Redfish
API Reference Guide v1.0

DELL EMC

Notes, cautions, and warnings

NOTE: A NOTE indicates important information that helps you make better use of your computer.

CAUTION: A CAUTION indicates either potential damage to hardware or loss of data and tells you how to avoid the problem.

WARNING: A WARNING indicates a potential for property damage, personal injury, or death.

© **2016 Dell Inc. All rights reserved.** This product is protected by U.S. and international copyright and intellectual property laws. Dell and the Dell logo are trademarks of Dell Inc. in the United States and/or other jurisdictions. All other marks and names mentioned herein may be trademarks of their respective companies.

2016 - 03

Rev. A00

Contents

Overview

The Redfish Scalable Platforms Management API is a standard defined by the Distributed Management Task Force (DMTF). Redfish is a next-generation systems management interface standard, which enables scalable, secure, and open server management. It is a new interface that uses RESTful interface semantics to access data that is defined in model format to perform out-of-band systems management. It is suitable for a wide range of servers ranging from stand-alone servers to rack mount and bladed environments and for large scale cloud environments.

Dell PowerEdge servers offer extensive and powerful management tools that are included in the Integrated Dell Remote Access Controller (iDRAC) with Lifecycle Controller. Dell has recently extended its support for the DMTF Redfish standard within the iDRAC with Lifecycle Controller firmware version 2.30.30.30.

iDRAC with Lifecycle Controller technology is part of a larger data center solution that helps keep business critical applications and workloads available always. The technology allows administrators to deploy, monitor, manage, configure, update, troubleshoot, and remediate Dell servers from any location, and without the use of agents. It accomplishes this regardless of an operating system or a Hypervisor presence or state.

This document provides a brief overview on Redfish and information on various aspects of Redfish protocol, supported schema, and Redfish Eventing implemented in iDRAC. It also provides guidelines for using the Dell Redfish APIs.

Benefits

Redfish is a new global standard for open server management. It has the capabilities to support single servers, converged infrastructure, and hyper—scale architecture. It provides the following benefits over existing server management methods:

- Increased simplicity and usability
- High data security
- Programmable interface that can be easily scripted
- Follows widely-used standards

Key Technologies

Redfish uses web and cloud-based technologies that enable communications with servers using common programming and scripting languages such as Python, JAVA, and C. The key technologies are as follows:

- REpresentational State Transfer (REST) interface — REST is a web based API, which provides a way to interact with a system over a normal web connection. It supports both HTTPS and HTTP.
- Java Script Notation (JSON) — JSON represents data in such a way that it is much easier to read than XML. It also provides the formatting that is required for scripting languages to interface with the data.
- OData — It is important to standardize the data format when implementing a common interface across multiple vendors. OData provides the required framework to ensure that the data structure remains interchangeable between server vendors.

Other documents you may need

In addition to this guide, the following documents available on the DMTF website provide additional information about Redfish.

- For information about Redfish Schema, see http://Redfish.dmtf.org/schemas/DSP8010_1.0.0.zip .
- For information about Redfish API, see the DMTF Redfish white paper available at https://www.dmtf.org/sites/default/files/standards/documents/DSP2044_1.0.0.pdf.

Redfish Based Systems Management

This section provides an overview of Redfish service implemented in the iDRAC firmware. It includes information about the Redfish API, schema, configuration, authentication, authorization, and so on.

URL support

Redfish is a web-based API which implies that resources are accessed using client supplied URLs. URLs are required to identify Redfish resources. The Redfish API uses a simple URL hierarchy which follows a `/redfish/v1/` pattern for all resources. To access a Redfish resource, use the URL pattern https://<iDRAC IP>/redfish/v1/<Resource Path>. For more information on the supported resources, see Redfish Resources. The following are the list of URI patterns that are supported by iDRAC:

- `/redfish` — URL for the Redfish version object.
- `/redfish/v1` — Root URL for version 1 of Redfish services.
- `/redfish/v1/odata` — Redfish services expose an OData service document at this URI. This service document provides a standard format for enumerating the resources that are exposed by the service by enabling all generic hypermedia-driven OData clients to navigate to the resources of the service.
- `/redfish/v1/$metadata` — Redfish services expose a metadata document in XML format. This document describes the resources and collections that are available at the service root URI. It also provides references to other metadata documents, which describe the complete set of resource types that are exposed by the service.
- `/redfish/v1/$metadata#<Collection or a Singleton resource>` — Metadata URL specified as a part of @odata.context property for all resources. This URL returns data in XML format.
- `/redfish/v1/JsonSchemas` — This URL returns data in JSON format. This returns a collection of JsonSchemaFile resource instances.
- `/redfish/v1/JsonSchemas/<resource URI>` — The JSON Schema File resource instance describes the location (URI) of a particular Redfish schema definition being implemented or referenced by a Redfish service. This URL returns data in JSON format.
- `/redfish/v1/<other resource specific URIs>` — All instrumentation resources follow this pattern.

NOTE: iDRAC's implementation of Redfish supports only HTTPs protocol.

Redfish Configuration

You can configure the Redfish interface on iDRAC by enabling or disabling the iDRAC attribute. If this attribute is disabled, HTTP requests to Redfish URIs will fail with a HTTP status code of 404 and an error message indicating that this attribute is currently disabled.

