

[UEFI Practice] Redfish BIOS Implementation - Generate EDK Data

原创

jiangwei0512

Posted on 2024-01-28 13:24:45

Read 1.6k

Collection 11

Likes 9

Category Column:

UEFI Development Basics

Article Tags:

uefi

redfish

Copyright CC 4.0 BY-SA

UEFI Development ...

This column includes this content

136 articles

Subscribe to our column

Generate Redfish file


The most commonly used representation format for Redfish data is JSON. Converting the data represented by JSON into a structure that can be operated by C language is an essential step. Of course, manual conversion will waste a lot of time, so the DMTF organization has developed a tool to quickly convert JSON data into C language modules.

The tool is located at <https://github.com/DMTF/Redfish-Schema-C-Struct-Generator>. The downloaded content is as follows:


名称	修改日期	类型	大小
 _Edk2OpenSourceTempFiles	2024/1/28 11:46	文件夹	
 AUTHORS.md	2024/1/28 11:46	Markdown File	1 KB
 CHANGELOG.md	2024/1/28 11:46	Markdown File	1 KB
 GenCRelatedFile.py	2024/1/28 11:46	Python File	112 KB
 GenEdk2Files.py	2024/1/28 11:46	Python File	24 KB
 GenMarkdown.py	2024/1/28 11:46	Python File	19 KB
 GenRedfishSchemaCS.py	2024/1/28 11:52	Python File	52 KB
 LICENSE.md	2024/1/28 11:46	Markdown File	4 KB
 README.md	2024/1/28 11:46	Markdown File	21 KB
 RedfishCSDef.py	2024/1/28 11:46	Python File	3 KB
 ToolLogger.py	2024/1/28 11:46	Python File	4 KB

You can see that it is a series of Python scripts, and its entry is the GenRedfishSchemaCS.py file.

To use this tool to convert JSON files, you first need JSON files supported by Redfish. You can download all JSON data files defined by Redfish at [Redfish Schema Index](#) | [Redfish™ Developer Hub \(dmtf.org\)](#) :



DMTF
REDFISH® DEVELOPERS HUB



HomeMockupsEducationDeveloper EssentialsCommunityAbout

Redfish Schema Index

Primary Schema Files

The following files are the primary schemas defined for JSON resources conforming to the Redfish Specification. Each schema definition is available in both CSDL (XML) and json-schema formats. A .ZIP archive (DSP8010) containing all of current schema files (both CSDL and json-schema) can be downloaded from the Redfish Standards page at: <http://www.dmtf.org/standards/redfish>

AccelerationFunction [csdl] [json-schema] [yaml]	v1.0.4	The AccelerationFunction schema describes an acceleration function that a processor implements. This can include functions such as audio processing, compression, encryption, packet inspection, packet switching, scheduling, or video processing.
AccountService [csdl] [json-schema] [yaml]	v1.15.0	The AccountService schema defines an account service. The properties are common to, and enable management of, all user accounts. The properties include the password requirements and control features, such as account lockout. Properties and actions in this service specify general behavior that should be followed for typical accounts, however implementations might override these behaviors for special accounts or situations to avoid denial of service or other deadlock situations.
ActionInfo [csdl] [json-schema] [yaml]	v1.4.1	The ActionInfo schema defines the supported parameters and other information for a Redfish action. Supported parameters can differ among vendors and even among resource instances. This data can ensure that action requests from applications contain supported parameters.
AddressPool [csdl] [json-schema] [yaml]	v1.2.4	The schema definition of an address pool and its configuration.
Aggregate [csdl] [json-schema] [yaml]	v1.0.2	The Aggregate schema describes a grouping method for an aggregation service. Aggregates are formal groups of resources that are more persistent than ad hoc groupings.
AggregationService [csdl] [json-schema] [yaml]	v1.0.2	The AggregationService schema contains properties for managing aggregation operations, either on ad hoc combinations of resources or on defined sets of resources called aggregates. Access points define the properties needed to access the entity being aggregated and connection methods describe the protocol or other semantics of the connection.
AggregationSource [csdl] [json-schema] [yaml]	v1.4.0	The AggregationSource schema is used to represent the source of information for a subset of the resources provided by a Redfish service. It can be thought of as a provider of information. As such, most such interfaces have requirements to support the gathering of information like address and account used to access the information.

