OCF Core Specification Extension

WiFi Easy Setup

VERSION 1.3.0 | December 2017

OPEN CONNECTIVITY FOUNDATION™

CONTENTS

Figures

Tables

1 Scope

This specification defines functional extensions to the capabilities defined in the OCF Core Specification to meet the requirements of Wi-Fi Easy Setup. This specification specifies new Resource Types to enable the functionality and any extensions to the existing capabilities defined in the OCF Core Specification.

2 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.

OCF Core Specification, *Open Connectivity Foundation Core Specification*, Version 1.3
Available at: https://openconnectivity.org/specs/OCF_Core_Specification_v1.3.0.pdf
Latest version available at: https://openconnectivity.org/specs/OCF_Core_Specification.pdf

OCF Security Specification, *Open Connectivity Foundation Security Capabilities*, Version 1.3
Available at: https://openconnectivity.org/specs/OCF_Security_Specification_v1.3.0.pdf
Latest version available at: https://openconnectivity.org/specs/OCF_Security_Specification.pdf

IEEE 802.11:2016, IEEE Standard for Information technology—Telecommunications and information exchange between systems Local and metropolitan area networks—Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, December 2016
https://standards.ieee.org/findstds/standard/802.11-2016.html

IETF RFC 7159, *The JavaScript Object Notation (JSON) Data Interchange Format*, March 2014
https://www.rfc-editor.org/info/rfc7159

IETF RFC 7252, *The Constrained Application Protocol (CoAP)*, June 2014
https://www.rfc-editor.org/info/rfc7252

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

OpenAPI specification, *aka Swagger RESTful API Documentation Specification*, Version 2.0
https://github.com/OAI/OpenAPI-Specification/blob/master/versions/2.0.md

3 Terms, definitions, symbols and abbreviations

All terms and definitions as defined in the OCF Core Specification also apply to this specification.

3.1 Terms and definitions

As defined in the OCF Core Specification with the following additions

3.1.1.
Easy Setup Enrollment
Easy Setup Enrollment is a step during Easy Setup in which the Enrollee is contacted by the Mediator to configure the Enroller's information by means of accessing Easy Setup Resources.

3.1.2.
Enrollee
The Device that needs to be configured and connected. E.g. Air-conditioner, Printer.

3.1.3.
Enroller
Target network entity to which the Enrollee connects. E.g. Wi-Fi AP

3.1.4.
Mediator
Device that enables the Enrollee to connect to the target network (Enroller). The Mediator transfers configuration information to the Enrollee. E.g. Mobile Phone

3.1.5.
Easy Setup
Process of configuring an Enrollee using Mediator (by transferring of essential information to the Enrollee).

3.1.6.
Soft AP
Software Enabled Access Point hosted on the device which is not a dedicated Access Point.

3.2 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.

3.3 Data types

As defined in the OCF Core Specification.

4 Document conventions and organization

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

Required (or shall or mandatory)(M).

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

Recommended (or should)(S).

- These features add functionality supported by Core Architecture and should be implemented. Recommended features take advantage of the capabilities 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 behaviour that is permitted but not recommended.

Allowed (may or allowed)(O).

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

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.

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.

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

Words that are emphasized are printed in italic.

5 Overview

5.1 Introduction

This specification describes a way to setup and configure a new OCF Device, using an already configured OCF Device or onboarding tool.

The described setup and configure mechanism is optional and other mechanisms are allowed to be used.

Specifically, this method allows the transferring of essential information to the new Device, which includes:

- Local network connection information, e.g. in case of Wi-Fi it will be Wi-Fi access point information.
- Device Configuration: Additional Device configuration information.

Easy Setup can be enhanced in future by incorporating other suitable technologies.

5.2 Architecture

Figure 1 shows the deployment architectural approach.

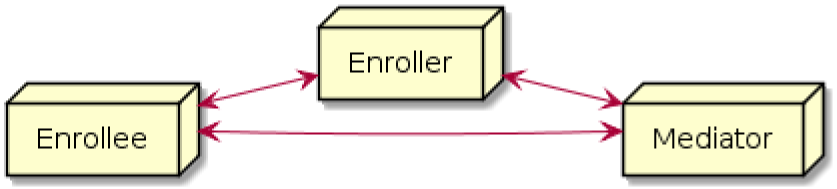

Figure 1. Easy Setup deployment architecture

Easy Setup defines the following roles: Enrollee, Enroller, and Mediator. Please refer to Section 3.1 for definitions thereof.

5.3 Example Scenario

The following scenario presents a typical setup case.

The configuration information and steps taken may vary depending on the Device’s type and status.

1. The Enrollee enters Easy Setup mode (when the Device is unboxed for the first time, it may be in this mode by default).
2. The Mediator discovers and connects to the Enrollee.
3. The Mediator performs Security Provisioning of the Enrollee.
4. The Mediator transmits Wi-Fi Setting Information to the Enrollee.
5. Using the information received from the Mediator, the Enrollee connects to the Enroller (Wi-Fi AP).

6 Resource model

6.1 Introduction

Devices capable of Easy Setup shall support the following Resource Types.

1. EasySetup Resource Type
2. WiFiConf Resource Type
3. DevConf Resource Type

The EasySetup Resource Type is a Collection Resource and shall contain Links to instances of at least WiFiConf and DevConf. A vendor may add links to other Resource Types.

Note that the EasySetup Resource Type supports the batch Interface (oic.if.b) which allows for efficient data delivery with a single request rather than multiple requests to each linked Resource.

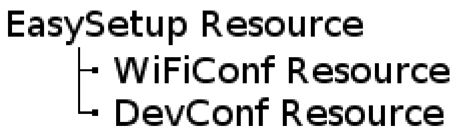

Figure 2. Easy Setup Resource Types

6.2 EasySetup Resource

6.2.1 Overview

The EasySetup Resource stores useful information including current status of Enrollee and last error code which was produced in the process of Easy Setup.

6.2.2 Resource

The Easy Setup Resource Type is as defined in Table 1. EasySetup Resource Type.

Table 1. EasySetup Resource Type

Example URI	Resource Type Title	Resource Type ID ("rt" value)	Interfaces	Description	Related Functional Interaction
/example/EasySetupResURI	EasySetup	oic.r.easysetup, oic.wk.col	oic.if.baseline, oic.if.ll, oic.if.b	Top level Resource for Easy Setup. Indicates easy setup status. The Resource properties exposed are listed in Table 2.	

Table 2. "oic.r.easysetup" Resource Type definition defines the details for the "oic.r.easysetup" Resource Type.

Table 2. "oic.r.easysetup" Resource Type definition

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
Easy Setup Provisioning Status	ps	integer	enum		R	Yes	Easy setup provisioning status of the Device 0: Need to Setup, 1: Connecting to Enroller, 2: Connected to Enroller,

							3: Failed to Connect to Enroller, 4~254: Reserved, 255: EOF
Last Error Code	lec	integer	enum		R	Yes	Indicates a failure reason if it fails to connect to Enroller 0: NO error, 1: Given SSID is not found, 2: Wi-Fi password is wrong, 3: IP address is not allocated, 4: NO internet connection, 5: Timeout, 6: Wi-Fi Auth Type is not supported by the Enrollee, 7: Wi-Fi Encryption Type is not supported by the Enrollee, 8: Wi-Fi Auth Type is wrong (failure while connecting to the Enroller), 9: Wi-Fi Encryption Type is wrong (failure while connecting to the Enroller), 10~254: Reserved, 255: Unknown error.
Connect	cn	array of integer			RW	Yes	Array of connection types to trigger Enrollee to initiate connection: 1 : Wi-Fi, 2 : Other transport to be added in a future (e.g. BLE))
Links	links	array			R	Yes	Array of links that are WiFiConf and DevConf Resource.

Enrollee shall set the following as default values (for example, when Device is unboxed first time):

- "ps" equal to 0.
- "lec" equal to 0.
- "cn" equal to an empty array.

6.3 WiFiConf Resource Type

6.3.1 Introduction

The WiFiConf Resource Type stores information to help an Enrollee to connect to an existing Wi-Fi AP.

6.3.2 Resource Type

The WiFiConf Resource Type is as defined in Table 3. WiFiConf Resource Type.

Table 3. WiFiConf Resource Type

Example URI	Resource Type Title	Resource Type ID ("rt" value)	Interfaces	Description	Related Functional Interaction
/example/WiFiConfResURI	WiFiConf	oic.r.wificonf	oic.if.baseline, oic.if.rw	Contains Wi-Fi related properties The Resource properties exposed are listed in Table 4.	

Table 4. "oic.r.wificonf" Resource Type definition defines the details for the "oic.r.wificonf" Resource Type.

