Open Connectivity Foundation

admin@openconnectivity.org

OCF Upnp Bridge and Resources specification  
V0.1

Legal Disclaimer

THIS IS A DRAFT SPECIFICATION ONLY AND HAS NOT BEEN ADOPTED BY THE OPEN CONNECTIVITY FOUNDATION. THIS DRAFT SPECIFICATION MAY NOT BE RELIED UPON FOR ANY PURPOSE OTHER THAN REVIEW OF THE CURRENT STATE OF THE DEVELOPMENT OF THIS DRAFT SPECIFICATION. THE OPEN CONNECTIVITY FOUNDATION AND ITS MEMBERS RESERVE THE RIGHT WITHOUT NOTICE TO YOU TO CHANGE ANY OR ALL PORTIONS HEREOF, DELETE PORTIONS HEREOF, MAKE ADDITIONS HERETO, DISCARD THIS DRAFT SPECIFICATION IN ITS ENTIRETY OR OTHERWISE MODIFY THIS DRAFT SPECIFICATION AT ANY TIME. YOU SHOULD NOT AND MAY NOT RELY UPON THIS DRAFT SPECIFICATION IN ANY WAY, INCLUDING BUT NOT LIMITED TO THE DEVELOPMENT OF ANY PRODUCTS OR SERVICES. IMPLEMENTATION OF THIS DRAFT SPECIFICATION IS DONE AT YOUR OWN RISK AMEND AND IT IS NOT SUBJECT TO ANY LICENSING GRANTS OR COMMITMENTS UNDER THE OPEN CONNECTIVITY FOUNDATION INTELLECTUAL PROPERTY RIGHTS POLICY OR OTHERWISE. IN CONSIDERATION OF THE OPEN CONNECTIVITY FOUNDATION GRANTING YOU ACCESS TO THIS DRAFT SPECIFICATION, YOU DO HEREBY WAIVE ANY AND ALL CLAIMS ASSOCIATED HEREWITH INCLUDING BUT NOT LIMITED TO THOSE CLAIMS DISCUSSED BELOW, AS WELL AS CLAIMS OF DETRIMENTAL RELIANCE.

The OIC logo is a trademark of Open Interconnect Consortium, Inc. in the United States or other countries. \*Other names and brands may be claimed as the property of others.

Copyright © 2016 Open Interconnect Consortium, Inc. All rights reserved.

Copying or other form of reproduction and/or distribution of these works are strictly prohibited

CONTENTS

[1 Scope 6](#_Toc469670715)

[2 Normative references 7](#_Toc469670716)

[3 Terms, definitions symbols and abbreviations 10](#_Toc469670717)

[3.1 Terms and definitions 10](#_Toc469670718)

[3.3 Conventions 11](#_Toc469670719)

[4 Document conventions and organization 12](#_Toc469670720)

[4.1 Notation 12](#_Toc469670721)

[4.2 Data types 12](#_Toc469670722)

[5 Operational Scenarios 13](#_Toc469670723)

[6 Discovery 13](#_Toc469670724)

[6.1 Endpoint Discovery 13](#_Toc469670725)

[6.2 Resource Discovery 13](#_Toc469670726)

[7 Introspection 13](#_Toc469670727)

[7.1 Use of introspection 13](#_Toc469670728)

[8 Security 13](#_Toc469670729)

[9 Device Type Mapping 14](#_Toc469670730)

[9.1 Introduction 14](#_Toc469670731)

[9.1 UPnP Device Types to OCF Device Types 14](#_Toc469670732)

[10 Resource Definitions 16](#_Toc469670733)

[10.1 Introduction 16](#_Toc469670734)

[10.2 UPnP DCP Types to OCF Resource Types 16](#_Toc469670735)

[10.3 UPnP Device 16](#_Toc469670736)

[10.3.1 Introduction 16](#_Toc469670737)

[10.3.2 Example URI 16](#_Toc469670738)

[10.3.3 Resource Type 17](#_Toc469670739)

[10.3.4 RAML Definition 17](#_Toc469670740)

[10.3.5 Property Definition 18](#_Toc469670741)

[10.3.6 CRUDN Behaviour 18](#_Toc469670742)

[10.4 Icon 18](#_Toc469670743)

[10.4.1 Introduction 18](#_Toc469670744)

[10.4.2 Example URI 18](#_Toc469670745)

[10.4.3 Resource Type 18](#_Toc469670746)