You can see that not only JSON is supported, but also CSDL and YAML, but the current tool only supports JSON conversion. Here is a JSON data supported by Redfish, such as BootOption related data:

```
1 {
2   "$id": "http://redfish.dmtf.org/schemas/v1/BootOption.v1_0_5.json",
3   "$ref": "#/definitions/BootOption",
4   "$schema": "http://redfish.dmtf.org/schemas/v1/redfish-schema-v1.json",
5   "copyright": "Copyright 2014-2023 DMTF. For the full DMTF copyright policy, see http://www.dmtf.org/about/policies/copyright",
6   "definitions": {
7     "Actions": {
8       "additionalProperties": false,
9       "description": "The available actions for this resource.",
10      "longDescription": "This type shall contain the available actions for this resource.",
11      "patternProperties": {
12        "([a-zA-Z][a-zA-Z0-9_]*)?@(odata|Redfish|Message)\\.([a-zA-Z][a-zA-Z0-9_]*)$": {
13          "description": "This property shall specify a valid odata or Redfish property.",
14          "type": [
15            "array",
16            "boolean",
17            "integer",
18            "number",
19            "null",
20            "object",
21            "string"
22          ]
23        }
24      },
25      "properties": {
26        "Oem": {
27          "$ref": "#/definitions/OemActions",
28          "description": "The available OEM-specific actions for this resource.",
29          "longDescription": "This property shall contain the available OEM-specific actions for this resource."
30        }
31      },
32      "type": "object"
33    },
34    "BootOption": {
35      "additionalProperties": false,
36      "description": "The BootOption schema reports information about a single boot option in a system. It represents the properties of a bootable device available in the
37      system.",
38      "longDescription": "This resource shall represent a single boot option within a system.",
39      "patternProperties": {
40        "([a-zA-Z][a-zA-Z0-9_]*)?@(odata|Redfish|Message)\\.([a-zA-Z][a-zA-Z0-9_]*)$": {
41          "description": "This property shall specify a valid odata or Redfish property.",
42          "type": [
43            "array",
44            "boolean",
45            "integer",
46            "number",
47            "null",
48            "object",
49            "string"
50          ]
51        }
52      },
53      "properties": {
54        "@odata.context": {
55          "$ref": "http://redfish.dmtf.org/schemas/v1/odata-v4.json#/definitions/context"
56        },
57        "@odata.etag": {
58          "$ref": "http://redfish.dmtf.org/schemas/v1/odata-v4.json#/definitions/etag"
59        },
60        "@odata.id": {
61          "$ref": "http://redfish.dmtf.org/schemas/v1/odata-v4.json#/definitions/id"
62        },
63        "@odata.type": {
64          "$ref": "http://redfish.dmtf.org/schemas/v1/odata-v4.json#/definitions/type"
65        },
66        "Actions": {
67          "$ref": "#/definitions/Actions",
68          "description": "The available actions for this resource.",
69          "longDescription": "This property shall contain the available actions for this resource."
70        }
71      },
72      "type": "object"
73    }
74  }
75 }
```

Download it and put it into a JSON file, then put the file into the Redfish-Schema-C-Struct-Generator directory and use the following command to convert it:

```
bash
1 GenRedfishSchemaCS.py -edk2 -file=BootOption.v1_0_5.json
```

AI generated projects 登录复制

The result is:

Edk2OpenSourceTempFiles2024/11/28 11:46文件夹

AUTHORS.md2024/11/28 11:46Markdown File1 KB

CHANGELOG.md

GenCRelatedFile.py

GenEdk2Files.py

GenMarkdown.py

GenRedfishSchemaCS.py

LICENSE.md

README.md

RedfishCSDef.py

ToolLogger.py

BootOption.v1_0_5.json

pycache

include

edk2library

RedfishCsIntp

src

Windows PowerShell

Windows PowerShell
版权所有 (C) Microsoft Corporation。保留所有权利。
安装最新的 PowerShell，了解新功能和改进！https://aka.ms/PSWindows
PS D:\Gitee\Redfish-Schema-C-Struct-Generator> cmd
Microsoft Windows [版本 10.0.22621.3007]
(c) Microsoft Corporation。保留所有权利。
D:\Gitee\Redfish-Schema-C-Struct-Generator>GenRedfishSchemaCS.py -edk2 -file=BootOption.v1_0_5.json

HPE Redfish Schema to C Structure Generator Copyrights 2018-2021 v1.2_schema2020.4
Generate C files of *BootOption.v1_0_5* from file: BootOption.v1_0_5.json

D:\Gitee\Redfish-Schema-C-Struct-Generator>

CSDN @jiangwei0512

These files can finally be found at <https://github.com/tianocore/edk2-redfish-client.git>. This open source library already contains the C structures corresponding to all JSON data supported by Redfish (however, the data versions may be different):

AttributeRegistry

Battery

BatteryCollection

BatteryMetrics

Bios

BootOption

v1_0_0

v1_0_1

v1_0_2

v1_0_3

v1_0_4

BootOption_V1_0_4_Dxe.c 1

RedfishBootOption_V1_0_4_Dxe.inf

名称	修改日期	类型	大小
BootOption_V1_0_5_Dxe.c	2024/1/28 12:46	VisualStudio.c.14.0	14 KB
RedfishBootOption_V1_0_5_Dxe.inf	2024/1/28 12:46	安装信息	2 KB

CSDN @jiangwei0512