NOTE: You do not need to restart the web server when enabling or disabling Redfish attribute.

Configuring Redfish service by using iDRAC web interface

To enable or disable the Redfish service on iDRAC, perform the following tasks:

1. In iDRAC web interface, click **Overview** → **iDRAC Settings** → **Network** → **Services**. The **Services** page is displayed.
2. Under **Redfish**, select the **Enabled** check box to enable the service. To disable the service, clear the check-box.
3. Click **Apply** to apply the changes.

Configuring Redfish service by using iDRAC RACADM

You can enable or disable the Redfish service using the iDRAC attribute `iDRAC.Redfish.Enable (Read or Write)`.

Configuring Redfish service by using WS-MAN

The Redfish attribute `iDRAC.Redfish.Enable` is modeled under the existing `DCIM_iDRACCardEnumeration` class. You can configure the Redfish service using existing methods such as SetAttribute, SetAttributes, and ApplyAttributes of `DCIM_iDRACCardService` class.

Redfish Schema

This is the Schema definitions for Redfish resources. It is defined according to OData Schema representation that can be directly translated to a JSON Schema representation.

Redfish Authentication and Authorization

For certain resources, Redfish clients may require to authenticate access. Redfish relies on the managed system for the required credentials and supported forms of authentication. In iDRAC, authentication is based on local credentials and remote protocols such as Active Directory and LDAP.

NOTE: You must have the required iDRAC license to use Active Directory and LDAP.

Authorization includes both user privilege and license authorization. The iDRAC Redfish support is included in all levels of iDRAC licensing. The following table details the authentication and authorization required for each iDRAC Redfish action:

Redfish Actions	Authentication Required	Authorization Required
Read operation on any instrumentation data	Yes	Yes
Modify instrumentation data	Yes	Yes
Invoke actions	Yes	Yes
View Service root	No	No
View Metadata document	No	No
View OData Service Document	No	No
View Message Registry	No	No
View Redfish Version URI	No	No
View JSONSchmeaFile resource URI	No	No
View JSON schemas URI	No	No

The Redfish service provides access to Redfish URLs by using the following methods:

- **Basic authentication:** In this method, user name and password are provided for each Redfish API request.
- **Session based authentication**: This method is used while issuing multiple Redfish operation requests.
  - Session login is initiated by accessing the Create session URI. The response for this request includes an X-Auth-Token header with a session token. Authentication for subsequent requests is made using the X-Auth-Token header.
  - Session logout is performed by issuing a DELETE of the Session resource provided by the Login operation including the X-Auth-Token header.

NOTE: The iDRAC firmware incorporates the concept of application sessions for various existing interfaces such as the GUI, WSMAN, and RACADM. With the introduction of Redfish-specific sessions, Redfish inherits the characteristics of web server sessions and the property Session Timeout inherits the web server session timeout value.

iDRAC Licensing

iDRAC Redfish support is included in all levels of iDRAC licensing. However, some of the iDRAC features require specific licenses. If a required license is not present, certain Redfish APIs are not accessible and return a HTTP 403 status code. In other cases, some of the properties in certain resource may not be returned in a response. The service may also return errors when such properties are modified. For information of specific license requirements for the resources, see Redfish Resources.

HTTP Methods

The REST API allows you to specify the type of request. It adheres to the Create, Retrieve, Update, and Delete (CRUD) standard format. The data is generated via access to URIs, which can be accessed by using the following HTTP methods:

- GET
- POST
- PUT
- PATCH
- DELETE

GET

Use the GET method to retrieve a representation of a resource. The representation can either be a single resource or a collection. The service returns the resource representation by using one of the media types specified in the Accept header depending on the media type requirement. If the Accept header is not present, the service returns the resource representations in the expected format either as application/json or application/xml. The formats are supported by that resource as per the Redfish standard.

The HTTP GET method is used to retrieve a resource without causing any side effects. The service ignores the content of the body on a GET. The GET operation is unchanged in the absence of external changes to the resource.

POST

Use the POST method to invoke actions and create a new resource. The POST request is submitted to the resource collection in which the new resource belongs. Submitting a POST request to a resource representing a collection is equivalent to submitting the same request to the Members property of that resource. Services that support adding members to a collection support both forms.

Services support the POST method for creating resources. If the resource does not support this, status code 405 is returned. The body of the create request contains a representation of the object to be created. The service can ignore any service controlled attributes such as ID, forcing those attributes for the service to be overridden. The service sets the Location header to the URI of the newly created resource. The response to a successful create request is status code 201, which indicates the new resource has been created and includes a response body containing the representation of the newly created resource.

PUT

Use the PUT method to replace the property values of a resource completely. Properties omitted from the request body are reset to their default value. Services support the PUT method to replace a resource completely. If a service does not implement this method, status code 405 is returned. Services may return a representation of the resource after any server-side transformations occur in the body of the response. The PUT operation must be unchanged in the absence of external changes to the resource, with the exception that the ETag values may change as a result of this operation.

PATCH

The PATCH method is the preferred method, which is used to perform updates on pre-existing resources. Changes to the resource are sent in the request body. The PATCH request does not change the properties that are not specified in the request body. The response is either empty or a representation of the resource after the update is done or a success code if the operation is done successfully.. The implementation may reject the update operation on certain fields based on its own policies and does not apply any of the requested updates.