[10.4.4 RAML Definition 18](#_Toc469670747)

[10.4.5 Property Definition 19](#_Toc469670748)

[10.4.6 CRUDN Behaviour 19](#_Toc469670749)

[10.5 UPnP Service 19](#_Toc469670750)

[10.5.1 Introduction 19](#_Toc469670751)

[10.5.2 Example URI 19](#_Toc469670752)

[10.5.3 Resource Type 19](#_Toc469670753)

[10.5.4 RAML Definition 19](#_Toc469670754)

[10.5.5 Property Definition 20](#_Toc469670755)

[10.5.6 CRUDN Behaviour 20](#_Toc469670756)

[10.6 UPnP State Variable 20](#_Toc469670757)

[10.6.1 Introduction 20](#_Toc469670758)

[10.6.2 Example URI 20](#_Toc469670759)

[10.6.3 Resource Type 21](#_Toc469670760)

[10.6.4 RAML Definition 21](#_Toc469670761)

[10.6.5 Property Definition 21](#_Toc469670762)

[10.6.6 CRUDN Behaviour 21](#_Toc469670763)

[10.7 UPnP Action 22](#_Toc469670764)

[10.7.1 Introduction 22](#_Toc469670765)

[10.7.2 Example URI 22](#_Toc469670766)

[10.7.3 Resource Type 22](#_Toc469670767)

[10.7.4 RAML Definition 22](#_Toc469670768)

[10.7.5 Property Definition 23](#_Toc469670769)

[10.7.6 CRUDN Behaviour 23](#_Toc469670770)

[10.8 Generic 23](#_Toc469670771)

[10.8.1 Introduction 23](#_Toc469670772)

[10.8.2 Example URI 23](#_Toc469670773)

[10.8.3 Resource Type 23](#_Toc469670774)

[10.8.4 RAML Definition 23](#_Toc469670775)

[10.8.5 Property Definition 24](#_Toc469670776)

[10.8.6 CRUDN Behaviour 24](#_Toc469670777)

Figures

**No table of figures entries found.**

Tables

[Table 8‑1 UPnP to OCF Device Type Mapping. 14](#_Toc469670778)

[Table 9‑1 UPnP to OCF Device Type Mapping. 16](#_Toc469670779)

[Table 9‑2 Resource Summary. 16](#_Toc469670780)

# 

# Scope

This document specifies and describes OCF-UPnP Data Models, the framework for translation between OCF devices and UPnP Devices, and specifies the behavior of a translator that exposes UPnP devices to OCF clients, and exposes OCF devices to UPnP applications. Translation of protocols other than UPnP is left to a future version of this specification. This document provides generic requirements that apply unless overridden by a more specific document.

# Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

OIC Specifications, *Open Interconnect Consortium Specifications*, Version 1.1.

<https://openconnectivity.org/resources/specifications>

* OIC Core Specification, *Open Interconnect Consortium Core Specification*, Version 1.1, October 11, 2016

<https://openconnectivity.org/specs/OIC_Core_Specification_v1.1.0.pdf>

* OIC Resource Type Specification, *Open Interconnect Consortium Resource Type Specification*, Version 1.1, October 11, 2016

<https://openconnectivity.org/specs/OIC_Resource_Type_Specification_v1.1.0.pdf>

* OIC Security Specification, *Open Interconnect Consortium Security Specification*, Version 1.1, October 11, 2016

<https://openconnectivity.org/specs/OIC_Security_Specification_v1.1.0.pdf>

OCF UPnP Specifications-Standards: Device Control Protocols, *Universal Plug and Play - Standards: Device Control Protocols* *Specifications*  
<https://openconnectivity.org/resources/specifications/upnp/specifications>

OCF UPnP Add-on Services Specification

* DeviceProtection:1 **Service** Specifications, *DeviceProtection:1 Service* *Specifications*  
  <https://openconnectivity.org/resources/specifications/upnp/specifications/device-protection-v-1-0>

OCF UPnP Certification Testing - *OCF* *UPnP Certification Testing*

<https://openconnectivity.org/certification/upnp-certification>

OCF-UPnP Data Models, *OCF-UPnP Data Models*  
<https://github.com/openconnectivityfoundation/UPnP-models>

IEEE 754, *IEEE Standard for Floating-Point Arithmetic*, August 2008

IETF RFC 4122, A Universally Unique IDentifier (UUID) URN Namespace, July 2005  
<https://www.rfc-editor.org/info/rfc4122>

