

OCF Wi-Fi Easy Setup Specification

VERSION 2.0.1 | February 11, 2019



OPEN CONNECTIVITY
FOUNDATION®

CONTACT admin@openconnectivity.org
Copyright OCF © 2019. All Rights Reserved.

Legal Disclaimer

NOTHING CONTAINED IN THIS DOCUMENT SHALL BE DEEMED AS GRANTING YOU ANY KIND OF LICENSE IN ITS CONTENT, EITHER EXPRESSLY OR IMPLIEDLY, OR TO ANY INTELLECTUAL PROPERTY OWNED OR CONTROLLED BY ANY OF THE AUTHORS OR DEVELOPERS OF THIS DOCUMENT. THE INFORMATION CONTAINED HEREIN IS PROVIDED ON AN "AS IS" BASIS, AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, THE AUTHORS AND DEVELOPERS OF THIS SPECIFICATION HEREBY DISCLAIM ALL OTHER WARRANTIES AND CONDITIONS, EITHER EXPRESS OR IMPLIED, STATUTORY OR AT COMMON LAW, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. OPEN CONNECTIVITY FOUNDATION, INC. FURTHER DISCLAIMS ANY AND ALL WARRANTIES OF NON-INFRINGEMENT, ACCURACY OR LACK OF VIRUSES.

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

Copyright © 2017-2019 Open Connectivity Foundation, Inc. All rights reserved.

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

CONTENTS

1	Scope	1
2	Normative references	1
3	Terms, definitions, and abbreviated terms	2
3.1	Terms and definitions	2
3.2	Abbreviated terms	2
4	Document conventions and organization	3
4.1	Conventions	3
4.2	Notation	3
5	Overview	4
5.1	Introduction	4
5.2	Architecture	4
5.3	Example Scenario	4
6	Resource model	5
6.1	Introduction	5
6.2	EasySetup Resource	5
6.2.1	Overview	5
6.2.2	Resource	5
6.3	WiFiConf Resource Type	7
6.3.1	Introduction	7
6.3.2	Resource Type	7
6.4	DevConf Resource Type	8
6.4.1	Introduction	8
6.4.2	Resource Type	8
7	Network and connectivity	9
8	Functional interactions	9
8.1	Onboarding, Provisioning and Configuration	9
8.2	Resource discovery	9
8.3	Retrieving and Updating Easy Setup Resources	9
8.4	Error Handling	9
8.5	Example Easy Setup Flow	10
8.6	Easy Setup SSID Tags	12
8.7	Easy Setup Information Element	12
8.7.1	Overview	12
8.7.2	OCF Device Information Element (IE)	12
9	Security	15
Annex A (normative)	OpenAPI 2.0 specification definitions	16
A.1	List of Resource Type definitions	16
A.2	Device Configuration	16
A.2.1	Introduction	16

A.2.2	Example URI	16
A.2.3	Resource type	16
A.2.4	OpenAPI 2.0 definition.....	16
A.2.5	Property definition	18
A.2.6	CRUDN behaviour	19
A.3	Easy Setup Collection Link List Interface	19
A.3.1	Introduction	19
A.3.2	Example URI	19
A.3.3	Resource type	19
A.3.4	OpenAPI 2.0 definition.....	19
A.3.5	Property definition	34
A.3.6	CRUDN behaviour	38
A.4	Wi-Fi Configuration Resource Baseline Interface	38
A.4.1	Introduction	38
A.4.2	Example URI	38
A.4.3	Resource type	38
A.4.4	OpenAPI 2.0 definition.....	38
A.4.5	Property definition	43
A.4.6	CRUDN behaviour	44

Figures

Figure 1 – Easy Setup deployment architecture	4
Figure 2 – Easy Setup Resource Types	5
Figure 3 – Easy Setup Flow (Informative)	11
Figure 4 – Easy Setup Information Element Definition.....	12
Figure 5 – Type-Length-Value Structure	13

Tables

Table 1 – EasySetup Resource Type	5
Table 2 – "oic.r.easyssetup" Resource Type definition.....	6
Table 3 – WiFiConf Resource Type.....	7
Table 4 – "oic.r.wificonf" Resource Type definition.....	7
Table 5 – DevConf Resource Type	8
Table 6 – "oic.r.devconf" Resource Type definition	8
Table 7 – Easy Setup Information Element TLVs	13
Table A.1 – Alphabetized list of resources	16
Table A.2 – The Property definitions of the Resource with type 'rt' = ['oic.r.devconf'].....	18
Table A.3 – The CRUDN operations of the Resource with type 'rt' = ['oic.r.devconf']	19
Table A.4 – The Property definitions of the Resource with type 'rt' = ['oic.r.easyssetup', 'oic.wk.col']	34
Table A.5 – The CRUDN operations of the Resource with type 'rt' = ['oic.r.easyssetup', 'oic.wk.col']	38
Table A.6 – The Property definitions of the Resource with type 'rt' = ['oic.r.wificonf']	43
Table A.7 – The CRUDN operations of the Resource with type 'rt' = ['oic.r.wificonf']	44