DELETE

Use the DELETE method to remove a resource. Services support the DELETE method for resources that can be deleted. If the resource cannot be deleted, status code 405 is returned. Services return a representation of the deleted resource in the response body. Services may return status code 404 or a success code if the resource is deleted successfully.

HTTP Headers

The server response contains only basic information about related resources. Any metadata, which is required to process a request or response is accessed by using HTTP headers. iDRAC supports the following request headers:

Header	Behavior
If-Match	Supported for AccountService requests. Ignored for all other URIs.
If-None-Match	Supported for AccountService and metadata URIs. Ignored for all other URIs.

iDRAC supports the following response headers:

Header	Behavior
Content-Length	Returned on all responses except those having Transfer-Encoding: chunked.
Content-Type	Responses other than OData metadata — application/json;charset=utf-8
	OData responses — application/xml;charset=utf-8
ETag	Supported on AccountService and metadata URIs.
Location	Service sets this header when resources are created or when HTTP requests are redirected to other resources.
Cache-Control	Returned on all responses. Metadata URIs support cached responses. Instrumentation resources are not cacheable.
X-Auth-Token	Used for authentication of user sessions. See “Session based authentication” under Redfish Authentication and Authorization

HTTP Status Codes and Error Messages

HTTP defines status codes that can be returned in response messages. When the HTTP status code indicates a failure, the response body contains an extended error resource, which provides meaningful and deterministic error semantics.

Dell Redfish service extended error information contains error or exception information that is unique to the Dell implementation. It provides additional details and recommendations for error resolution. To learn more about extended error information, see the iDRAC with Lifecycle Controller Dell Event Message Reference white paper available at http://en.community.dell.com/techcenter/extras/m/white_papers/20442267.

For more information about supported status codes, see the Redfish Scalable Platforms Management API Specification document available at https://www.dmtf.org/standards/redfish.

For more information about error messages, see the white papers available at https://www.dmtf.org/standards/redfish.

SSL Certificates of iDRAC

iDRAC includes a web server that is configured to use the industry-standard SSL security protocol to transfer encrypted data over a network. Built upon asymmetric encryption technology, SSL is widely accepted for providing authenticated and encrypted communication between clients and servers to prevent eavesdropping across a network. Redfish service reuses SSL certificate installed on the iDRAC web server. The iDRAC web server has a Dell self-signed unique SSL digital certificate by default. You can replace the default SSL certificate with a certificate signed by a well-known Certificate Authority (CA). SSL certificates can be replaced using the iDRAC interfaces such as web interface, RACADM, or WS-MAN. For more information on managing SSL certificates of iDRAC, see the latest iDRAC User’s Guide available at **dell.com/idracmanuals**.

Eventing

The Redfish service generates asynchronous notifications (events) that are defined by Redfish subscription for the eventing service. These events are sent to an event destination by using HTTP POST method. Events are generated when some significant

change or error condition typically of time critical nature occurs. When an event occurs on the service, it notifies the clients. Redfish service must be enabled and iDRAC must be configured to create event subscriptions and to gain read-only privilege for viewing event subscriptions.

The iDRAC implementation of a Redfish service supports only HTTPS notifications. In certain situations, iDRAC may not be able to verify certificates sent by a peer. To handle such situations, iDRAC can be configured to skip certificate verification by using the attribute `iDRAC.RedfishEventing.IgnoreCertificateErrors`. This attribute can be configured to True or False (Default) using RACADM or the WS-MAN interface. Set this attribute to True if certificate validation is not required.

NOTE: In this release, this attribute is read-only and is set to True.

Redfish service provides Lifecycle and Alert events. Lifecycle events may occur when resources are created, modified, or destroyed. Alert events occur when a resource needs to indicate a significant event. This may be either directly or indirectly pertaining to the resource. Examples of this kind of event are when a chassis is opened, button is pressed, cable is unplugged, or threshold is exceeded. iDRAC supports up to 20 event subscriptions.

NOTE: In this release, iDRAC supports only Alert event notifications.

In case an event delivery fails, the event service of iDRAC retries delivering the failed event. The number of retries and delivery intervals can be configured using the following attributes:

- `iDRAC.RedfishEventing.DeliveryRetryAttempts`
- `iDRAC.RedfishEventing.DeliveryRetryIntervalInSeconds`

Event Delivery Retry Settings in RACADM

Table 1. iDRAC.Redfisheventing.DeliveryRetryAttempts (Read or Write)

Description	Specifies the number of retry attempts made for Redfish event delivery
Legal Values	Value ranges from 0 to 5
Default Value	3
Write Privilege	Configure iDRAC

Table 2. iDRAC.Redfisheventing.DeliveryRetryIntervalInSeconds (Read or Write)

Description	Specifies the intervals (in seconds) of retry attempts made for Redfish event delivery
Legal Values	Value ranges from 5 to 60
Default Value	30
Write Privilege	Configure iDRAC

Eventing Operations

The Redfish event service provides the following URIs:

HTTP Method Type	Description	URI	Metadata Reference
GET	Get Event Service detailed information	/redfish/v1/EventService	EventService.xml
POST	Register an event notification receiver	/redfish/v1/EventService/Subscriptions	EventDestination.xml
DELETE	Remove a subscription	/redfish/v1/EventService/ Subscriptions /<Subscription ID>	EventService.xml

PowerEdge FX2/FX2s Chassis Management using iDRAC Redfish

On a PowerEdge FX2/FX2s chassis, iDRAC can monitor and manage chassis components such as fans, power supplies, and so on. Redfish service provides information about chassis components when **Chassis Management and Monitoring** is set to **Enabled** on iDRAC. This setting allows you to monitor and manage the chassis even if the CMC is not on the network. On an FX2/FX2s CMC, ensure that the **Chassis Management at Server** setting is set to **Monitor** or **Manage and Monitor**. While this feature is enabled, iDRAC also generates Redfish notifications for chassis events.