IETF RFC 4648, The Base16, Base32, and Base64 Data Encodings, October 2006  
<https://www.rfc-editor.org/info/rfc4648>

IETF RFC 6973, *Privacy Considerations for Internet Protocols,* July 2013  
<https://www.rfc-editor.org/info/rfc6973>

IETF RFC 7049, *Concise Binary Object Representation (CBOR)*, October 2013  
<https://www.rfc-editor.org/info/rfc7049>

IETF RFC 7159, *The JavaScript Object Notation (JSON) Data Interchange Format*, March 2014 [http://www/ietf.org/rfc/rfc7159.txt](http://www/ietf.org/rfc/rfc7159.txt%20)

JSON Schema Core, *JSON Schema: core definitions and terminology*, January 2013  
<http://json-schema.org/latest/json-schema-core.html>

JSON Schema Validation, *JSON Schema: interactive and non interactive validation*, January 2013  
<http://json-schema.org/latest/json-schema-validation.html>

RAML, *Restful API modelling language,* Version 0.8.

<https://github.com/raml-org/raml-spec/blob/master/raml-0.8.md>

# Terms, definitions symbols and abbreviations

## Terms and definitions

Term

Definition

UPnP Bridge Device(s)

An UPnP Bridge Device(s) is(are) conformant to the normative requirements contained in this specification.

UPnP Device(s)

A Device(s) that is(are) conformant to the normative requirements contained in *Universal Plug and Play - Standards: Device Control Protocols* *Specifications*.

* 1. Symbols and abbreviations

**UPnP**

Universal Plug and Play

A set of [networking protocols](https://en.wikipedia.org/wiki/Networking_protocol) that permits networked [devices](https://en.wikipedia.org/wiki/Peripheral_device), such as personal computers, printers, Internet gateways, [Wi-Fi](https://en.wikipedia.org/wiki/Wi-Fi) access points and mobile devices to seamlessly discover each other's presence on the network and establish functional [network services](https://en.wikipedia.org/wiki/Network_service) for data sharing, communications, and entertainment. It is fully defined in UPnP Specifications-Standards: Device Control Protocols. Since 2016, all UPnP standards are now managed by the OCF.

**DCP**

Device Control Protocols are the specifications, produced by specific UPnP Task Groups, which have been standardized.

**CRUDN**

Create Read Update Delete Notify

Indicating which operations are possible on the resource

**CSV**

Comma Separated Value List

Construction to have more fields in 1 string separated by commas. If a value contains a comma, then the comma can be escaped by adding “\” in front of the comma.

**OCF**

Open Connectivity Foundation

Organization that created these specifications

**RAML**

RESTful API Modeling Language

Simple and succinct way of describing practically RESTful APIs (see RAML)

## Conventions

In this specification, a number of terms, conditions, mechanisms, sequences, parameters, events, states, or similar terms are printed with the first letter of each word in uppercase and the rest lowercase (e.g., Network Architecture). Any lowercase uses of these words have the normal technical English meaning.

# Document conventions and organization

For the purposes of this document, the terms and definitions given in OIC Core Specification and OIC Resource Type Specification apply.

## Notation

In this document, features are described as required, recommended, allowed or DEPRECATED as follows:

Required (or shall or mandatory).

These basic features shall be implemented to comply with OIC Core Architecture. The phrases “shall not”, and “PROHIBITED” indicate behavior that is prohibited, i.e. that if performed means the implementation is not in compliance.

Recommended (or should).

These features and functionality supported by OIC Core Architecture and should be implemented. Recommended features take advantage of the capabilities OIC Core Architecture, usually without imposing major increase of complexity. Notice that for compliance testing, if a recommended feature is implemented, it shall meet the specified requirements to be in compliance with these guidelines. Some recommended features could become requirements in the future. The phrase “should not” indicates behavior that is permitted but not recommended.

Allowed (or allowed).

These features are neither required nor recommended by OIC Core Architecture, but if the feature is implemented, it shall meet the specified requirements to be in compliance with these guidelines.

Conditionally allowed (CA)

The definition or behaviour depends on a condition. If the specified condition is met, then the definition or behaviour is allowed, otherwise it is not allowed.

Conditionally required (CR)

The definition or behaviour depends on a condition. If the specified condition is met, then the definition or behaviour is required. Otherwise the definition or behaviour is allowed as default unless specifically defined as not allowed.