1 Scope

This document defines functional extensions to the capabilities defined in ISO/IEC 30118-1:2018 to meet the requirements of Wi-Fi Easy Setup. It specifies new Resource Types to enable the functionality and any extensions to the existing capabilities defined in ISO/IEC 30118-1:2018.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 30118-1:2018 Information technology -- Open Connectivity Foundation (OCF) Specification -- Part 1: Core specification
<https://www.iso.org/standard/53238.html>
Latest version available at: https://openconnectivity.org/specs/OCF_Core_Specification.pdf

ISO/IEC 30118-2:2018 Information technology -- Open Connectivity Foundation (OCF) Specification -- Part 2: Security specification
<https://www.iso.org/standard/74239.html>
Latest version available at: https://openconnectivity.org/specs/OCF_Security_Specification.pdf

ISO/IEC 30118-5:2018 Information technology -- Open Connectivity Foundation (OCF) Specification -- Part 5: Smart home device specification
<https://www.iso.org/standard/74242.html>
Latest version available at: https://openconnectivity.org/specs/OCF_Device_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 5646, *Tags for Identifying Languages*, September 2009
<https://www.rfc-editor.org/info/rfc5646>

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

31

32 **3 Terms, definitions, and abbreviated terms**

33 **3.1 Terms and definitions**

34 For the purposes of this document, the terms and definitions given in ISO/IEC 30118-1:2018 and
35 the following apply.

36 ISO and IEC maintain terminological databases for use in standardization at the following
37 addresses:

38 – ISO Online browsing platform: available at <https://www.iso.org/obp>

39 – IEC Electropedia: available at <http://www.electropedia.org/>

40 **3.1.1**

41 **Easy Setup**

42 process of configuring an Enrollee (3.1.3) using a Mediator (3.1.5) by transferring of essential
43 information to the Enrollee (3.1.3)

44 **3.1.2**

45 **Easy Setup Enrollment**

46 step during Easy Setup in which the Enrollee (3.1.3) is contacted by the Mediator (3.1.5) to
47 configure the Enroller's (3.1.4) information by means of accessing Easy Setup (3.1.1) Resources

48 **3.1.3**

49 **Enrollee**

50 device that needs to be configured and connected. E.g. Air-conditioner, Printer

51 **3.1.4**

52 **Enroller**

53 target network entity to which the Enrollee (3.1.3) connects. E.g. Wi-Fi AP

54 **3.1.5**

55 **Mediator**

56 logical function that enables the Enrollee (3.1.3) to connect to the target network (i.e. Enroller
57 (3.1.4))

58 Note 1 to Entry: The Mediator transfers configuration information to the Enrollee. E.g. Mobile Phone

59 **3.2 Abbreviated terms**

60 **3.2.1**

61 **CID**

62 Company Identifier (ID)

63 **3.2.2**

64 **IE**

65 Information Element

66 **3.2.3**

67 **Soft AP**

68 Software Enabled Access Point

69 **3.2.4**

70 **TLV**

71 type-length-value

72 **4 Document conventions and organization**

73 **4.1 Conventions**

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

78 **4.2 Notation**

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

81 Required (or shall or mandatory)(M).

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

85 Recommended (or should)(S).

- 86 – These features add functionality supported by Core Architecture and should be implemented.
87 Recommended features take advantage of the capabilities Core Architecture, usually without
88 imposing major increase of complexity. Notice that for compliance testing, if a recommended
89 feature is implemented, it shall meet the specified requirements to be in compliance with these
90 guidelines. Some recommended features could become requirements in the future. The phrase
91 "should not" indicates behaviour that is permitted but not recommended.

92 Allowed (may or allowed)(O).

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

96 DEPRECATED.

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

103 Conditionally allowed (CA)

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

106 Conditionally required (CR)

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

110

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

112 Words that are emphasized are printed in italic.

113 **5 Overview**

114 **5.1 Introduction**

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

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

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

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

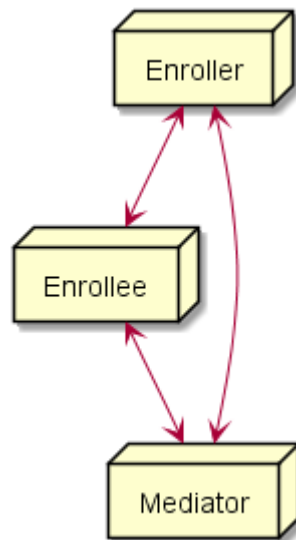
124 Easy Setup can be enhanced in future by incorporating other suitable technologies.

125 Annex A specifies the Resource Type definitions using the schema defined in the OpenAPI
126 specification as the API definition language that shall be followed by an OCF Device realizing the
127 Resources specified in this document.

128 **5.2 Architecture**

129 Figure 1 shows the deployment architectural approach.

130



131

132

Figure 1 – Easy Setup deployment architecture

133 Easy Setup defines the following roles: Enrollee, Enroller, and Mediator. Please refer to clause 3
134 for the definitions thereof.