Table 4. "oic.r.wificonf" Resource Type definition

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
Supported Wi-Fi Mode Type	swmt	array of string	enum		R	Yes	Supported Wi-Fi modes by Enrollee. Can be multiple. ("A", "B", "G","N", "AC")
Supported Wi-Fi Frequency	swf	array of string	Refer to description for valid values.		R	Yes	Supported Wi-Fi frequencies by Enrollee. Can be multiple. ("2.4G", "5G")
Target Network Name	tnn	string			RW	Yes	Target network name (SSID of Wi-Fi AP i.e. enroller)
Credential	cd	string			RW	No	Credential information of Wi-Fi AP (Password used to connect to enroller).
Wi-Fi Auth Type	wat	string	enum		RW	Yes	Wi-Fi auth type ("None", "WEP", "WPA_PSK", "WPA2_PSK")
Wi-Fi Encryption Type	wet	string	enum		RW	Yes	Wi-Fi encryption type ("None", "WEP_64", "WEP_128", "TKIP", "AES", "TKIP_AES")
Supported Wi-Fi Auth Type	swat	array of string	enum		R	Yes	Supported Wi-Fi Auth types. Can be multiple. ("None", "WEP", "WPA_PSK", "WPA2_PSK")
Supported Wi-Fi Encryption Type	swet	array of string	enum		R	Yes	Supported Wi-Fi Encryption types. Can be multiple. ("None", "WEP-64", "WEP_128", "TKIP", "AES", "TKIP_AES")

6.4 DevConf Resource Type

6.4.1 Introduction

The DevConf Resource Type stores Device configuration information required in Wi-Fi Easy Setup.

6.4.2 Resource Type

The DevConf Resource Type is as defined in Table 5. DevConf Resource Type

Table 5. DevConf Resource Type

Example URI	Resource Type Title	Resource Type ID ("rt" value)	Interfaces	Description	Related Functional Interaction
/example/DevConfResURI	**DevConf**	oic.r.devconf	oic.if.baseline, "oic.if.r"	Stores device configuration information required in Easy Setup process The Resource properties exposed are listed in Table 6.	

Table 6. "oic.r.devconf" Resource Type definition defines the details for the "oic.r.devconf" Resource Type.

Table 6. "oic.r.devconf" Resource Type definition

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
Device Name	dn	one of: string or array of object			R	Yes	Indicates a pre-configured device name in language indicated by 'dl' in /oic/con. or An array of objects where each object has a 'language' field (containing an IETF RFC 5646 language tag) and a 'value' field containing the pre-configured device name in the indicated language. The pre-configured device name is presented by enrollee to mediator during easy-setup process.

7 Network and connectivity

Both the Mediator and Enrollee communicate via a common connectivity (e.g. Wi-Fi).

If using Wi-Fi for Easy Setup then the Enrollee shall have capability to act as a Soft AP. A Soft AP shall support the access point requirements defined by IEEE 802.11:2016.

8 Functional interactions

8.1 Onboarding, Provisioning and Configuration

Mediator may perform Ownership Transfer on the Enrollee and may also perform ACL provisioning. If it does so, the Mediator must conform to any existing DOXS and AMS requirements respectively. For details refer to the OCF Security Specification.

8.2 Resource discovery

The Mediator connects to the Enrollee via a mutually supported connection.

When in Easy Setup phase, if using Wi-Fi as the connectivity between the Enrollee and the Mediator then the Enrollee shall make itself discoverable as a Soft AP. The Soft AP has additional availability constraints which are documented in the OCF Security Specification.

8.3 Retrieving and Updating Easy Setup Resources

The Enrollee shall expose Easy Setup Resources such that a Mediator is able to discover them using standard OCF Resource discovery methods (i.e. via a RETRIEVE on /oic/res); see the OCF Core Specification, Section 11.3.

Easy Setup Resources shall expose only secure Endpoints (e.g. CoAPS); see the OCF Core Specification, Section 10.

The Mediator may request retrieval of an Easy Setup Resource to check the Enrollee's status at any stage of Easy Setup. This applies only when the Enrollee & the Mediator are on a common network.

The Mediator may request that the Enrollee update its Resource Property(-ies). Upon request from the Mediator the Enrollee shall update its current Resource Property Values, and shall perform any required action. For example, if the "cn" Property of "EasySetup" Resource is updated by the Mediator, to indicate connection to Wi-Fi, the Enrollee shall start the connection to Enroller.

For details of Easy Setup Resources refer to Section 6.

8.4 Error Handling

The "lec" Property of the EasySetup Resource (i.e. oic.r.easysetup) is used to indicate the error that occurred in the Easy Setup process while trying to connect to the Enroller (using the information provided by the Mediator in WiFiConf Resource):

- The Enrollee shall set "lec" Property to 1, if it fails to connect because it can't find the SSID.
- The Enrollee shall set "lec" Property to 2, if it fails to connect due to wrong credential (password) information.
- The Enrollee should set "lec" Property to 6, if the Auth type is not supported by the Enrollee.
- The Enrollee should set "lec" Property to 7, if the Encryption type is not supported by the Enrollee.
- The Enrollee should set "lec" Property to 8, if it fails to connect due to wrong Auth type information (even though it's supported by the Enrollee).
- The Enrollee should set "lec" Property to 9, if it fails to connect due to wrong Encryption type information (even though it's supported by the Enrollee).

When using Wi-Fi as the connectivity between the Enrollee and Mediator, if the Enrollee fails to connect to the Enroller, it shall again make itself discoverable as a Soft AP (in case it destroyed its Soft AP earlier).

8.5 Example Easy Setup Flow

The following figure shows an example Easy Setup flow for informative purposes:

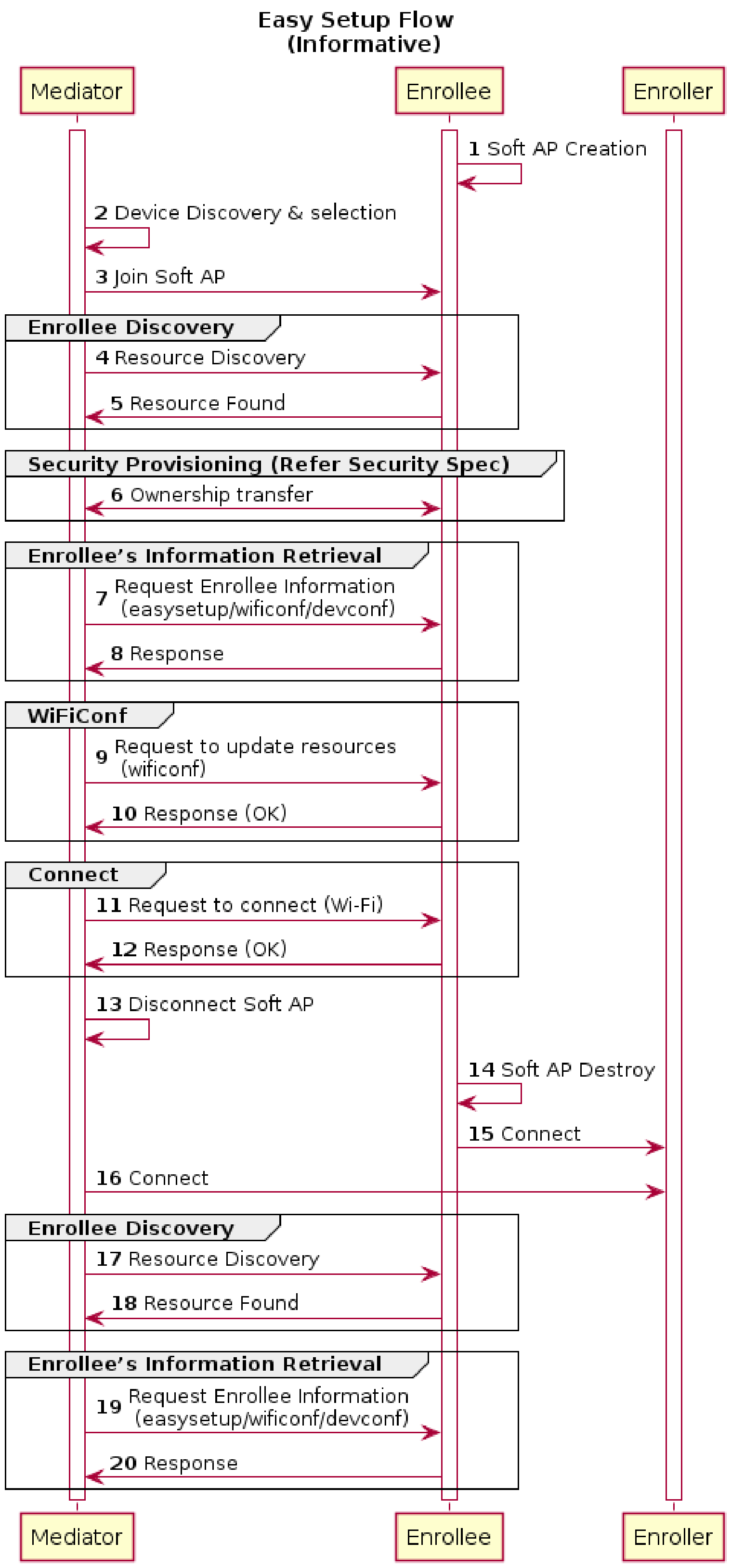
Easy Setup Flow
(Informative)
Mediator
Enrollee
Enroller
1 Soft AP Creation
2 Device Discovery & selection
3 Join Soft AP
Enrollee Discovery
4 Resource Discovery
5 Resource Found
Security Provisioning (Refer Security Spec)
6 Ownership transfer
Enrollee's Information Retrieval
7 Request Enrollee Information (easysetup/wificonf/devconf)
8 Response
WiFiConf
9 Request to update resources (wificonf)
10 Response (OK)
Connect
11 Request to connect (Wi-Fi)
12 Response (OK)
13 Disconnect Soft AP
14 Soft AP Destroy
15 Connect
16 Connect
Enrollee Discovery
17 Resource Discovery
18 Resource Found
Enrollee's Information Retrieval
19 Request Enrollee Information (easysetup/wificonf/devconf)
20 Response
Mediator
Enrollee
Enroller

Figure 3. Easy Setup Flow (Informative)

The example flow above undergoes security provisioning (step 6) during Easy Setup. Alternatively security provisioning can be done before Enrollee Discovery (steps 4 and 5) if preferred. Please refer to the OCF Security Specification for more information on the different scenarios.