DEPRECATED

Although these features are still described in this specification, they should not be implemented except for backward compatibility. The occurrence of a deprecated feature during operation of an implementation compliant with the current specification has no effect on the implementation’s operation and does not produce any error conditions. Backward compatibility may require that a feature is implemented and functions as specified but it shall never be used by implementations compliant with this specification.

Strings that are to be taken literally are enclosed in “double quotes”.

Words that are emphasized are printed in italic.

## Data types

See OIC Core Specification.

# Operational Scenarios

The overall goals are to:

1. Make UPnP Devices appear to OCF clients as if they were native OCF devices, and
2. Make OCF devices appear to non-OCF (e.g., UPnP) applications as if they were native non-OCF devices

# Discovery

## Endpoint Discovery

Section intentionally left blank - Todo

~~Clients may discover Servers by using the mechanisms defined by the OIC Core Specification Section 10. A Client may populate an “rt” query parameter with the Device Types that the Client wants to discover, or if no “rt” query parameter is provided then the search is for all available Device Types irrespective.~~

~~Smart Home Devices may be discovered by Device Type or implemented Resource Type. This difference is conveyed by the wanted “rt” argument of the OIC Core Specification discovery method (see section 11.3 of the OIC Core Specification).~~

~~The values that may be used for discovering a specific Device Type are listed in Table 10‑1 . The values that may be used to discover a specific Resource Type are listed in the OIC Resource Type Specification in section 6.~~

~~The discovery process provides the base URI of the Device that is acting as a Server to the Client. The structure of the detected Device can then be retrieved by Resource Discovery.~~

## Resource Discovery

Section intentionally left blank - Todo

# Introspection

## Use of introspection

Whenever possible, the translation code should make use of metadata available that indicates what the sender and recipient of the message in question are expecting. For example, devices certified by the *OCF* *UPnP Certification Testing* are required to carry the introspection data for each object and interface they expose. The *OIC 1.1 Core Specification* makes no such requirement, but the *OCF 1.0 Core Specification* will. When the metadata is available, translators should convert the incoming payload to exactly the format expected by the recipient and should use information when translating replies to form a more useful message.

# Security

UPnP Bridge Device shall implement the mandated security Resources specified in OIC Security Specification. Additionally UPnP Bridge Device shall secure all links used to access Resources using DTLS

The general problem is how to ensure that devices and controllers in OCF and UPnP contexts to not open significant security risks when connected via a bridge. The process of looking at this problem in the context of OCF and UPnP will be instructive in the more general problem of connecting any ecosystem to OCF through a bridge.

Here are some attack vectors to consider:

* UPnP control points might get unauthorized access to OCF servers and do bad stuff.
* OCF clients might get unauthorized access to UPnP devices and do similarly bad stuff.
* Bad actors may get access to a compromised bridge and wreak havoc all around.
* Communication channels might be compromised and rogue messages inserted.
* Communications might be vulnerable to DOS attacks.

This is just some security risks this will need to review by the Security Work Group.

UPnP has a security service called DeviceProtection that allows for x.509 authentication and secure communications over TLS. DeviceProtection is optional for UPnP V1.0 and V1.1 and required UPnP V2.0. While many existing UPnP implementations have chosen **not** to implement the DeviceProtection service and of course it is available for use when implemented. Based on that, it could argue the following points:

* UPnP devices and control points that implement DeviceProtection have adequate security to be used in the IoT contexts addressed by OCF. In other words, they can be authenticated, have levels of authorization associated with different roles, and communicate with each other over trusted secure channels.
* UPnP devices that do not implement DeviceProtection should not be trusted. They should only be used in contexts where that trust is not a requirement or confirmed by the Application/End-user.

Therefore, it suggested the following general policy for bridging to UPnP (and similar no-secure ecosystems):

* OCF is responsible for all aspects of security on the OCF side of the bridge.
* OCF is responsible for the security of the bridge itself.
* If an UPnP control point does not implement UPnP DeviceProtection then the bridge shall limit its access and authorization in the OCF domain to contexts that don’t require security (if there are any). An UPnP “control point” is any device that can send UPnP control point messages regardless of whether it has UPnP device features.
* If an UPnP device does not implement UPnP DeviceProtection, OCF does not necessarily need to do anything. The OCF side of the bridge must not trust the device, but the insecurity of the UPnP side of the bridge is not expected to be repaired by OCF.
* If a UPnP device or control point implements DeviceProtection (and is certified), and if it has OCF-trusted credentials, it may be trusted to act as a full participant in the OCF ecosystem. (NOTE: UPnP DeviceProtection allows for self-signed certificates. OCF policy must decide whether to trust these certificates.)