Configuring Chassis Management and Monitoring using iDRAC web interface

1. In iDRAC web interface, click **Overview** → **iDRAC Settings** → **CMC**.
2. Under **Chassis Management** → **Capability from iDRAC** at the Server section, select the **Enabled** check box. To disable the service, clear the check box.

Configuring Chassis Management and Monitoring using iDRAC RACADM

To configure the chassis management and monitoring using RACADM, type

```
racadm set system.chassiscontrol.chassismanagementmonitoring Enabled
```

For more information on Chassis Management and Monitoring, see the *iDRAC User's Guide* available at **dell.com/idracmanuals**.

Redfish Resources

This section describes the resource URIs and related operations that are available in the iDRAC implementation of a Redfish service API.

AccountService

Description

This resource is used to represent a management account service for a Redfish implementation.

URL

`/redfish/v1/Managers/<ID>/AccountService`

Supported HTTP Methods and Privileges

HTTP Method	Required Privilege
GET	Login

Chassis

Description

This resource is used to represent a chassis or other physical enclosure for a Redfish implementation.

URL

`/redfish/v1/Chassis`

Supported HTTP Methods and Privileges

HTTP Method	Required Privilege
GET	Login

Navigation URL

`/redfish/v1/Chassis/<ID>`

Supported HTTP Methods and Privileges

HTTP Method	Required Privilege
PATCH	ConfigureManager

Supported Status Codes

HTTP Status Code	Extended Information
200	Base.1.0.success
400	Base.1.0.PropertyValueTypeError Base.1.0.PropertyValueNotInList Base.1.0.PropertyUnknown Base.1.0.PropertyNotWritable Base.1.0.PropertyValueFormatError
500	Base.1.0.InternalError

Supported Attributes and Values

Attributes	Values
ResetType	On ForceOff
IndicatorLed	Lit Off
ChassisType	Rack StandAlone Blade Enclosure sled

Supported Action — Reset

URL

`/redfish/v1/Chassis/System.Embedded.1/Actions/Chassis.Reset`

Description

This action is used to reset the chassis.

Supported HTTP Methods and Privileges

HTTP Method	Required Privilege
POST	ConfigureComponent

Updatable Properties

Parameter	Value
ResetType	On ForceOff

Supported Status Codes

HTTP Status Code	Extended Information
204	
400	Base.1.0.PropertyValueTypeError Base.1.0.PropertyValueNotInList Base.1.0.PropertyUnknown

HTTP Status Code	Extended Information
	Base.1.0.PropertyNotWritable Base.1.0.PropertyValueFormatError
500	Base.1.0.InternalError
503	

/redfish/v1/Chassis/<ID>/Sensors/Voltages/<ID>

Description

Sensor information

Supported HTTP Methods and Privileges

HTTP Method	Required Privilege
GET	Login

Supported Status Codes

HTTP Status Code
200
400
500

NOTE: On PowerEdge FX2 systems, iDRAC can display additional instrumentation data from the sensors on the server, only if the Chassis Monitoring setting is set to enabled on iDRAC and CMC.

Contained Resources

- Power
- Thermal

ComputerSystem

Description

This resource is used to represent resources that represent a computing system in the Redfish specification.

URL

```
/redfish/v1/Systems
```

Supported HTTP Methods and Privileges

HTTP Method	Required Privilege
GET	Login

Navigation URL

```
/redfish/v1/Systems/<ID>
```

Supported HTTP Methods and Privileges

HTTP Method	Required Privilege
PATCH	ConfigureManager

Supported Status Codes and Error Messages

HTTP Status Code	Extended Information
200	Base.1.0.success
400	Base.1.0.PropertyValueTypeError Base.1.0.PropertyValueNotInList Base.1.0.PropertyUnknown Base.1.0.PropertyNotWritable Base.1.0.PropertyValueFormatError
500	Base.1.0.InternalError

Supported Attributes and Values

Attributes	Values
ResetType	On ForceOff GracefulRestart PushPowerButton Nmi
PowerState	On Off
SystemType	Physical
BootSource	Pxe Floppy CD Usb Hdd Utilities UefiTarget BiosSetup
BootSourceOverrideEnabled	Disabled Once Continuous
UefiTargetBootSourceOverride	Any valid UEFIDevicepath
IndicatorLed	Lit Off

Implementation Notes

Some of the properties in this schema are dependent on the installed BIOS version. If a compatible BIOS version is not installed, UefiTargetBootSourceOverride property is not supported in this resource.

Supported Action — Reset

Description

Resets computer system.