9 Security

Wi-Fi Easy Setup security requirements are captured in the OCF Security Specification.

Annex A (normative)

Resource Type definitions

A.1 List of Resource Type definitions

Table 7 contains the list of defined resources in this specification.

Table 7. Alphabetized list of resources

Friendly Name (informative)	Resource Type (rt)	Section
Easy Setup	“oic.r.easysetup”	A.2
Wi-Fi Configuration	“oic.r.wificonf”	A.3
Device Configuration	“oic.r.devconf”	A.4

A.2 Easy Setup Collection Baseline Interface

A.2.1 Introduction

Easy Setup resource stores useful information including current status of unboxing device and last error code which are produced in a process of easy setup. Note that, Easy Setup resource is a type of collection resource, which contains links to WiFiConf, DevConf resources and may additionally contain links to other resources.

A.2.2 Example URI

/example/EasySetupBaselineInterfaceResURI

A.2.3 Resource Type

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

A.2.4 RAML Definition

```
#%RAML 0.8

title: Easy Setup Resource
version: v0.0.3-20170611

traits:
 - interface-ll :
     queryParameters:

       if:
         enum: ["oic.if.ll"]

 - interface-baseline :
     queryParameters:

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

 - interface-all :
     queryParameters:

       if:
         enum: ["oic.if.baseline", "oic.if.ll", "oic.if.b"]
 - interface-batch :
   queryParameters:
    if:
     enum: ["oic.if.b"]


/example/EasySetupBaselineInterfaceResURI:
 description: |
  Easy Setup resource stores useful information including current status of
  unboxing device and last error code which are produced in a process of
  easy setup.
  Note that, Easy Setup resource is a type of collection resource, which
  contains links to WiFiConf, DevConf resources and may additionally contain
  links to other resources.

 is : ['interface-baseline']
 get:
  description: |
   Retrieve useful information during easy setup process :
    1
   A current status in easy setup process.
    2
   A last error code describing reason for failure occurred at the last
     time.

  responses :
   200:
    body:
     application/json:
      schema: |
       {
        "$schema": "http://json-schema.org/draft-04/schema#",
        "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights reserved.",
        "id": "http://www.openconnectivity.org/ocf-apis/core/schemas/oic.r.easysetup-schema.json#",
        "definitions": {
         "oic.r.easysetup": {
          "type": "object",
          "allOf": [
           {
            "$ref": "oic.collection-schema.json#/definitions/oic.collection"
           },
           {
            "properties": {
             "rt": {
              "type": "array",
              "minItems": 2,
              "maxItems": 2,
              "uniqueItems": true,
              "items": {
                "enum": ["oic.r.easysetup","oic.wk.col"]
              }
             },
             "ps": {
              "type": "integer",
              "enum": [0, 1, 2, 3],
              "description": "Indicates the easy setup status of the device. (0: Need to Setup, 1: Connecting to Enroller, 2: Connected to Enroller, 3: Failed to Connect to Enroller, 4~254: Reserved, 255: EOF)",
              "readOnly": true
             },
             "lec": {
              "type": "integer",
              "enum": [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 255],
                    "description": "Indicates a failure reason (0: NO error, 1: A given
SSID is not found, 2: Wi-Fi's password is wrong, 3: IP address is not allocated, 4: No internet
connection, 5: Timeout, 6: Wi-Fi Auth Type is not supported by the Enrollee, 7: Wi-Fi Encryption
Type is not supported by the Enrollee, 8: Wi-Fi Auth Type is wrong (failure while connecting to the
Enroller), 9: Wi-Fi Encryption Type is wrong (failure while connecting to the Enroller), 10~254:
Reserved, 255: Unknown error)",
                    "readOnly": true
                  },
                  "cn": {
                    "type": "array",
                    "description": "Indicates an array of connection types that trigger an
attempt to connect to the Enroller to start.",
                    "items": {
                      "type": "integer",
                      "description": "Connection type to attempt. (1 : Wi-Fi, 2 : other
entities / transports to be added in future (e.g. Connect to cloud / BLE))"
                    }
                  }
                },
                "required": ["ps", "lec", "cn"]
              }
            ]
          }
        },
        "type": "object",
        "allOf": [
          { "$ref": "oic.core-schema.json#/definitions/oic.core"},
          { "$ref": "#/definitions/oic.r.easysetup" }
        ]
      }

    example: |
      {
        "rt" : ["oic.r.easysetup", "oic.wk.col"],
        "if" : ["oic.if.ll", "oic.if.baseline", "oic.if.b"],
        "ps" : 0,
        "lec": 0,
        "cn": [1],
        "links": [
          {
            "href": "/EasySetupResURI",
            "rt": ["oic.r.easysetup", "oic.wk.col"],
            "if": ["oic.if.b"],
            "p":{"bm":3},
            "eps": [
              {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2}
            ],
            "rel":["self", "item"]
          },
          {
            "href": "/WiFiConfResURI",
            "rt":   ["oic.r.wificonf"],
            "if":   ["oic.if.baseline"],
            "p":{"bm":3},
            "eps": [
              {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2}
            ]
          },
          {
            "href": "/DevConfResURI",
            "rt":   ["oic.r.devconf"],
            "if":   ["oic.if.baseline"],
            "p":{"bm":3},
            "eps": [
              {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2}
            ]
          }
        ]
      }
```

523 ## A.2.5 Property Definition

Property name	Value type	Mandatory	Access mode	Description
rt	array: see schema		Read Write	
ps	integer	yes	Read Write	Indicates the easy setup status of the device. (0: Need to Setup, 1: Connecting to Enroller, 2: Connected to Enroller, 3: Failed to Connect to Enroller, 4~254: Reserved, 255: EOF)
lec	integer	yes	Read Write	Indicates a failure reason (0: NO error, 1: A given SSID is not found, 2: Wi-Fi's password is wrong, 3: IP address is not allocated, 4: No internet connection, 5: Timeout, 6: Wi-Fi Auth Type is not supported by the Enrollee, 7: Wi-Fi Encryption Type is not supported by the Enrollee, 8: Wi-Fi Auth Type is wrong (failure while connecting to the Enroller), 9: Wi-Fi Encryption Type is wrong (failure while connecting to the Enroller), 10~254: Reserved, 255: Unknown error)
cn	array: see schema	yes	Read Write	Indicates an array of connection types that trigger an attempt to

				connect to the Enroller to start.

A.2.6 CRUDN behavior

Resource	Create	Read	Update	Delete	Notify
/example/EasySetupBaselineInterfaceResURI		get			

A.3 Wi-Fi Configuration Resource Baseline Interface

A.3.1 Introduction

WiFiConf resource stores essential information to help an unboxing device to connect to an existing Wi-Fi AP.