# Device Type Mapping

## Introduction

This Section contains the mappings of to/from Device Types.

## UPnP Device Types to OCF Device Types

The following table captures the equivalency mapping between UPnP defined Device Types (see UPnP Specifications-Standards: Device Control Protocols) and OCF defined Device Types (see Table 10-1 in OIC Smart Home Device Specification). The minimum Resource sets for each OCF Device is provided in OIC Smart Home Device Specification.

Table 8‑1 UPnP to OCF Device Type Mapping.

|  |  |  |
| --- | --- | --- |
| Classification | UPnP Device Type | OCF Device Type |
| Lighting | BinaryLight:1 | oic.d.light |
| DimmableLight:1 |

# Resource Definitions

This section contains definitions for all Resource Types; the complete set is listed in **Error! Reference source not found.**. All sections provide example representations of the Resource Type following the application of the default interface that is applied for that specific Resource Type.

All Resource Types shall be created in accordance with the OIC Core Specification Section 7.2. All comparisons against a Resource Type shall be case insensitive.

All Resource Types in this document are prefixed with “oic.r” denoting that it is an OIC defined Resource Type.

## Introduction

This Section contains the mappings of to/from Resource Types.

## UPnP DCP Types to OCF Resource Types

The following table captures the equivalency mapping between UPnP defined Service Types (see UPnP Specifications-Standards: Device Control Protocols) and OCF defined Resource Types (see Table 10-1 in OIC Smart Home Device Specification). The minimum Resource sets for each OCF Device is provided in OIC Smart Home Device Specification.

Table 9‑1 UPnP to OCF Device Type Mapping.

|  |  |  |
| --- | --- | --- |
| Classification | UPnP Service Type | OCF Resource Type |
| Lighting | SwitchPower:1 | oic.r.switch.binary |
| Dimming:1 | oic.r.switch.binary |
| oic.r.light.brightness |

Table 9‑2 Resource Summary.

|  |  |  |
| --- | --- | --- |
| Friendly Name | Resource Type | Section |
|  |  | <link> |

## UPnP Device

### Introduction

This resource provides the list of UPnP Device available on the device. The devices are an array of upnpDevice(s) as separately defined. The basic resource type oic.r.deviceList is not able provide the serviceList and iconlist which are provide here in this device.

### Example URI

/upnpDeviceResURI

### Resource Type

The resource type (rt) is defined as: oic.r.upnp.device.

### RAML Definition

#%RAML 0.8

title: *UPnP Device*

version: *v1.1.0-20160519*

traits:

- interface :

queryParameters:

if:

enum: ["oic.if.baseline"]

/upnpDeviceResURI:

description: |

This resource provides the list of UPnP Device available on the device.  
 The devices are an array of upnpDevice(s) as separately defined.  
 The basic resource type oic.r.deviceList is not able provide the serviceList and iconlist which are provide here in this device.

is : ['interface']

get:

description: |

Retrieves the current upnp device.

responses :

200:

body:

application/json:

schema: |

upnpDevice

example: |

{  
 "rt": ["oic.r.upnp.device", "oic.wk.col"],  
 "if": ["oic.if.baseline"],  
 "ins": "uuid:UUID",  
 "id": "/oic/upnp/device/uuid:UUID",  
 "deviceType": "device\_type",  
 "friendlyName": "friendly\_name",  
 "manufacturer": "manufacturer\_name",  
 "modelDescription": "model\_description",  
 "modelName": "model\_name",  
 "modelNumber": "model\_number",  
 "modelURL": "model\_url",  
 "serialNumber": "serial\_number",  
 "UDN": "uuid:UUID",  
 "UPC": "Universal\_Product\_Code",  
 "links":  
 [  
 {  
 "href": "/icon-small",  
 "sizes": ["48x48"],  
 "type": "image/png",  
 "rel": "icon",  
 "ins": "uuid:UUID",  
 "rt": ["oic.r.icon"],  
 "if": ["oic.if.r"]  
 },  
 {  
 "href": "/icon-large",  
 "sizes": ["192x192"],  
 "type": "image/png",  
 "rel": "icon",  
 "ins": "uuid:UUID",  
 "rt": ["oic.r.icon"],  
 "if": ["oic.if.r"]  
 },  
 {  
 "href": "/service-name\_1st",  
 "ins": "uuid:UUID",  
 "rt": ["oic.r.upnp.service"],  
 "if": ["oic.if.baseline"]  
 },  
 {  
 "href": "/service-name\_nth",  
 "ins": "uuid:UUID",  
 "rt": ["oic.r.upnp.service"],  
 "if": ["oic.if.baseline"]  
 }  
 ]  
 }

