortunes/adages
- shows fortune
- changes text on action "Generate"

Redfish:

Concepts

- OData standard
- schema definitions CSDL/json

Redfish schemas

- structure, links
- mockups

Resources

- OData: <https://www.odata.org/documentation/>
- Main page: <https://www.dmtf.org/standards/redfish>
- Schemas definitions: <https://www.dmtf.org/schemas/v1>
- Mockups: <https://redfish.dmtf.org/redfish/v1>

OpenBMC implementation

- C++17
- sdbusplus (dbus library)
 - <https://github.com/openbmc/sdbusplus>
- boost
 - ASIO
 - coroutines
- one thread

DBus

- `xyz.openbmc_project.<SERVICE-NAME>` namespace
- `org.freedesktop.DBus.ObjectManager` interface

systemd

- `.service` file

Schema definition

- Redfish forum
 - <https://redfishforum.com>
- Redfish tools repository
 - <https://github.com/DMTF/Redfish-Tools/>
 - CSDL to JSON Converter
- `<bmcweb>/static/redfish/v1`
 - `schema/` for xml
 - `JsonSchema/` for json
 - `$metadata` to add schema

Where

- `<bmcweb>/redfish-core/lib` implementations
- file naming
 - `VirtualMedia -> virtual_media.hpp`
 - `FortuneGenerator -> fortune_generator.hpp`

What

- Node class
 - base of all redfish nodes

What

- **Collection**
 - List of items available
- Item
 - properties of item
- Actions (optional)
 - to perform action on given object

```
{
  "@odata.context": "/redfish/v1/$metadata#FortuneGeneratorCollection.FortuneGeneratorCollection",
  "@odata.id": "/redfish/v1/Systems/systems/FortuneGenerator/",
  "@odata.type": "#...rCollection.FortuneGeneratorCollection",
  "Members": [
    {
      "@odata.id": "/redfish/v1/...Generator/FortuneQuestions"
    },
    {
      "@odata.id": "/redfish/v1/...Generator/FortuneComputer"
    }
  ],
  "Members@odata.count": 2,
  "Name": "Fortune Generator Services"
}
```

What

- Collection
 - List of items available
- **Item**
 - properties of item
- Actions (optional)
 - to perform action on given object

```
{
  "@odata.context": "...FortuneGenerator",
  "@odata.id": "...ator/FortuneComputer",
  "@odata.type": "...0_0.FortuneGenerator",
  "Actions": {
    "#FortuneGenerator.Generator": {
      "target": "/redfish/v1/Systems/system/FortuneGenerator/
FortuneComputer/Actions/FortuneGenerator.Generate"
    }
  },
  "Fortune": "fortune If it happens ...",
  "Id": "FortuneComputer",
  "Name": "Fortune Generator"
}
```

What

- Collection
 - List of items available
- Item
 - properties of item
- **Actions** (optional)
 - to perform action on given object

```
"Actions": {  
  "#FortuneGenerator.Generator": {  
    "target": "/redfish/v1/Systems/system/FortuneGenerator/  
FortuneComputer/Actions/FortuneGenerator.Generate"  
  }  
}
```

How

- Registration
 - classes registered in
`<bmcweb>/redfish-code/include/redfish.hpp`
 - file included
 - emplaced unique pointer in node list
- Helpers
 - Many helpers created to simplify implementation
 - messages classes
 - json utilities
 - lots of boilerplate code (copy/paste programming)
- Compatibility check
 - Redfish-Service-Validator

Code walkthrough

```
http --auth root:OpenBmc --auth-type=basic --verify=no \  
--body GET https://redfishhost:443/redfish/v1/Systems/system
```

```
http --auth root:OpenBmc --auth-type=basic --verify=no \  
--body GET https://redfishhost:443/redfish/v1/Systems/system\  
/FortuneGenerator
```

```
http --auth root:OpenBmc --auth-type=basic --verify=no \  
--body GET https://redfishhost:443/redfish/v1/Systems/system\  
/FortuneGenerator/FortuneComputer
```

```
http --auth root:OpenBmc --auth-type=basic --verify=no \  
--body POST https://redfishhost:443/redfish/v1/Systems/system\  
/FortuneGenerator/FortuneComputer/Actions/FortuneGenerator.Generate
```

```
http --auth root:OpenBmc --auth-type=basic --verify=no \  
--body GET https://redfishhost:443/redfish/v1/Systems/system\  
/FortuneGenerator/FortuneComputer
```

Thank you

Thank you

BusCtl commands to inspect service

Some are counterparts to ones used in Redfish implementation

```
busctl --no-pager --json=pretty tree xyz.openbmc_project.FortuneGenerator
```

```
busctl --no-pager --json=pretty introspect xyz.openbmc_project.FortuneGenerator \
/xyz/openbmc_project/FortuneGenerator
```

```
busctl --no-pager --json=pretty introspect xyz.openbmc_project.FortuneGenerator \
/xyz/openbmc_project/FortuneGenerator/FortuneComputer
```

```
busctl --no-pager --json=pretty call xyz.openbmc_project.FortuneGenerator \
/xyz/openbmc_project/FortuneGenerator org.freedesktop.DBus.ObjectManager GetManaged
```

```
busctl --no-pager --json=pretty call xyz.openbmc_project.ObjectMapper \
/xyz/openbmc_project/object_mapper xyz.openbmc_project.ObjectMapper \
GetObject sas /xyz/openbmc_project/FortuneGenerator 0
```

```
busctl --no-pager --json=pretty call xyz.openbmc_project.FortuneGenerator \
/xyz/openbmc_project/FortuneGenerator/FortuneComputers \
xyz.openbmc_project.FortuneGenerator.Fortune Generator
```

Some links to use with browser to conduct demo

```
http --print=hb --auth root:OpenBmc --auth-type=basic --verify=no POST \  
https://redfishhost:8443/redfish/v1/Systems/system/FortuneGenerator/FortuneComputer/Actions/  
FortuneGenerator.Generate | grep "^HTTP/"
```

```
https://redfishhost:8443/redfish/v1
```

```
https://redfishhost:8443/redfish/v1/Systems
```

```
https://redfishhost:8443/redfish/v1/Systems/system
```

```
https://redfishhost:8443/redfish/v1/Systems/system/FortuneGenerator
```

```
https://redfishhost:8443/redfish/v1/Systems/system/FortuneGenerator/FortuneComputer
```

```
https://redfishhost:8443/redfish/v1/Systems/system/FortuneGenerator/FortuneQuestion
```