A.3.2 Example URI

/example/WiFiConfBaselineInterfaceResURI

A.3.3 Resource Type

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

A.3.4 RAML Definition

```
#%RAML 0.8
title: Wi-Fi Configuration Resource
version: v0.0.3-20170611
traits:
 - interface-rw :
     queryParameters:
       if:
         enum: ["oic.if.rw"]
 - interface-baseline :
     queryParameters:
       if:
         enum: ["oic.if.baseline"]
 - interface-all :
     queryParameters:
       if:
         enum: ["oic.if.baseline", "oic.if.rw"]


/example/WiFiConfBaselineInterfaceResURI:
  description: |
    WiFiConf resource stores essential information to help an unboxing device
    to connect to an existing Wi-Fi AP.

  is : ['interface-baseline']
  get:
    description: |
      Retrieve properties of WiFiConf resource.
      The information includes :
      1
      Wi-Fi SSID and password
      2
      Wi-Fi Security type (i.e
      auth type and encription type)
      3
      Wi-Fi hardware capability (i.e
      supported frequencies, modes,
        auth types and encryption types)

    responses :
      200:
        body:
          application/json:
            schema: |
              {
                "$schema": "http://json-schema.org/draft-04/schema#",
                "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
reserved.",
                "id": "http://www.openconnectivity.org/ocf-apis/core/schemas/oic.r.wificonf-
schema.json#",
                "definitions": {
                  "oic.r.wificonf": {
                    "type": "object",
                    "properties": {
                      "swmt": {
                        "type": "array",
                        "description": "Indicates supported Wi-Fi mode types. It can be multiple",
                        "readOnly": true,
                        "items":
                          {
                            "type": "string",
                            "enum": ["A","B","G","N","AC"],
                            "description": "Supported Wi-Fi Mode Type."
                          }
                      },
                      "swf": {
                        "type": "array",
                        "description": "Indicates Supported Wi-Fi frequencies by the Enrollee. Can
be multiple. Valid values are ('2.4G', '5G')",
                        "readOnly": true,
                        "items":
                          {
                            "type": "string",
                            "pattern": "^(2\\.4|5)G$"
                          }
                      },
                      "tnn": {
                        "type": "string",
                        "description": "Indicates Target Network Name (SSID of Wi-Fi AP)",
                        "pattern": "^.*$"
                      },
                      "cd": {
                        "type": "string",
                        "description": "Indicates credential information of Wi-Fi AP",
                        "pattern": "^.*$"
                      },
                      "wat": {
                        "type": "string",
                        "enum": ["None", "WEP", "WPA_PSK", "WPA2_PSK"],
                        "description": "Indicates Wi-Fi Auth Type"
                      },
                      "wet": {
                        "type": "string",
                        "enum": ["None", "WEP_64", "WEP_128", "TKIP", "AES", "TKIP_AES"],
                        "description": "Indicates Wi-Fi Encryption Type"
                      },
                      "swat": {
                        "type": "array",
                        "description": "Indicates supported Wi-Fi Auth types. It can be multiple",
                        "readOnly": true,
                        "items":
                          {
                            "type": "string",
                            "enum": ["None", "WEP", "WPA_PSK", "WPA2_PSK"],
                            "description": "Indicates Wi-Fi Auth Type"
                          }
                      },
                      "swet": {
                        "type": "array",
                        "description": "Indicates supported Wi-Fi Encryption types. It can be
multiple",
                        "readOnly": true,
                        "items":
                          {
                            "type": "string",
                            "enum": ["None", "WEP_64", "WEP_128", "TKIP", "AES", "TKIP_AES"],
                            "description": "Indicates Wi-Fi Encryption Type"
                          }
                      }
                    },
                    "required":["swmt", "swf", "swat", "swet", "tnn", "wat", "wet"]
                  }
                },
                "type": "object",
                "allOf": [
                  { "$ref": "oic.core-schema.json#/definitions/oic.core"},
                  { "$ref": "#/definitions/oic.r.wificonf" }
                ]
              }

            example: |
              {
                "rt": ["oic.r.wificonf"],
                "swmt" : ["A", "B", "G"],
                "swf": ["2.4G", "5G"],
                "tnn": "Home_AP_SSID",
                "cd": "Home_AP_PWD",
                "wat": "WPA2_PSK",
                "wet": "TKIP",
                "swat": ["WPA_PSK", "WPA2_PSK"],
                "swet": ["TKIP", "AES", "TKIP_AES"]
              }

  post:
    description: |
      Deliver Wi-Fi AP's information for an unboxing device to connect to it.

    body:
      application/json:
        schema: |
          {
            "$schema": "http://json-schema.org/draft-v4/schema#",
            "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
reserved.",
            "id": "http://www.openconnectivity.org/ocf-apis/core/schemas/oic.r.wificonf-update-
schema.json#",
            "definitions": {
              "oic.r.wificonf": {
                "type": "object",
                "properties": {
                  "tnn": {
                    "type": "string",
                    "description": "Indicates Target Network Name (SSID of Wi-Fi AP)",
                    "pattern": "^.*$"
                  },
                  "cd": {
                    "type": "string",
                    "description": "Indicates credential information of Wi-Fi AP",
                    "pattern": "^.*$"
                  },
                  "wat": {
                    "enum": ["None", "WEP", "WPA_PSK", "WPA2_PSK"],
                    "description": "Indicates Wi-Fi Auth Type"
                  },
                  "wet": {
                    "enum": ["None", "WEP_64", "WEP_128", "TKIP", "AES", "TKIP_AES"],
                    "description": "Indicates Wi-Fi Encryption Type"
                  }
                },
                "required":["tnn", "wat", "wet"]
              }
            },
            "type": "object",
            "allOf": [
              { "$ref": "oic.core-schema.json#/definitions/oic.core"},
              { "$ref": "#/definitions/oic.r.wificonf" }
            ]
          }

        example: |
          {
            "tnn": "Home_AP_SSID",
            "cd": "Home_AP_PWD",
            "wat": "WPA2_PSK",
            "wet": "AES"
          }

    responses :
      200:
        body:
          application/json:
            schema: |
              {
                "$schema": "http://json-schema.org/draft-v4/schema#",
                "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights reserved.",
                "id": "http://www.openconnectivity.org/ocf-apis/core/schemas/oic.r.wificonf-update-schema.json#",
                "definitions": {
                  "oic.r.wificonf": {
                    "type": "object",
                    "properties": {
                      "tnn": {
                        "type": "string",
                        "description": "Indicates Target Network Name (SSID of Wi-Fi AP)",
                        "pattern": "^.*$"
                      },
                      "cd": {
                        "type": "string",
                        "description": "Indicates credential information of Wi-Fi AP",
                        "pattern": "^.*$"
                      },
                      "wat": {
                        "enum": ["None", "WEP", "WPA_PSK", "WPA2_PSK"],
                        "description": "Indicates Wi-Fi Auth Type"
                      },
                      "wet": {
                        "enum": ["None", "WEP_64", "WEP_128", "TKIP", "AES", "TKIP_AES"],
                        "description": "Indicates Wi-Fi Encryption Type"
                      }
                    },
                    "required":["tnn", "wat", "wet"]
                  }
                },
                "type": "object",
                "allOf": [
                  { "$ref": "oic.core-schema.json#/definitions/oic.core"},
                  { "$ref": "#/definitions/oic.r.wificonf" }
                ]
              }

            example: |
              {
                "tnn": "Home_AP_SSID",
                "cd": "Home_AP_PWD",
                "wat": "WPA2_PSK",
                "wet": "AES"
              }
```

A.3.5 Property Definition

Property name	Value type	Mandatory	Access mode	Description
tnn	string	yes	Read Write	Indicates Target Network Name (SSID of Wi-Fi AP)
swmt	array: see schema	yes	Read Only	Indicates supported Wi-Fi mode types. It can be multiple
swat	array: see schema	yes	Read Only	Indicates supported Wi-Fi Auth types. It can be multiple
cd	string		Read Write	Indicates credential information of Wi-Fi AP
swf	array: see schema	yes	Read Only	Indicates Supported Wi-Fi frequencies by the Enrollee. Can be multiple. Valid values are ('2.4G', '5G')
wet	string	yes	Read Write	Indicates Wi-Fi Encryption Type
wat	string	yes	Read Write	Indicates Wi-Fi Auth Type
swet	array: see schema	yes	Read Only	Indicates supported Wi-Fi Encryption types. It can be multiple

A.3.6 CRUDN behavior

Resource	Create	Read	Update	Delete	Notify
/example/WiFiConfBaselineInterfaceResURI		get	post		

A.4 Device Configuration

A.4.1 Introduction

Device configuration resource stores a preference of device settings like device name. Vender-specfic information can be added to the resource.

A.4.2 Example URI

/example/DevConfResURI

A.4.3 Resource Type

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

A.4.4 RAML Definition

```
#%RAML 0.8
title: Device Configuration Resource
version: v0.0.2-20170604
traits:
 - interface :
     queryParameters:
       if:
         enum: ["oic.if.baseline", "oic.if.r"]


/example/DevConfResURI:
  description: |
    Device configuration resource stores a preference of device settings like
    device name
    Vender-specfic information can be added to the resource.

  is : ['interface']
  get:
    description: |
      Retrieve various settings regarding to device-specific settings
      1
      Device name (human-friendly name to be detected by mediator during
         easy setup)

    responses :
      200:
        body:
          application/json:
            schema: |
              {
                "$schema": "http://json-schema.org/draft-04/schema#",
                "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
reserved.",
                "id": "http://www.openconnectivity.org/ocf-apis/core/schemas/oic.r.devconf-
schema.json#",
                "definitions": {
                  "oic.r.devconf": {
                    "type": "object",
                    "oneOf": [
                      {
                        "properties": {
                          "dn": {
                            "type": "string",
                            "description": "Indicates a pre-configured device name in language
indicated by 'dl' in /oic/con;  presented by enrollee device to mediator device during easy-setup
process",
                            "pattern": "^.*$",
                            "readOnly": true
                          }
                        },
                        "required":["dn"]
                      },
                      {
                        "properties": {
                          "dn": {
                            "type": "array",
                            "items": {
                              "type": "object",
                              "properties": {
                                "language": {
                                  "$ref": "oic.types-schema.json#/definitions/language-tag",
                                  "readOnly": true,
                                  "description": "An RFC 5646 language tag."
                                },
                                "value": {
                                  "type": "string",
                                  "description": "Pre-configured device name in the indicated
language.",
                                  "pattern": "^.*$",
                                  "readOnly": true
                                }
                              }
                            },
                            "minItems" : 1,
                            "readOnly": true,
                            "description": "Localized device name."
                          }
                        },
                        "required": ["dn"]
                      }
                    ]
                  }
                },
                "type": "object",
                "allOf": [
                  { "$ref": "oic.core-schema.json#/definitions/oic.core"},
                  { "$ref": "#/definitions/oic.r.devconf" }
                ]
              }

            example: |
              {
                "rt": ["oic.r.devconf"],
                "dn" : "My Refrigerator"
              }
```

A.4.5 Property Definition

Property name	Value type	Mandatory	Access mode	Description
dn	array: see schema	yes	Read Only	Localized device name.

A.4.6 CRUDN behavior

Resource	Create	Read	Update	Delete	Notify
/example/DevConfResURI		get			

Annex B(informative)

Swagger2.0 definitions

B.1 Device Configuration

B.1.1 Introduction

Device configuration resource stores a preference of device settings like device name. Vender-specfic information can be added to the resource. Retrieve various settings regarding to device-specific settings 1. Device name (human-friendly name to be detected by mediator during easy setup)

B.1.2 Wellknown URI

/example/DevConfResURI

B.1.3 Resource Type

The resource type (rt) is defined as: ['oic.r.devconf'].