### Property Definition

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Property name | Value type | Mandatory | Access mode | Description |

### CRUDN Behaviour

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Resource | Create | Read | Update | Delete | Notify |
| /upnpDeviceResURI |  | get |  |  | get |

## Icon

### Introduction

This resource describes the attributes associated with an Icon.

### Example URI

/IconResURI

### Resource Type

The resource type (rt) is defined as: oic.r.icon.

### RAML Definition

#%RAML 0.8

title: *OICIcon*

version: *v1.1.0-20161107*

traits:

- interface :

queryParameters:

if:

enum: ["oic.if.r", "oic.if.baseline"]

/IconResURI:

description: |

This resource describes the attributes associated with an Icon.

is : ['interface']

get:

description: |

Retrieves the current icon properties.

responses :

200:

body:

application/json:

schema: |

Icon

example: |

{  
 "rt": ["oic.r.icon"],  
 "id": "unique\_example\_id",  
 "mimeType": "mime\_type\_string",  
 "sizes": "64x64",  
 "media": "media\_string"  
 }

### Property Definition

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Property name | Value type | Mandatory | Access mode | Description |

### CRUDN Behaviour

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Resource | Create | Read | Update | Delete | Notify |
| /IconResURI |  | get |  |  | get |

## UPnP Service

### Introduction

This resource provides the list of upnp service available on the device. The services are an array of upnpService(s) as separately defined. The basic resource type oic.r.device is not able provide the actionList which are provide here in this device.

### Example URI

/upnpServiceResURI

### Resource Type

The resource type (rt) is defined as: oic.r.upnp.service.

### RAML Definition

#%RAML 0.8

title: *UPnP Service*

version: *v1.1.0-20160519*

traits:

- interface :

queryParameters:

if:

enum: ["oic.if.baseline"]

/upnpServiceResURI:

description: |

This resource provides the list of upnp service available on the device.  
 The services are an array of upnpService(s) as separately defined.  
 The basic resource type oic.r.device is not able provide the actionList which are provide here in this device.

is : ['interface']

get:

description: |

Retrieves the current upnp service.

responses :

200:

body:

application/json:

schema: |

upnpService

example: |

{  
 "rt": ["oic.r.upnp.service", "oic.wk.col"],  
 "if": ["oic.if.baseline"],  
 "ins": "uuid:UUID",  
 "id": "/oic/upnp/service/uuid:UUID",  
 "serviceType": "service\_type",  
 "serviceId": "service\_id",  
 "SCPDURL": "scpd\_url\_string",  
 "controlURL": "control\_url\_string",  
 "eventSubURL": "event\_sub\_url\_string",  
 "links":  
 [  
 {  
 "href": "/stateVariable-name\_1st",  
 "ins": "uuid:UUID",  
 "rt": ["oic.r.upnp.stateVariable"],  
 "if": ["oic.if.baseline"]  
 },  
 {  
 "href": "/stateVariable-name\_nth",  
 "ins": "uuid:UUID",  
 "rt": ["oic.r.upnp.stateVariable"],  
 "if": ["oic.if.baseline"]  
 },  
 {  
 "href": "/action-name\_1st",  
 "ins": "uuid:UUID",  
 "rt": ["oic.r.upnp.action"],  
 "if": ["oic.if.baseline"]  
 },  
 {  
 "href": "/action-name\_nth",  
 "ins": "uuid:UUID",  
 "rt": ["oic.r.upnp.action"],  
 "if": ["oic.if.baseline"]  
 }  
 ]  
 }

### Property Definition

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Property name | Value type | Mandatory | Access mode | Description |

### CRUDN Behaviour

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Resource | Create | Read | Update | Delete | Notify |
| /upnpServiceResURI |  | get |  |  | get |

## UPnP State Variable

### Introduction

This resource describes whether a upnp state variables.

### Example URI

/upnpStateVariableResURI

### Resource Type

The resource type (rt) is defined as: oic.r.upnp.stateVariable.