URL

`/redfish/v1/Systems/<ID>/Actions/ComputerSystem.Reset`

Supported HTTP Methods and Privileges

HTTP Method	Required Privilege
POST	ConfigureComponent

Updatable Properties

Parameter	Value
ResetType	On ForceOff GracefulRestart PushPowerButton NMI

Supported Status Codes

HTTP Status Code	Extended Information
200	
400	Base.1.0.PropertyValueTypeError Base.1.0.PropertyValueNotInList Base.1.0.PropertyUnknown Base.1.0.PropertyNotWritable Base.1.0.PropertyValueFormatError
500	Base.1.0.InternalError

Contained Resources

- Processor
- EthernetInterfaces
- SimpleStorage
- LogService

EthernetInterfaces

Description

This resource is used to represent NIC resources as part of the Redfish specification. It also updates the properties of Manager Ethernet Interface.

URL

`/redfish/v1/Managers/<ID>/EthernetInterfaces`

Supported HTTP Methods and Privileges

HTTP Method	Required Privilege
GET	Login

URL

```
/redfish/v1/Managers/<ManagerInstanceID>/EthernetInterfaces/<EthernetInstanceID>
```

Supported HTTP Methods and Privileges

HTTP Method	Required Privilege
PATCH	ConfigureManager

Updatable Properties

Parameter	Description
HostName	updates HostName
IPv4	updates IPv4
IPv6	updates IPv6
IPv6Static	updates IPv6Static

Supported Status Codes and Error Message Codes

HTTP Status Code	Extended Information	Error Message Code
200	Base.1.0.success	
400	Base.1.0.PropertyValueTypeError Base.1.0.PropertyValueNotInList Base.1.0.PropertyUnknown Base.1.0.PropertyNotWritable Base.1.0.PropertyValueFormatError	ISM0013 RAC0253 RAC0254 RAC0255 RAC0259 SWC0296
500	Base.1.0.InternalError	

Reference Properties

/Systems/<ServiceTag+nodeid>/EthernetInterfaces

Description

This resource is used to represent NIC resources as part of the Redfish specification.

Supported HTTP Methods and Privileges

HTTP Method	Required Privilege
GET	Login

Supported Status Codes

HTTP Status Code
200
400
500

Implementation Notes

Some of the properties in this schema are dependent on the installed BIOS and iSM version. If a compatible version of BIOS is not installed, UefiDevicePath is not supported in this resource. If a compatible version of iSM is not installed, certain properties may not be supported.

EventService

Description

It represents the properties for the service itself and has links to the actual list of subscriptions.

URL

`/redfish/v1/EventService`

Supported HTTP Methods and Privileges

HTTP Method	Required Privilege
GET	Login

Supported Status Codes

HTTP Status Code
200
400
500

Supported Action — SubmitTestEvent

URL

`/redfish/v1/EventService/Actions/EventService.SubmitTestEvent`

Description

Defines the name of the custom action supported when used in conjunction with a POST operation to this resource. When issued, this operation will perform a reset of the manager.

Supported HTTP Methods and Privileges

HTTP Method	Required Privilege
POST	ConfigureManager

Updatable Properties

Parameter	Value
MessageId	<Subscription-id>

Supported Status Codes

HTTP Status Code	Extended Information
204	
400	Base.1.0.PropertyValueTypeError Base.1.0.PropertyValueNotInList Base.1.0.PropertyUnknown Base.1.0.PropertyNotWritable Base.1.0.PropertyValueFormatError
500	Base.1.0.InternalError

Contained Resources

EventDestination

EventDestination

Description

This property contain an URI to the destination where the events are sent.

URL

`/redfish/v1/EventService/Subscriptions`

Supported HTTP Methods and Privileges

HTTP Method	Required Privilege
GET	Login
POST	ConfigureManager
DELETE	ConfigureManager

Updatable Properties

Parameter	Description
Destination	Destination IP to send event
EventTypes	Contains types of events
Context	Client supplied String
Protocol	Protocol type used by event

Supported Status Codes

HTTP Status Code	Extended Information
200	
201	
400	Base.1.0.PropertyValueTypeError Base.1.0.PropertyValueNotInList Base.1.0.PropertyUnknown Base.1.0.PropertyNotWritable Base.1.0.PropertyValueFormatError
500	Base.1.0.InternalError
503	

JSONSchemas

Description

This resource is used for representing the Schema File locator resource for a Redfish implementation.

URL

`/redfish/v1/JSONSchemas`

Supported HTTP Methods and Privileges

HTTP Method	Required Privilege
GET	Login

Supported Status Codes

HTTP Status Code
200
400
500

LogEntry

Description

This resource represents the log format for log services in a Redfish implementation.

URL

`/redfish/v1/Managers/<ID>/Logs`

Supported HTTP Methods and Privileges

HTTP Method	Required Privilege
GET	Login

Supported Status Codes

HTTP Status Code
200
400
500

Reference Properties

/redfish/v1/Managers/<ID>/Logs/Lclog

Description

This resource represents the Lifecycle Controller logs for the manager in a Redfish implementation.

Supported HTTP Methods and Privileges

HTTP Method	Required Privilege
GET	Login

Supported Status Codes

HTTP Status Code	Extended Information
200	
400	
500	

/redfish/v1/Managers/<ID>/Logs/Sel

Description

This resource represents the System Event logs for the manager in a Redfish implementation.