B.1.4 Swagger2.0 Definition

```
{
  "swagger": "2.0",
  "info": {
    "title": "Device Configuration",
    "version": "v0.0.2-20170604",
    "license": {
      "name": "copyright 2016-2017 Open Connectivity Foundation, Inc. All rights reserved.",
      "x-description": "Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:\n       1.
Redistributions of source code must retain the above copyright notice, this list of conditions and
the following disclaimer.\n        2.  Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer in the documentation and/or
other materials provided with the distribution.\n\n        THIS SOFTWARE IS PROVIDED BY THE Open
Connectivity Foundation, INC. \"AS IS\" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR
WARRANTIES OF NON-INFRINGEMENT, ARE DISCLAIMED.\n        IN NO EVENT SHALL THE Open Connectivity
Foundation, INC. OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)\n        HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY
OF SUCH DAMAGE.\n"
    }
  },
  "schemes": ["http"],
  "consumes": ["application/json"],
  "produces": ["application/json"],
  "paths": {
    "/example/DevConfResURI" : {
      "get": {
        "description": "Device configuration resource stores a preference of device settings
like\ndevice name. Vender-specfic information can be added to the resource.\nRetrieve various
settings regarding to device-specific settings\n1. Device name (human-friendly name to be detected
by mediator during\n   easy setup)\n",
        "parameters": [
          {"$ref": "#/parameters/interface"}
        ],
        "responses": {
            "200": {
              "description" : "",
              "x-example":
                {
                  "rt": ["oic.r.devconf"],
                  "dn" : "My Refrigerator"
                }
                ,
              "schema": { "$ref": "#/definitions/DevConf" }
            }
        }
      }
    }
  },
  "parameters": {
    "interface" : {
      "in" : "query",
      "name" : "if",
      "type" : "string",
      "enum" : ["oic.if.baseline", "oic.if.r"]
    }
  },
  "definitions": {
    "DevConf" :
            {
        "oneOf": [
          {
            "properties": {
              "dn": {
                "description": "Indicates a pre-configured device name in language indicated by
'dl' in /oic/con;  presented by enrollee device to mediator device during easy-setup process",
                "pattern": "^.*$",
                "readOnly": true,
                "type": "string"
              }
            },
            "required": [
              "dn"
            ]
          },
          {
            "properties": {
              "dn": {
                "description": "Localized device name.",
                "items": {
                  "properties": {
                    "language": {
                      "description": "An RFC 5646 language tag.",
                      "pattern": "^[A-Za-z]{1,8}(-[A-Za-z0-9]{1,8})*$",
                      "readOnly": true,
                      "type": "string"
                    },
                    "value": {
                      "description": "Pre-configured device name in the indicated language.",
                      "pattern": "^.*$",
                      "readOnly": true,
                      "type": "string"
                    }
                  },
                  "type": "object"
                },
                "minItems": 1,
                "readOnly": true,
                "type": "array"
              }
            },
            "required": [
              "dn"
            ]
          }
        ],
        "type": "object"
      }

  }
}
```

B.1.5 Property Definition

Property name	Value type	Mandatory	Access mode	Description
dn	array: see schema	yes	Read Only	Localized device name.

B.1.6 CRUDN behaviour

Resource	Create	Read	Update	Delete	Notify
/example/DevConfResURI		get			

B.2 Easy Setup Collection

B.2.1 Introduction

Easy Setup resource stores useful information including current status of unboxing device and last error code which are produced in a process of easy setup.
Note that, Easy Setup resource is a type of collection resource, which contains links to WiFiConf, DevConf resources and may additionally contain links to other resources.
Retrieve useful information during easy setup process:
1. A current status in easy setup process.
2. A last error code describing reason for failure occurred at the last time.

B.2.2 Wellknown URI

/example/EasySetupBaselineInterfaceResURI

B.2.3 Resource Type

The resource type (rt) is defined as: ['oic.r.easysetup', 'oic.wk.col'].

