Redfish Sessions

Jeff Hilland
HPE

Redfish

www.dmtf.org

Agenda

- Sessions Overview
- Sessions Resource Overview and Map
- Examples:
  - Session Service
  - Starting a Session
  - Stopping a Session

Authentication and Redfish Sessions

- Very few Redfish Resources are unprotected
  - ServiceRoot and POSTs to the Sessions Collection are the two main ones
    - There are few others: the version object, the Odata Service Document and $metadata, but these are not important to this presentation.
  - You need to provide user credentials to the Redfish Service to access the rest of the resources. If you don't, you will get a 401 Forbidden error.
- How are credentials sent to the Redfish Service?
  - There are basically two ways
  - Basic Authorization
    - Uses the Authorization header and follows RFC 7235.
    - Typically a base-64 encoding of the username, a colon followed by the password.
    - Equivalent of logging in and logging out on each operation.
  - Session Establishment
    - Uses the X-Auth-Token header.
    - This header is retrieved by POSTing credentials to the Sessions Collection.
    - It is then used on subsequent operations.
    - Equivalent of logging in.

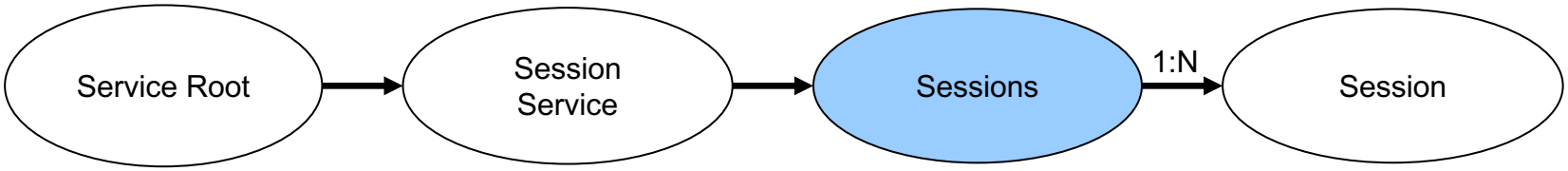
Basic Auth vs. Sessions

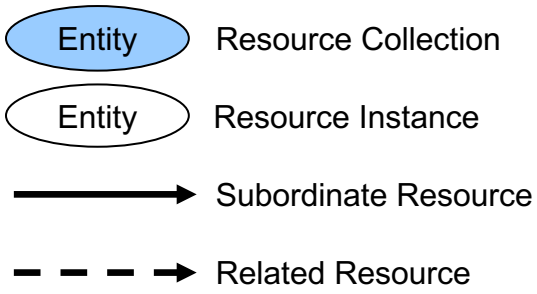
- What are the differences between Basic Authorization and Sessions?
  - Both of them are going to use the credentials specified to determine what operations are enabled for the particular user account
- Basic Authorization
  - What this is actually doing is setting up authorization, performing the operation and then tearing down the authorized connection
  - This works fine if you are only doing a quick command once in a while, but if you are performing multiple commands in sequence, this will add a great deal of time to each request.
  - If you're just doing a quick "curl" command where latency is not an issue, this method works just fine. Note that some services may see to many of these as a DOS attack, so your mileage may vary.
- Sessions
  - This requires the client POST to the Sessions Collection to get the token and then DELETE the Session when done, so there is a couple of more operations involved.
  - This has the advantage of having all the operations in between being validated against the X-Auth-Token value, reducing the latency of each operation considerably.

Resource Overview

- Session Service: A representation the session service inside of the Redfish Service
  - Contains Attributes of the service:
    - Such as status and timeout policy.
  - Contains a collection of Sessions
    - The ability to see the members of this collection will be account dependent.
- Session: A representation of the “Session Resource”
  - Contains Attributes of the Session:
    - The URI for the Delete operation, like the Location header, is the most important one.

Session Service Model Map





Session Service


```
{
    "@Redfish.Copyright": "Copyright 2014-2016 DMTF. All rights reserved.",
    "@odata.context": "/redfish/v1/$metadata#SessionService.SessionService",
    "@odata.id": "/redfish/v1/SessionService",
    "@odata.type": "#SessionService.v1_0_0.SessionService",
    "Id": "SessionService",
    "Name": "Session Service",
    "Description": "Session Service",
    "Status": {
        "State": "Enabled",
        "Health": "OK"
    },
    "ServiceEnabled": true,
    "SessionTimeout": 30,
    "Sessions": {
        "@odata.id": "/redfish/v1/SessionService/Sessions"
    }
}
```


Timeout & State

Sessions

Session

```
{
    "@Redfish.Copyright":"Copyright 2014-2016 DMTF. All rights reserved.",
    "@odata.context": "/redfish/v1/$metadata#Session.Session",
    "@odata.id": "/redfish/v1/SessionService/Sessions/1234567890ABCDEF",
    "@odata.type": "#Session.v1_0_0.Session",
    "Id": "1234567890ABCDEF",
    "Name": "User Session",
    "Description": "Manager User Session",
    "UserName": "Administrator",
    "Password": "<realpassword",
    "Oem": {}
}
```

URI for DELETE

User Name

Only on POST

How to create a Session

1. GET /redfish/v1
   - This is called the Service Root in the specification.
   - It has the URI of the Session Collection in a property called "SessionCollection"
     - It has other information as well, such the URI for the Systems, Chassis and Managers Collections, Event and Task Service (if any) and other services locations. It also has information about the Redfish Service itself such as versioning information, query options support and more.
   - If this is a 1.6 or later Redfish Service, the Session Collection will be at "/redfish/v1/SessionService/Sessions"
   - Note: It is best practice to perform a GET on /redfish to ensure that there is a /v1 version of the service before doing a GET on /redfish/v

How do you create a Session (continued)

2. POST to the Sessions Collection
   - Technically, you should traverse to the schema/metadata resource like you would for anything else to find the appropriate definition.
   - But the properties for Session establishment have not changed since 1.0 nor are they likely to.
   - The properties that need to be filled in are UserName and Password.
   - So the JSON Body of the Request would look like:

```
{
    "UserName": "<your username>",
    "Password": "<your password>"
}
```

   - The operation would be a POST to /redfish/v1/SessionService/Sessions
   - And you would need all of the necessary headers
     - Older implementations will require the OData-Version header set to 4.0
     - It is advisable to set Content-Type to application/json;charset=utf-8 since services are allowed to reject operations that are not specific.

How do you create a Session (continued)

3. Process the Response Body and headers
   - There are two headers that are very important: Location and X-Auth-Token
     - The Location header will be the URI of your Session Resource. It will be the member of the Sessions Collection. You use the URI to logoff by performing a DELETE operation on the URI.
     - The X-Auth-Token header is the header the Client uses on all subsequent operations (including the DELETE) to indicate the authorization of the Client to perform the operation.
   - There may be a JSON Body that comes back, depending if you get a 200 or a 204

- Include The X-Auth-Token on all subsequent operations
- Perform a DELETE operation on the URI from the Location header in order to log out of the session.
  - Most implementations will have a limit to the number of concurrent sessions and this number will be very small in minimal implementations.

Thank you for watching!

- Redfish Standards
  - Schemas, Specs, Mockups, White Papers, FAQ, Educational Material & more
  - http://www.dmtf.org/standards/redfish
- Redfish Developer Hub
  - Redfish Interactive Explorer, Hosted Schema at Namespace & other links
  - http://redfish.dmtf.org
- Redfish Forum
  (DMTF WG that defines Redfish)
  - Companies involved, Upcoming Schedules & Future work, Charter, Information on joining.
  - http://www.dmtf.org/standards/spmf