Supported HTTP Methods and Privileges

HTTP Method	Required Privilege
GET	Login

Supported Status Codes

HTTP Status Code	Extended Information
200	
400	
500	

LogService

Description

This resource is used to represent a log service for a Redfish implementation.

URL

`/redfish/v1/Managers/<ID>/LogService`

Supported HTTP Methods and Privileges

HTTP Method	Required Privilege
GET	Login

Supported Status Codes

HTTP Status Code
200
400
500

Reference Properties

/redfish/v1/Managers/<ID>/LogServices/Lclog

Description

This resource represents the Lifecycle Controller log service in a Redfish implementation.

Supported HTTP Methods and Privileges

HTTP Method	Required Privilege
GET	Login

Supported Status Codes

HTTP Status Code
200
400
500

/redfish/v1/Managers/<ID>/LogServices/Sel

Description

This resource represents the SEL log service in a Redfish implementation.

Supported HTTP Methods and Privileges

HTTP Method	Required Privilege
GET	Login

Supported Status Codes

HTTP Status Code
200
400
500

Supported Action — ClearLog

URL

`/redfish/v1/Managers/<ID>/LogServices/Sel/Actions/LogService.ClearLog`

Description

Performs clear operation on logs.

Supported HTTP Methods and Privileges

HTTP Method	Required Privilege
POST	ConfigureManager

Updatable Properties

Parameter	Description
ClearLog	ClearLog

Supported Status Codes

HTTP Status Code	Extended Information
204	
400	Base.1.0.PropertyValueTypeError Base.1.0.PropertyValueNotInList Base.1.0.PropertyUnknown Base.1.0.PropertyNotWritable Base.1.0.PropertyValueFormatError
500	Base.1.0.InternalError

Manager

Description

This resource is used to represent a management sub-system for a Redfish implementation.

URL

`/redfish/v1/Managers`

Supported HTTP Methods and Privileges

HTTP Method	Required Privilege
GET	Login

Supported Status Codes

HTTP Status Code
200
400
500

Supported Attributes and Values

Attributes	Values
ManagerType	BMC
CommandConnectTypesSupported	SSH Telnet IPMI
GraphicalConnectTypesSupported	KVMIP
ResetType	GracefulRestart

Supported Action — Reset

Description

This defines the name of the custom action supported when used in conjunction with a POST operation to this resource. When issued, this operation will perform a reset of the manager.

URL

`/redfish/v1/Managers/<ID>/Actions/Manager.Reset`

Supported HTTP Methods and Privileges

HTTP Method	Required Privilege
POST	ConfigureManager

Updatable Properties

Parameter	Value
ResetType	GracefulRestart

HTTP Status Code	Extended Information
204	
400	Base.1.0.PropertyValueTypeError Base.1.0.PropertyValueNotInList Base.1.0.PropertyUnknown Base.1.0.PropertyNotWritable Base.1.0.PropertyValueFormatError
500	Base.1.0.InternalError

ManagerAccount

Description

This resource represents the BMC user accounts collection for a Redfish implementation.

URL

`/redfish/v1/Managers/<ID>/Accounts`

Supported HTTP Methods and Privileges

HTTP Method	Required Privilege
GET	Login

Navigation URL

`/redfish/v1/Managers/<ID>/Accounts/<Account-id>`

Supported HTTP Methods and Privileges

HTTP Method	Required Privilege
PATCH	ConfigureManager

Supported Status Codes

HTTP Status Code	Error Message Code
200	
400	RAC0288 RAC0291
404	
500	

Contained Resources

Role

ManagerNetworkProtocol

Description

This object is used to represent the network service settings for the manager.

URL

`/redfish/v1/Managers/<ID>/NetworkProtocol`

Supported HTTP Methods and Privileges

HTTP Method	Required Privilege
GET	Login
PATCH	ConfigureManager

Updatable Properties

Parameter	Description
FQDN	updates FQDN
hostname	updates hostname
HTTP	updates HTTP
HTTPS	updates HTTPS
IPMI	updates IPMI
KVMIP	updates KVMIP
SNMP	updates SNMP
SSH	updates SSH
Telnet	updates Telnet
VirtualMedia	updates VirtualMedia

Supported Status Codes and Error Messages

HTTP Status Code	Extended Information
200	Base.1.0.success
400	Base.1.0.PropertyValueTypeError Base.1.0.PropertyValueNotInList Base.1.0.PropertyUnknown Base.1.0.PropertyNotWritable Base.1.0.PropertyValueFormatError
403	
500	Base.1.0.InternalError

MessageRegistry

Description

This resource is used to represent a message registry for a Redfish implementation.

URL

```
/redfish/v1/Registries/Messages/En
```

Supported HTTP Methods and Privileges

HTTP Method	Required Privilege
GET	Login

Supported Status Codes

HTTP Status Code
200
400
500

Power

Description

This resource is used to represent a power metrics resource for a Redfish implementation.

URL

`/redfish/v1/Chassis/<ID>/Power`

Supported HTTP Methods and Privileges

HTTP Method	Required Privilege
GET	Login
PATCH	ConfigureManager

Supported Status Codes

HTTP Status Code	Extended Information
200	Base.1.0.Success
400	Base.1.0.PropertyValueTypeError Base.1.0.PropertyValueNotInList Base.1.0.PropertyUnknown Base.1.0.PropertyNotWritable Base.1.0.PropertyValueFormatError
403	
500	

Reference Properties

/redfish/v1/Chassis/<ID>/Power/PowerControl

Description

Updates the properties of PowerControl in Chassis Collection.