B.2.4 Swagger2.0 Definition

```
{
  "swagger": "2.0",
  "info": {
    "title": "Easy Setup Collection Batch Interface",
    "version": "v0.0.3-20170611",
    "license": {
      "name": "copyright 2016-2017 Open Connectivity Foundation, Inc. All rights reserved.",
      "x-description": "Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:\n       1.
Redistributions of source code must retain the above copyright notice, this list of conditions and
the following disclaimer.\n        2.  Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer in the documentation and/or
other materials provided with the distribution.\n\n        THIS SOFTWARE IS PROVIDED BY THE Open
Connectivity Foundation, INC. \"AS IS\" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR
WARRANTIES OF NON-INFRINGEMENT, ARE DISCLAIMED.\n        IN NO EVENT SHALL THE Open Connectivity
Foundation, INC. OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)\n        HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY
OF SUCH DAMAGE.\n"
    }
  },
  "schemes": ["http"],
  "consumes": ["application/json"],
  "produces": ["application/json"],
  "paths": {
    "/example/EasySetupBatchInterfaceResURI" : {
      "get": {
        "description": "Easy Setup resource stores useful information including current status
```

```
of\nunboxing device and last error code which are produced in a process of\neasy setup.\nNote that,
Easy Setup resource is a type of collection resource, which\ncontains links to WiFiConf, DevConf
resources and may additionally contain\nlinks to other resources.\nRetrieve useful information
during easy setup process :\n1. A current status in easy setup process.\n2. A last error code
describing reason for failure occurred at the last\n   time.\n",
        "parameters": [
          {"$ref": "#/parameters/interface-batch"}
        ],
        "responses": {
            "200": {
              "description" : "",
              "x-example":
                [
                  {
                    "href": "/EasySetupResURI",
                    "rep":{
                      "ps" : 0,
                      "lec": 0,
                      "cn": [1]
                    }
                  },
                  {
                    "href": "/WiFiConfResURI",
                    "rep":{
                      "swmt" : ["A", "B", "G"],
                      "swf": ["2.4G", "5G"],
                      "tnn": "Home_AP_SSID",
                      "cd": "Home_AP_PWD",
                      "wat": "WPA2_PSK",
                      "wet": "AES"
                    }
                  },
                  {
                    "href": "/DevConfResURI",
                    "rep":{
                      "dn" : "My Refrigerator"
                    }
                  }
                ]
                ,
              "schema": { "$ref": "#/definitions/sbatch" }
            }
        }
      },
      "post": {
        "description": "Able to deliver Wi-Fi, Device configuration and other
configuration\ninformation in a batch by utilizing 'batch' interface.\nIf you want to deliver Wi-Fi
and Device configuration information in a batch,\nyou can write all properties you want to send
with a 'batch' interface.\nThe below example is the case to send Easy Setup and Wi-Fi
configuration\n(i.e. connection type, target network, auth type information) in a batch.\n",
        "parameters": [
          {"$ref": "#/parameters/interface-batch"},
          {
            "name": "body",
            "in": "body",
            "required": true,
            "schema": { "$ref": "#/definitions/sbatch-update" },
            "x-example":
              [
                {
                  "href": "/EasySetupResURI",
                  "rep":{
                    "cn": [1]
                  }
                },
                {
                  "href": "/WiFiConfResURI",
                  "rep":{
                    "tnn": "Home_AP_SSID",
                    "cd": "Home_AP_PWD",
                    "wat": "WPA2_PSK",
                    "wet": "AES"
                  }
                }
              ]
          }
        ],
        "responses": {
            "200": {
              "description" : "",
              "x-example":
                [
                  {
                    "href": "/EasySetupResURI",
                    "rep" : {
                      "ps" : 0,
                      "lec": 0,
                      "cn": [1]
                    }
                  },
                  {
                    "href": "/WiFiConfResURI",
                    "rep" : {
                      "swmt" : ["A", "B", "G"],
                      "swf": ["2.4G", "5G"],
                      "tnn": "Home_AP_SSID",
                      "cd": "Home_AP_PWD",
                      "wat": "WPA2_PSK",
                      "wet": "AES"
                    }
                  },
                  {
                    "href": "/DevConfResURI",
                    "rep" : {
                      "dn" : "My Refrigerator"
                    }
                  }
                ]
                ,
              "schema": { "$ref": "#/definitions/sbatch" }
            }
        }
      }
    },
    "/example/EasySetupLLInterfaceResURI" : {
      "get": {
        "description": "Easy Setup resource stores useful information including current status
of\nunboxing device and last error code which are produced in a process of\neasy setup.\nNote that,
Easy Setup resource is a type of collection resource, which\ncontains links to WiFiConf, DevConf
resources and may additionally contain\nlinks to other resources.\nRetrieve useful information
during easy setup process :\n1. A current status in easy setup process.\n2. A last error code
describing reason for failure occurred at the last\n   time.\n",
        "parameters": [
          {"$ref": "#/parameters/interface-ll"}
        ],
        "responses": {
            "200": {
              "description" : "",
              "x-example":
                [
                  {
                    "href": "/EasySetupResURI",
                    "rt": ["oic.r.easysetup", "oic.wk.col"],
                    "if": ["oic.if.b"],
                    "p":{"bm":3},
                    "eps": [
                      {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2}
                    ],
                    "rel":["self", "item"]
                  },
                  {
                    "href": "/WiFiConfResURI",
                    "rt":   ["oic.r.wificonf"],
                    "if":   ["oic.if.baseline"],
                    "p":{"bm":3},
                    "eps": [
                      {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2}
                    ]
                  },
                  {
                    "href": "/DevConfResURI",
                    "rt":   ["oic.r.devconf"],
                    "if":   ["oic.if.baseline"],
                    "p":{"bm":3},
                    "eps": [
                      {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2}
                    ]
                  }
                ]
                ,
              "schema": { "$ref": "#/definitions/slinks" }
            }
        }
      }
    },
    "/example/EasySetupBaselineInterfaceResURI" : {
      "get": {
        "description": "Easy Setup resource stores useful information including current status
of\nunboxing device and last error code which are produced in a process of\neasy setup.\nNote that,
Easy Setup resource is a type of collection resource, which\ncontains links to WiFiConf, DevConf
resources and may additionally contain\nlinks to other resources.\nRetrieve useful information
during easy setup process :\n  1. A current status in easy setup process.\n  2. A last error code
describing reason for failure occurred at the last\n     time.\n",
        "parameters": [
          {"$ref": "#/parameters/interface-baseline"}
        ],
        "responses": {
            "200": {
              "description" : "",
              "x-example":
                {
                  "rt" : ["oic.r.easysetup", "oic.wk.col"],
                  "if" : ["oic.if.ll", "oic.if.baseline", "oic.if.b"],
                  "ps" : 0,
                  "lec": 0,
                  "cn": [1],
                  "links": [
                    {
                      "href": "/EasySetupResURI",
                      "rt": ["oic.r.easysetup", "oic.wk.col"],
                      "if": ["oic.if.b"],
                      "p":{"bm":3},
                      "eps": [
                        {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2}
                      ],
                      "rel":["self", "item"]
                    },
                    {
                      "href": "/WiFiConfResURI",
                      "rt":   ["oic.r.wificonf"],
                      "if":   ["oic.if.baseline"],
                      "p":{"bm":3},
                      "eps": [
                        {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2}
                      ]
                    },
                    {
                      "href": "/DevConfResURI",
                      "rt":   ["oic.r.devconf"],
                      "if":   ["oic.if.baseline"],
                      "p":{"bm":3},
                      "eps": [
                        {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2}
                      ]
                    }
                  ]
                }
                ,
              "schema": { "$ref": "#/definitions/EasySetup" }
            }
        }
      }
    }
  },
  "parameters": {
    "interface-ll" : {
      "in" : "query",
      "name" : "if",
      "type" : "string",
      "enum" : ["oic.if.ll"]
    },
    "interface-baseline" : {
      "in" : "query",
      "name" : "if",
      "type" : "string",
      "enum" : ["oic.if.baseline"]
    },
    "interface-all" : {
      "in" : "query",
      "name" : "if",
      "type" : "string",
      "enum" : ["oic.if.baseline", "oic.if.ll", "oic.if.b"]
    },
    "interface-batch" : {
      "in" : "query",
      "name" : "if",
      "type" : "string",
      "enum" : ["oic.if.b"]
    }
  },
  "definitions": {
    "sbatch" :
            {
        "items": {
          "additionalProperties": true,
          "properties": {
            "href": {
              "description": "URI of the target resource relative assuming the collection URI as anchor",
              "format": "uri",
              "maxLength": 256,
              "type": "string"
            },
            "rep": {
              "oneOf": [
                {
                  "description": "The response payload from a single resource",
                  "type": "object"
                },
                {
                  "description": " The response payload from a collection (batch) resource",
                  "type": "array"
                }
              ]
            }
          },
          "required": [
            "href",
            "rep"
          ],
          "type": "object"
        },
        "minItems": 1,
        "type": "array"
```

```
   }

  ,
  "sbatch-update" :
      {
    "description": "array of resource representations to apply to the batch collection, using
href to indicate which resource(s) in the batch to update. If the href property is empty,
effectively making the URI reference to the collection itself, the representation is to be applied
to all resources in the batch",
    "items": {
     "additionalProperties": true,
     "properties": {
      "href": {
       "description": "URI of the target resource relative assuming the collection URI as
anchor",
       "format": "uri",
       "maxLength": 256,
       "type": "string"
      },
      "rep": {
       "oneOf": [
        {
         "description": "The response payload from a single resource",
         "type": "object"
        },
        {
         "description": " The response payload from a collection (batch) resource",
         "type": "array"
        }
       ]
      }
     },
     "required": [
      "href",
      "rep"
     ],
     "type": "object"
    },
    "minItems": 1,
    "type": "array"
   }

  ,
  "slinks" :
      {
    "description": "All forms of links in a collection",
    "oneOf": [
     {
      "description": "A set (array) of simple or individual OIC Links. In addition to
properties required for an OIC Link, the identifier for that link in this set is also required",
      "items": {
       "properties": {
        "anchor": {
         "description": "This is used to override the context URI e.g. override the URI of
the containing collection",
         "format": "uri",
         "maxLength": 256,
         "type": "string"
        },
        "di": {
         "description": "Unique identifier for device (UUID)",
         "pattern": "^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-
F0-9]{12}$",
         "type": "string"
        },
        "eps": {
         "description": "the Endpoint information of the target Resource",
         "items": {
          "properties": {
           "ep": {
            "description": "URI with Transport Protocol Suites + Endpoint Locator as
specified in 10.2.1",
                        "format": "uri",
                        "type": "string"
                      },
                      "pri": {
                        "description": "The priority among multiple Endpoints as specified in
10.2.3",
                        "minimum": 1,
                        "type": "integer"
                      }
                    },
                    "type": "object"
                  },
                  "type": "array"
                },
                "href": {
                  "description": "This is the target URI, it can be specified as a Relative
Reference or fully-qualified URI. Relative Reference should be used along with the di parameter to
make it unique.",
                  "format": "uri",
                  "maxLength": 256,
                  "type": "string"
                },
                "if": {
                  "description": "The interface set supported by this resource",
                  "items": {
                    "enum": [
                      "oic.if.baseline",
                      "oic.if.ll",
                      "oic.if.b",
                      "oic.if.rw",
                      "oic.if.r",
                      "oic.if.a",
                      "oic.if.s"
                    ],
                    "type": "string"
                  },
                  "minItems": 1,
                  "type": "array"
                },
                "ins": {
                  "description": "The instance identifier for this web link in an array of web
links - used in collections",
                  "oneOf": [
                    {
                      "description": "An ordinal number that is not repeated - must be unique in
the collection context",
                      "type": "integer"
                    },
                    {
                      "description": "Any unique string including a URI",
                      "format": "uri",
                      "maxLength": 256,
                      "type": "string"
                    },
                    {
                      "description": "Unique identifier (UUID)",
                      "pattern": "^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-
fA-F0-9]{12}$",
                      "type": "string"
                    }
                  ]
                },
                "p": {
                  "description": "Specifies the framework policies on the Resource referenced by
the target URI",
                  "properties": {
                    "bm": {
                      "description": "Specifies the framework policies on the Resource referenced
by the target URI for e.g. observable and discoverable",
                      "type": "integer"
                    }
                  },
                  "required": [
                    "bm"
                  ],
                  "type": "object"
                },
                "rel": {
                  "description": "The relation of the target URI referenced by the link to the
context URI",
                  "oneOf": [
                    {
                      "default": [
                        "hosts"
                      ],
                      "items": {
                        "maxLength": 64,
                        "type": "string"
                      },
                      "minItems": 1,
                      "type": "array"
                    },
                    {
                      "default": "hosts",
                      "maxLength": 64,
                      "type": "string"
                    }
                  ]
                },
                "rt": {
                  "description": "Resource Type",
                  "items": {
                    "maxLength": 64,
                    "type": "string"
                  },
                  "minItems": 1,
                  "type": "array"
                },
                "title": {
                  "description": "A title for the link relation. Can be used by the UI to provide a
context",
                  "maxLength": 64,
                  "type": "string"
                },
                "type": {
                  "default": "application/cbor",
                  "description": "A hint at the representation of the resource referenced by the
target URI. This represents the media types that are used for both accepting and emitting",
                  "items": {
                    "maxLength": 64,
                    "type": "string"
                  },
                  "minItems": 1,
                  "type": "array"
                }
              },
              "required": [
                "href",
                "rt",
                "if"
              ],
              "type": "object"
            },
            "type": "array"
          }
        ]
      }

    ,
    "EasySetup" :
            {
```

```
        "description": "A collection is a set (array) of tagged-link or set (array) of simple links
along with additional properties to describe the collection itself",
        "properties": {
          "cn": {
            "description": "Indicates an array of connection types that trigger an attempt to
connect to the Enroller to start.",
            "items": {
              "description": "Connection type to attempt. (1 : Wi-Fi, 2 : other entities /
transports to be added in future (e.g. Connect to cloud / BLE))",
              "type": "integer"
            },
            "type": "array"
          },
          "di": {
            "description": "The device ID which is an UUIDv4 string; used for backward
compatibility with Spec A definition of /oic/res",
            "pattern": "^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-
9]{12}$",
            "type": "string"
          },
          "drel": {
            "description": "When specified this is the default relationship to use when an OIC Link
does not specify an explicit relationship with *rel* parameter",
            "type": "string"
          },
          "id": {
            "anyOf": [
              {
                "description": "A number that is unique to that collection; like an ordinal number
that is not repeated",
                "type": "integer"
              },
              {
                "description": "A unique string that could be a hash or similarly unique",
                "type": "string"
              },
              {
                "description": "A unique string that could be a UUIDv4",
                "pattern": "^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-
9]{12}$",
                "type": "string"
              }
            ],
            "description": "ID for the collection. Can be an value that is unique to the use
context or a UUIDv4"
          },
          "lec": {
            "description": "Indicates a failure reason (0: NO error, 1: A given SSID is not found,
2: Wi-Fi's password is wrong, 3: IP address is not allocated, 4: No internet connection, 5:
Timeout, 6: Wi-Fi Auth Type is not supported by the Enrollee, 7: Wi-Fi Encryption Type is not
supported by the Enrollee, 8: Wi-Fi Auth Type is wrong (failure while connecting to the Enroller),
9: Wi-Fi Encryption Type is wrong (failure while connecting to the Enroller), 10~254: Reserved,
255: Unknown error)",
            "enum": [
              0,
              1,
              2,
              3,
              4,
              5,
              6,
              7,
              8,
              9,
              255
            ],
            "readOnly": true,
            "type": "integer"
          },
          "links": {
            "description": "All forms of links in a collection",
            "oneOf": [
              {
                "description": "A set (array) of simple or individual OIC Links. In addition to
properties required for an OIC Link, the identifier for that link in this set is also required",
                "items": {
                  "properties": {
                    "anchor": {
                      "description": "This is used to override the context URI e.g. override the
URI of the containing collection",
                      "format": "uri",
                      "maxLength": 256,
                      "type": "string"
                    },
                    "di": {
                      "description": "Unique identifier for device (UUID)",
                      "pattern": "^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-
fA-F0-9]{12}$",
                      "type": "string"
                    },
                    "eps": {
                      "description": "the Endpoint information of the target Resource",
                      "items": {
                        "properties": {
                          "ep": {
                            "description": "URI with Transport Protocol Suites + Endpoint Locator
as specified in 10.2.1",
                            "format": "uri",
                            "type": "string"
                          },
                          "pri": {
                            "description": "The priority among multiple Endpoints as specified in
10.2.3",
                            "minimum": 1,
                            "type": "integer"
                          }
                        },
                        "type": "object"
                      },
                      "type": "array"
                    },
                    "href": {
                      "description": "This is the target URI, it can be specified as a Relative
Reference or fully-qualified URI. Relative Reference should be used along with the di parameter to
make it unique.",
                      "format": "uri",
                      "maxLength": 256,
                      "type": "string"
                    },
                    "if": {
                      "description": "The interface set supported by this resource",
                      "items": {
                        "enum": [
                          "oic.if.baseline",
                          "oic.if.ll",
                          "oic.if.b",
                          "oic.if.rw",
                          "oic.if.r",
                          "oic.if.a",
                          "oic.if.s"
                        ],
                        "type": "string"
                      },
                      "minItems": 1,
                      "type": "array"
                    },
                    "ins": {
                      "description": "The instance identifier for this web link in an array of web
links - used in collections",
                      "oneOf": [
                        {
                          "description": "An ordinal number that is not repeated - must be unique
in the collection context",
                          "type": "integer"
                        },
                        {
                          "description": "Any unique string including a URI",
                          "format": "uri",
                          "maxLength": 256,
                          "type": "string"
                        },
                        {
                          "description": "Unique identifier (UUID)",
                          "pattern": "^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-
[a-fA-F0-9]{12}$",
                          "type": "string"
                        }
                      ]
                    },
                    "p": {
                      "description": "Specifies the framework policies on the Resource referenced
by the target URI",
                      "properties": {
                        "bm": {
                          "description": "Specifies the framework policies on the Resource
referenced by the target URI for e.g. observable and discoverable",
                          "type": "integer"
                        }
                      },
                      "required": [
                        "bm"
                      ],
                      "type": "object"
                    },
                    "rel": {
                      "description": "The relation of the target URI referenced by the link to the
context URI",
                      "oneOf": [
                        {
                          "default": [
                            "hosts"
                          ],
                          "items": {
                            "maxLength": 64,
                            "type": "string"
                          },
                          "minItems": 1,
                          "type": "array"
                        },
                        {
                          "default": "hosts",
                          "maxLength": 64,
                          "type": "string"
                        }
                      ]
                    },
                    "rt": {
                      "description": "Resource Type",
                      "items": {
                        "maxLength": 64,
                        "type": "string"
                      },
                      "minItems": 1,
                      "type": "array"
                    },
                    "title": {
                      "description": "A title for the link relation. Can be used by the UI to
provide a context",
                      "maxLength": 64,
                      "type": "string"
                    },
                    "type": {
                      "default": "application/cbor",
                      "description": "A hint at the representation of the resource referenced by
the target URI. This represents the media types that are used for both accepting and emitting",
                      "items": {
                        "maxLength": 64,
                        "type": "string"
                      },
                      "minItems": 1,
                      "type": "array"
                    }
                  },
                  "required": [
                    "href",
                    "rt",
                    "if"
                  ],
                  "type": "object"
                },
                "type": "array"
              }
            ]
          },
          "ps": {
            "description": "Indicates the easy setup status of the device. (0: Need to Setup, 1:
Connecting to Enroller, 2: Connected to Enroller, 3: Failed to Connect to Enroller, 4~254:
Reserved, 255: EOF)",
            "enum": [
              0,
              1,
              2,
              3
            ],
            "readOnly": true,
            "type": "integer"
          },
          "rt": {
            "items": {
              "enum": [
                "oic.r.easysetup",
                "oic.wk.col"
              ]
            },
            "maxItems": 2,
            "minItems": 2,
            "type": "array",
            "uniqueItems": true
          },
          "rts": {
            "description": "Defines the list of allowable resource types (for Target and anchors)
in links included in the collection; new links being created can only be from this list",
            "items": {
              "maxLength": 64,
              "type": "string"
            },
            "minItems": 1,
            "readOnly": true,
            "type": "array"
          }
        },
        "required": [
          "ps",
          "lec",
          "cn"
        ],
        "type": "object"
      }

  }
}
```