### RAML Definition

#%RAML 0.8

title: *U*

version: *v1.1.0-20160519*

traits:

- interface :

queryParameters:

if:

enum: ["oic.if.r", "oic.if.baseline"]

/upnpStateVariableResURI:

description: |

This resource describes whether a upnp state variables.

is : ['interface']

get:

description: |

Retrieves the current upnp state variable.

responses :

200:

body:

application/json:

schema: |

upnpStateVariable

example: |

{  
 "rt": ["oic.r.upnp.stateVariable"],  
 "id": "unique\_example\_id",  
 "name": "state\_variable\_name",  
 "dataType": "dataType\_string",  
 "defaultValue": "defaultValue\_string",  
 "allowedValueList": [  
 {  
 "allowedValue": "allowedValue\_1st"  
 },  
 {  
 "allowedValue": "allowedValue\_nth"  
 }  
 ]  
 }

### Property Definition

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Property name | Value type | Mandatory | Access mode | Description |

### CRUDN Behaviour

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Resource | Create | Read | Update | Delete | Notify |
| /upnpStateVariableResURI |  | get |  |  | get |

## UPnP Action

### Introduction

This resource provides the action available on the service. The sources are an array of mediaSource(s) as separately defined. The basic resource type oic.r.service is not able provide the inputArgs and outputArgs which are provide here in this device.

### Example URI

/upnpActionResURI

### Resource Type

The resource type (rt) is defined as: oic.r.upnp.action.

### RAML Definition

#%RAML 0.8

title: *UPnP Action*

version: *v1.1.0-20160519*

traits:

- interface :

queryParameters:

if:

enum: ["oic.if.rw", "oic.if.baseline"]

/upnpActionResURI:

description: |

This resource provides the action available on the service.  
 The sources are an array of mediaSource(s) as separately defined.  
 The basic resource type oic.r.service is not able provide the inputArgs and outputArgs which are provide here in this device.

is : ['interface']

post:

description: |

Changes the status of the action(s).  
 Allows changes of the sourceName and the status.

body:

application/json:

schema: |

upnpAction

example: |

{  
 "rt": ["oic.r.upnp.action"],  
 "ins": "uuid:UUID",  
 "id": "/oic/upnp/service\_name/uuid:UUID",  
 "actionName": "action\_name",  
 "inputArgs": [  
 {  
 "genName": "arg\_name\_1st",  
 "genType": "arg\_type\_1st",  
 "genValue": "arg\_value\_1st"  
 },  
 {  
 "genName": "arg\_name\_nth",  
 "genType": "arg\_type\_nth",  
 "genValue": "arg\_value\_nth"  
 }  
 ]  
 }

responses :

200:

body:

application/json:

schema: |

upnpAction

example: |

{  
 "rt": ["oic.r.upnp.action"],  
 "id": "unique\_example\_id",  
 "actionName": "action\_name",  
 "outputArgs": [  
 {  
 "genName": "arg\_name\_1st",  
 "genType": "arg\_type\_1st",  
 "genValue": "arg\_value\_1st"  
 },  
 {  
 "genName": "arg\_name\_nth",  
 "genType": "arg\_type\_nth",  
 "genValue": "arg\_value\_nth"  
 }  
 ]  
 }

### Property Definition

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Property name | Value type | Mandatory | Access mode | Description |

### CRUDN Behaviour

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Resource | Create | Read | Update | Delete | Notify |
| /upnpActionResURI |  |  | post |  | get |

## Generic

### Introduction

This resource describes whether a Generic Information.

### Example URI

/GenericResURI

### Resource Type

The resource type (rt) is defined as: oic.r.generic.

### RAML Definition

#%RAML 0.8

title: *OICGeneric*

version: *v1.1.0-20160519*

traits:

- interface :

queryParameters:

if:

enum: ["oic.if.r", "oic.if.baseline"]

/GenericResURI:

description: |

This resource describes whether a Generic Information.

is : ['interface']

get:

responses :

200:

body:

application/json:

schema: |

Generic

example: |

{  
 "rt": ["oic.r.generic"],  
 "id": "unique\_example\_id",  
 "genName": "generic\_name",  
 "genType": "generic\_type",  
 "genValue": "generic\_value"  
 }

### Property Definition

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Property name | Value type | Mandatory | Access mode | Description |

### CRUDN Behaviour

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Resource | Create | Read | Update | Delete | Notify |
| /GenericResURI |  | get |  |  | get |