Supported HTTP Methods and Privileges

HTTP Method	Required Privilege
GET	Login
PATCH	ConfigureManager

Updatable Properties

Parameter	Description
PowerLimit	updates PowerLimit

Supported Status Codes

HTTP Status Code	Extended Information
200	Base.1.0.success
400	Base.1.0.PropertyValueTypeError Base.1.0.PropertyValueNotInList Base.1.0.PropertyUnknown Base.1.0.PropertyNotWritable Base.1.0.PropertyValueFormatError
500	Base.1.0.InternalError

/redfish/v1/Chassis/<ID>/Power/PowerSupplies/<ID>

Description

Provides details of the power supplies that are associated with the system or device.

Supported HTTP Methods and Privileges

HTTP Method	Required Privilege
GET	Login

Supported Status Codes

HTTP Status Code
200
400
403
500

/redfish/v1/Chassis/<ID>/Sensors/Voltages/<ID>

Description

Sensor information

Supported HTTP Methods and Privileges

HTTP Method	Required Privilege
GET	Login

Supported Status Codes

HTTP Status Code
200
400
500

NOTE: On PowerEdge FX2 systems, iDRAC can display additional instrumentation data from the sensors on the server, only if the Chassis Monitoring setting is set to enabled on iDRAC and CMC.

/redfish/v1/Chassis/<ID>/Power/Redundancy/<ID>

Description

This object represents the Redundancy element property.

Supported HTTP Methods and Privileges

HTTP Method	Required Privilege
GET	Login

Supported Status Codes

HTTP Status Code
200
400
500

Processor

Description

This is the schema definition for the Processor resource. It represents the properties of a processor attached to a system.

URL

`/redfish/v1/Systems/<ID>/Processors`

Supported HTTP Methods and Privileges

HTTP Method	Required Privilege
GET	Login

Supported Status Codes

HTTP Status Code
200
400
500

Supported Attributes and Values

Attributes	Values
ProcessorType	CPU
ProcessorArchitecture	x86
InstructionSet	x86-64

Implementation Notes

Some of the properties in this schema are dependent on the installed BIOS version. If a compatible BIOS version is not installed, some of the properties may not be supported on this resource.

Role

Description

This resource is used to represent resources that represent the user role for the user account.

URL

```
/redfish/v1/Managers/<ID>/Roles
```

Supported HTTP Methods and Privileges

HTTP Method	Required Privilege
GET	Login

Supported Status Codes

HTTP Status Code
200
400
500

SerialInterfaces

Description

This resource is used to represent serial resources as part of the Redfish specification.

URL

```
/redfish/v1/Managers/<ID>/SerialInterfaces
```

Supported HTTP Methods and Privileges

HTTP Method	Required Privilege
GET	Login

Navigation URL

```
/redfish/v1/Managers/<ID>/SerialInterfaces/<Serial-key>
```

Supported HTTP Methods and Privileges

HTTP Method	Required Privilege
PATCH	ConfigureManager

Updatable Properties

Parameter	Description
BitRate	updates BitRate
InterfaceEnabled	updates InterfaceEnabled

Supported Status Codes

HTTP Status Code	Extended Information
200	Base.1.0.success
400	Base.1.0.PropertyValueTypeError Base.1.0.PropertyValueNotInList Base.1.0.PropertyUnknown Base.1.0.PropertyNotWritable Base.1.0.PropertyValueFormatError
500	Base.1.0.InternalError

Supported Attributes and Values

Attributes	Values
SignalType	Rs232
BitRate	9600 19200 38400 57600 115200
Parity	None
DataBits	8
StopBits	1
FlowControl	Hardware
PinOut	Cisco
ConnectorType	DB9 Male

ServiceRoot

Description

This object represents the root Redfish service. All values for resources described by this schema must comply with the requirements described in the Redfish specification.

URL

```
/redfish/v1
```

Supported HTTP Methods and Privileges

HTTP Method	Required Privilege
GET	Login

Supported Status Codes

HTTP Status Code
200
400
403
500

Session

URL

`/redfish/v1/Sessions`

Description

This resource is used to represent a session for a Redfish implementation.

Supported HTTP Methods and Privileges

HTTP Method	Required Privilege
GET	Login
POST	ConfigureManager

Updatable Properties

Parameter	Description
username	username
password	password

Supported Status Codes

HTTP Status Code	Extended Information
200	
400	Base.1.0.PropertyValueTypeError Base.1.0.PropertyValueNotInList Base.1.0.PropertyUnknown Base.1.0.PropertyNotWritable Base.1.0.PropertyValueFormatError
500	Base.1.0.InternalError

/redfish/v1/Sessions/<session-id>

Description

Performs delete operation on session-id.

Supported HTTP Methods and Privileges

HTTP Method	Required Privilege
DELETE	Login

Supported Status Codes

HTTP Status Code
200
400
500

SessionService

Description

This resource is used to represent the Session Service properties for a Redfish implementation.

URL

`/redfish/v1/SessionService`

Supported HTTP Methods and Privileges

HTTP Method	Required Privilege
GET	Login
PATCH	ConfigureManager

NOTE: The properties for this resource are inherited from the web-server properties.