1852 ### B.2.5 Property Definition

Property name	Value type	Mandatory	Access mode	Description
rep	multiple types: see schema	yes		
href	string	yes		URI of the target resource relative assuming the collection URI as anchor
cn	array: see schema	yes		Indicates an array of connection types that trigger an attempt to connect to the Enroller to start.
di	string			The device ID which is an UUIDv4 string; used for backward compatibility with Spec A definition of /oic/res
lec	integer	yes	Read Only	Indicates a failure reason (0: NO error, 1: A given SSID is not found, 2: Wi-Fi's password is wrong, 3: IP address is not allocated, 4: No internet connection, 5: Timeout, 6: Wi-Fi Auth Type is not supported by the Enrollee, 7: Wi-Fi Encryption Type is not supported by the Enrollee, 8: Wi-Fi Auth Type is wrong (failure while connecting to the Enroller), 9: Wi-Fi Encryption Type is wrong (failure while connecting to the Enroller), 10~254: Reserved, 255: Unknown error)

drel	string			When specified this is the default relationship to use when an OIC Link does not specify an explicit relationship with *rel* parameter
id	multiple types: see schema			ID for the collection. Can be an value that is unique to the use context or a UUIDv4
rt	array: see schema			
rts	array: see schema		Read Only	Defines the list of allowable resource types (for Target and anchors) in links included in the collection; new links being created can only be from this list
ps	integer	yes	Read Only	Indicates the easy setup status of the device. (0: Need to Setup, 1: Connecting to Enroller, 2: Connected to Enroller, 3: Failed to Connect to Enroller, 4~254: Reserved, 255: EOF)
links	multiple types: see schema			All forms of links in a collection
anchor	string			This is used to override the context URI e.g. override the URI of the containing collection
di	string			Unique identifier for device (UUID)
title	string			A title for the link relation. Can be used by the UI to provide a context

p	object: see schema			Specifies the framework policies on the Resource referenced by the target URI
rel	multiple types: see schema			The relation of the target URI referenced by the link to the context URI
rt	array: see schema	yes		Resource Type
ins	multiple types: see schema			The instance identifier for this web link in an array of web links - used in collections
eps	array: see schema			the Endpoint information of the target Resource
if	array: see schema	yes		The interface set supported by this resource
type	array: see schema			A hint at the representation of the resource referenced by the target URI. This represents the media types that are used for both accepting and emitting
href	string	yes		This is the target URI, it can be specified as a Relative Reference or fully-qualified URI. Relative Reference should be used along with the di parameter to make it unique.
rep	multiple types: see schema	yes		
href	string	yes		URI of the target resource relative assuming the collection URI as anchor

B.2.6 CRUDN behaviour

Resource	Create	Read	Update	Delete	Notify
/example/EasySetupBaselineInterfaceResURI		get			

B.3 Wi-Fi Configuration Resource

B.3.1 Introduction

WiFiConf resource stores essential information to help an unboxing device
to connect to an existing Wi-Fi AP.
Retrieve properties of WiFiConf resource.
The information includes:
1. Wi-Fi SSID and password
2. Wi-Fi Security type (i.e. auth type and encription type)
3. Wi-Fi hardware capability (i.e. supported frequencies, modes,
 auth types and encryption types)

B.3.2 Wellknown URI

/example/WiFiConfBaselineInterfaceResURI

B.3.3 Resource Type

The resource type (rt) is defined as: ['oic.r.wificonf'].

B.3.4 Swagger2.0 Definition

```
{
  "swagger": "2.0",
  "info": {
    "title": "Wi-Fi Configuration Resource Baseline Interface",
    "version": "v0.0.3-20170611",
    "license": {
      "name": "copyright 2016-2017 Open Connectivity Foundation, Inc. All rights reserved.",
      "x-description": "Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:\n        1.
Redistributions of source code must retain the above copyright notice, this list of conditions and
the following disclaimer.\n        2.  Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer in the documentation and/or
other materials provided with the distribution.\n\n        THIS SOFTWARE IS PROVIDED BY THE Open
Connectivity Foundation, INC. \"AS IS\" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR
WARRANTIES OF NON-INFRINGEMENT, ARE DISCLAIMED.\n        IN NO EVENT SHALL THE Open Connectivity
Foundation, INC. OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)\n        HOWEVER CAUSED AND
ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY
OF SUCH DAMAGE.\n"
    }
  },
  "schemes": ["http"],
  "consumes": ["application/json"],
  "produces": ["application/json"],
  "paths": {
    "/example/WiFiConfBaselineInterfaceResURI" : {
      "get": {
        "description": "WiFiConf resource stores essential information to help an unboxing
device\nto connect to an existing Wi-Fi AP.\nRetrieve properties of WiFiConf resource.\nThe
information includes :\n1. Wi-Fi SSID and password\n2. Wi-Fi Security type (i.e. auth type and
encription type)\n3. Wi-Fi hardware capability (i.e. supported frequencies, modes,\n  auth types
and encryption types)\n",
        "parameters": [
          {"$ref": "#/parameters/interface-baseline"}
        ],
        "responses": {
            "200": {
              "description" : "",
              "x-example":
                {
                  "rt": ["oic.r.wificonf"],
                  "swmt" : ["A", "B", "G"],
                  "swf": ["2.4G", "5G"],
                  "tnn": "Home_AP_SSID",
                  "cd": "Home_AP_PWD",
                  "wat": "WPA2_PSK",
                  "wet": "TKIP",
                  "swat": ["WPA_PSK", "WPA2_PSK"],
                  "swet": ["TKIP", "AES", "TKIP_AES"]
                }
                ,
              "schema": { "$ref": "#/definitions/WiFiConf" }
            }
        }
      },
      "post": {
        "description": "Deliver Wi-Fi AP's information for an unboxing device to connect to it.\n",
        "parameters": [
          {"$ref": "#/parameters/interface-baseline"},
          {
            "name": "body",
            "in": "body",
            "required": true,
            "schema": { "$ref": "#/definitions/WiFiConfUpdate" },
            "x-example":
              {
                "tnn": "Home_AP_SSID",
                "cd": "Home_AP_PWD",
                "wat": "WPA2_PSK",
                "wet": "AES"
              }
          }
        ],
        "responses": {
            "200": {
              "description" : "",
              "x-example":
                {
                  "tnn": "Home_AP_SSID",
                  "cd": "Home_AP_PWD",
                  "wat": "WPA2_PSK",
                  "wet": "AES"
                }
                ,
              "schema": { "$ref": "#/definitions/WiFiConfUpdate" }
            }
        }
      }
    },
    "/example/WiFiConfRWInterfaceResURI" : {
      "get": {
        "description": "WiFiConf resource stores essential information to help an unboxing
device\nto connect to an existing Wi-Fi AP.\nRetrieve properties of WiFiConf resource that can be
updated by a client.\n",
        "parameters": [
          {"$ref": "#/parameters/interface-rw"}
        ],
        "responses": {
            "200": {
              "description" : "",
              "x-example":
                {
                  "tnn": "Home_AP_SSID",
                  "cd": "Home_AP_PWD",
                  "wat": "WPA2_PSK",
                  "wet": "AES"
                }
                ,
              "schema": { "$ref": "#/definitions/WiFiConfUpdate" }
            }
        }
      },
      "post": {
        "description": "Deliver Wi-Fi AP's information for an unboxing device to connect to it.\n",
        "parameters": [
          {"$ref": "#/parameters/interface-rw"},
          {
            "name": "body",
            "in": "body",
            "required": true,
            "schema": { "$ref": "#/definitions/WiFiConfUpdate" },
            "x-example":
              {
                "tnn": "Home_AP_SSID",
                "cd": "Home_AP_PWD",
                "wat": "WPA2_PSK",
                "wet": "AES"
              }
          }
        ],
        "responses": {
            "200": {
              "description" : "",
              "x-example":
                {
                  "tnn": "Home_AP_SSID",
                  "cd": "Home_AP_PWD",
                  "wat": "WPA2_PSK",
                  "wet": "AES"
                }
                ,
              "schema": { "$ref": "#/definitions/WiFiConfUpdate" }
            }
        }
      }
    }
  },
  "parameters": {
    "interface-rw" : {
      "in" : "query",
      "name" : "if",
      "type" : "string",
      "enum" : ["oic.if.rw"]
    },
    "interface-baseline" : {
      "in" : "query",
      "name" : "if",
      "type" : "string",
      "enum" : ["oic.if.baseline"]
    },
    "interface-all" : {
      "in" : "query",
      "name" : "if",
      "type" : "string",
      "enum" : ["oic.if.baseline", "oic.if.rw"]
    }
  },
  "definitions": {
    "WiFiConf" :
            {
        "properties": {
          "cd": {
            "description": "Indicates credential information of Wi-Fi AP",
            "pattern": "^.*$",
            "type": "string"
          },
          "swat": {
            "description": "Indicates supported Wi-Fi Auth types. It can be multiple",
            "items": {
              "description": "Indicates Wi-Fi Auth Type",
              "enum": [
                "None",
                "WEP",
                "WPA_PSK",
                "WPA2_PSK"
              ],
              "type": "string"
            },
            "readOnly": true,
            "type": "array"
          },
          "swet": {
            "description": "Indicates supported Wi-Fi Encryption types. It can be multiple",
            "items": {
              "description": "Indicates Wi-Fi Encryption Type",
              "enum": [
                "None",
                "WEP_64",
                "WEP_128",
                "TKIP",
                "AES",
                "TKIP_AES"
              ],
              "type": "string"
            },
            "readOnly": true,
            "type": "array"
          },
          "swf": {
            "description": "Indicates Supported Wi-Fi frequencies by the Enrollee. Can be multiple.
Valid values are ('2.4G', '5G')",
            "items": {
              "pattern": "^(2\\.4|5)G$",
              "type": "string"
            },
            "readOnly": true,
            "type": "array"
          },
          "swmt": {
            "description": "Indicates supported Wi-Fi mode types. It can be multiple",
            "items": {
              "description": "Supported Wi-Fi Mode Type.",
              "enum": [
                "A",
                "B",
                "G",
                "N",
                "AC"
              ],
              "type": "string"
            },
            "readOnly": true,
            "type": "array"
          },
          "tnn": {
            "description": "Indicates Target Network Name (SSID of Wi-Fi AP)",
            "pattern": "^.*$",
            "type": "string"
          },
          "wat": {
            "description": "Indicates Wi-Fi Auth Type",
            "enum": [
              "None",
              "WEP",
              "WPA_PSK",
              "WPA2_PSK"
            ],
            "type": "string"
          },
          "wet": {
```

```
            "description": "Indicates Wi-Fi Encryption Type",
            "enum": [
              "None",
              "WEP_64",
              "WEP_128",
              "TKIP",
              "AES",
              "TKIP_AES"
            ],
            "type": "string"
          }
        },
        "required": [
          "swmt",
          "swf",
          "swat",
          "swet",
          "tnn",
          "wat",
          "wet"
        ],
        "type": "object"
      }

    ,
    "WiFiConfUpdate" :
            {
        "properties": {
          "cd": {
            "description": "Indicates credential information of Wi-Fi AP",
            "pattern": "^.*$",
            "type": "string"
          },
          "tnn": {
            "description": "Indicates Target Network Name (SSID of Wi-Fi AP)",
            "pattern": "^.*$",
            "type": "string"
          },
          "wat": {
            "description": "Indicates Wi-Fi Auth Type",
            "enum": [
              "None",
              "WEP",
              "WPA_PSK",
              "WPA2_PSK"
            ]
          },
          "wet": {
            "description": "Indicates Wi-Fi Encryption Type",
            "enum": [
              "None",
              "WEP_64",
              "WEP_128",
              "TKIP",
              "AES",
              "TKIP_AES"
            ]
          }
        },
        "required": [
          "tnn",
          "wat",
          "wet"
        ],
        "type": "object"
      }

  }
}
```

2193 ### B.3.5 Property Definition

Property name	Value type	Mandatory	Access mode	Description
swet	array: see schema	yes	Read Only	Indicates supported Wi-Fi Encryption types. It can be multiple
wet	string	yes		Indicates Wi-Fi Encryption Type
swmt	array: see schema	yes	Read Only	Indicates supported Wi-Fi mode types. It can be multiple
swat	array: see schema	yes	Read Only	Indicates supported Wi-Fi Auth types. It can be multiple
cd	string			Indicates credential information of Wi-Fi AP
tnn	string	yes		Indicates Target Network Name (SSID of Wi-Fi AP)
wat	string	yes		Indicates Wi-Fi Auth Type
swf	array: see schema	yes	Read Only	Indicates Supported Wi-Fi frequencies by the Enrollee. Can be multiple. Valid values are ('2.4G', '5G')
tnn	string	yes		Indicates Target Network Name (SSID of Wi-Fi AP)
wet	multiple types: see schema	yes		Indicates Wi-Fi Encryption Type
wat	multiple types: see schema	yes		Indicates Wi-Fi Auth Type
cd	string			Indicates credential information of Wi-Fi AP

2194 ### B.3.6 CRUDN behaviour

Resource	Create	Read	Update	Delete	Notify
/example/WiFiConfBaselineInterfaceResURI		get	post		

2195