Supported Status Codes

HTTP Status Code	Extended Information
200	Base.1.0.success
400	Base.1.0.PropertyValueTypeError Base.1.0.PropertyValueNotInList Base.1.0.PropertyUnknown Base.1.0.PropertyNotWritable Base.1.0.PropertyValueFormatError
500	Base.1.0.InternalError

SimpleStorage

Description

This property contains the UEFI device path used to identify and locate a specific storage controller.

URL

`/redfish/v1/Systems/<ID>/Storage/Controllers`

Supported HTTP Methods and Privileges

HTTP Method	Required Privilege
GET	Login

Supported Status Codes

HTTP Status Code
200
400
403
500

TaskService

Description

This resource represents a task service for a Redfish implementation.

URL

`/redfish/v1/TaskService`

Supported HTTP Methods and Privileges

HTTP Method	Required Privilege
GET	Login

Supported Status Codes

HTTP Status Code
200
400
500

Thermal

Description

This resource is used to represent the thermal metrics resource for a Redfish implementation.

URL

`/redfish/v1/Chassis/<ID>/Thermal`

Supported HTTP Methods and Privileges

HTTP Method	Required Privilege
GET	Login

Supported Status Codes

HTTP Status Code
200
400
500

Reference Properties

/redfish/v1/Chassis/<ID>/Sensors/Fans/<ID>

Description

Provides details of the fan that is associated with the system or chassis.

Supported HTTP Methods and Privileges

HTTP Method	Required Privilege
GET	Login

Supported Status Codes

HTTP Status Code
200
400
500

NOTE: On PowerEdge FX2 systems, iDRAC can display additional instrumentation data from the sensors on the server, only if the Chassis Monitoring setting is set to enabled on iDRAC and CMC.

/redfish/v1/Chassis/<ID>/Sensors/Temperatures/<ID>

Description

Represents the properties for temperature sensors.

Supported HTTP Methods and Privileges

HTTP Method	Required Privilege
GET	Login

Supported Status Codes and Error Messages

HTTP Status Code
200
400
500

NOTE: On PowerEdge FX2 systems, iDRAC can display additional instrumentation data from the sensors on the server, only if the Chassis Monitoring setting is set to enabled on iDRAC and CMC.

/redfish/v1/Chassis/<ID>/Thermal/Redundancy/<ID>

Description

Provides redundant information that is available for fans and other elements in this resource.

Supported HTTP Methods and Privileges

HTTP Method	Required Privilege
GET	Login

Supported Status Codes

HTTP Status Code
200
400
403
500

VirtualMedia

Description

This resource is used to represent a virtual media service for a Redfish implementation.

URL

```
/redfish/v1/Managers/<ID>/VirtualMedia
```

Supported HTTP Methods and Privileges

HTTP Method	Required Privilege
GET	Login

Supported Status Codes

HTTP Status Code
200
400
403
500

Supported Attributes and Values

Attributes	Values
MediaTypes	CD
	DVD
	USBStick
ConnectedVia	NotConnected
	Applet

VLanNetworkInterface

Description

The value of this property indicates if VLAN is enabled for this interface.

URL

`/redfish/v1/Systems/<ID>/EthernetInterfaces/<EthernetInstanceID>/Vlans`

Supported HTTP Methods and Response Content Type

HTTP Method	Response Content Type
GET	application/json

Supported Status Codes

HTTP Status Code
200
400
500

Examples

The following table provides usage examples for the HTTP supported methods such as GET, POST, PATCH, and DELETE:

GET

Table 3. GET Usage Examples

URL	`/redfish/v1/Managers/iDRAC.Embedded.1/SerialInterfaces`

Output	`{`
	`"@odata.context": "/redfish/v1/$metadata#Managers/Members/iDRAC.Embedded.1/SerialInterfaces/$entity",`
	`"@odata.count": 1,`
	`"@odata.id": "/redfish/v1/Managers/iDRAC.Embedded.1/SerialInterfaces",`
	`"@odata.type": "#SerialInterface.1.0.0.SerialInterfaceCollection",`
	`"Description": "Collection of Serial Interfaces for this System",`
	`"Members": [`
	`{`
	`"@odata.id": "/redfish/v1/Managers/iDRAC.Embedded.1/SerialInterfaces/iDRAC.Embedded.1#Serial.1"`
	`}`
	`],`
	`"Name": "Serial Interface Collection"`
	`}`

DELETE

Table 4. DELETE Usage Examples

URL	`/redfish/v1/EventService/Subscriptions/<SubscriptionId>`

Output	`200 Ok`
	`{`
	`"INFO": "<SubscriptionId> subscription deleted successfully"`
	`}`

PATCH

Table 5. PATCH Usage Examples

URL	`/redfish/v1/Managers/iDRAC.Embedded.1/Accounts/<Account-id>`
Input	`{"Password": "123","UserName": "reader"}`

Output	`{`
	`"Success": {`
	`"Message": "Successfully Completed Request",`
	`"MessageId": "Base.1.0.Success",`
	`"Resolution": "None",`
	`"Severity": "Ok"`
	`}`
	`}`

POST

Table 6. POST Usage Examples

URL	`/redfish/v1/Systems/System.Embedded.1/Actions/ComputerSystem.Reset`
Input	`{"ResetType":"GracefulRestart"}`
Output	`204: No Content`