135 **5.3 Example Scenario**

136 The following scenario presents a typical setup case.

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

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

145 **6 Resource model**

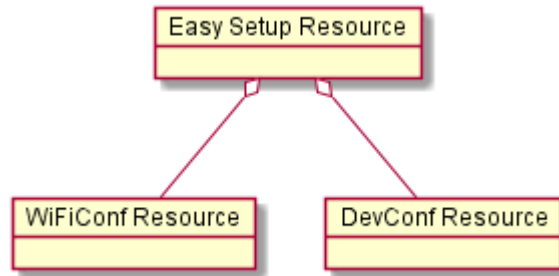
146 **6.1 Introduction**

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

- 148 1) EasySetup Resource Type
- 149 2) WiFiConf Resource Type
- 150 3) DevConf Resource Type

151 The EasySetup Resource Type is a Collection Resource and shall contain Links to instances of at
 152 least WiFiConf and DevConf. A vendor may add links to other Resource Types. The relationship
 153 between the EasySetup Resource Type and linked Resources is shown in Figure 2.

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



156
 157

Figure 2 – Easy Setup Resource Types

158 **6.2 EasySetup Resource**

159 **6.2.1 Overview**

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

162 **6.2.2 Resource**

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

164

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.easyssetup, oic.wk.col	oic.if.baseline, oic.if.ll, oic.if.b	Top level Resource for Easy Setup. Indicates easy setup status.	N/A

				The Resource properties exposed are listed in Table 2.	
--	--	--	--	--	--

165

166 Table 2 defines the details for the "oic.r.easyssetup" Resource Type.

167

Table 2 – "oic.r.easyssetup" Resource Type definition

Property title	Property name	Value type	Value rule	Unit	Access mode	Mandatory	Description
Easy Setup Provisioning Status	ps	integer	enum	N/A	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	N/A	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	N/A	N/A	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	N/A	N/A	R	Yes	Array of links that are WiFiConf and DevConf Resource.

168

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

170 – "ps" equal to 0.

171 – "lec" equal to 0.

172 – "cn" equal to an empty array.

173 6.3 WiFiConf Resource Type

174 6.3.1 Introduction

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

177 6.3.2 Resource Type

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

179 **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.	N/A

180

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

182 **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	N/A	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.	N/A	R	Yes	Supported Wi-Fi frequencies by Enrollee. Can be multiple. ("2.4G", "5G")
Target Network Name	tnn	string	N/A	N/A	RW	Yes	Target network name (SSID of Wi-Fi AP i.e. enroller)
Credential	cd	string	N/A	N/A	RW	No	Credential information of Wi-Fi AP (Password used to connect to enroller).
Wi-Fi Auth Type	wat	string	enum	N/A	RW	Yes	Wi-Fi auth type ("None", "WEP", "WPA_PSK", "WPA2_PSK")
Wi-Fi Encryption Type	wet	string	enum	N/A	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	N/A	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	N/A	R	Yes	Supported Wi-Fi Encryption types. Can be multiple. ("None", "WEP-64", "WEP_128", "TKIP", "AES", "TKIP_AES")

183

184 6.4 DevConf Resource Type

185 6.4.1 Introduction

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

187 6.4.2 Resource Type

188 The DevConf Resource Type is as defined in Table 5

189

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.	N/A

190

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

192

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	N/A	N/A	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.

193

194 **7 Network and connectivity**

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

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

198 **8 Functional interactions**

199 **8.1 Onboarding, Provisioning and Configuration**

200 The Mediator may be present as a standalone function or in conjunction with other functions or
201 services such as AMS as part of an OBT (Onboarding Tool); please refer to the ISO/IEC 30118-
202 2:2018.

203 **8.2 Resource discovery**

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

205 When in Easy Setup phase, if using Wi-Fi as the connectivity between the Enrollee and the Mediator
206 then the Enrollee shall make itself discoverable as a Soft AP. The Soft AP has additional availability
207 constraints which are documented in ISO/IEC 30118-2:2018.

208 **8.3 Retrieving and Updating Easy Setup Resources**

209 The Enrollee shall expose Easy Setup Resources such that a Mediator is able to discover them
210 using standard OCF Resource discovery methods (i.e. via a RETRIEVE on /oic/res); see the
211 ISO/IEC 30118-1:2018, clause 11.3.

212 Easy Setup Resources shall expose only secure Endpoints (e.g. CoAPS); see the ISO/IEC 30118-
213 1:2018, clause 10.

214 The Mediator may RETRIEVE a Resource within the Easy Setup Collection or the Collection itself
215 to check the Enrollee's status at any stage of Easy Setup. This applies only when the Enrollee &
216 the Mediator are on a common network.

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

221 For details of Easy Setup Resources refer to clause 6.

222 **8.4 Error Handling**

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

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

234 – The Enrollee should set "lec" Property to 9, if it fails to connect due to wrong Encryption type
235 information (even though it's supported by the Enrollee).

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

239 **8.5 Example Easy Setup Flow**

240 Figure 3 shows an example Easy Setup flow for informative purposes:

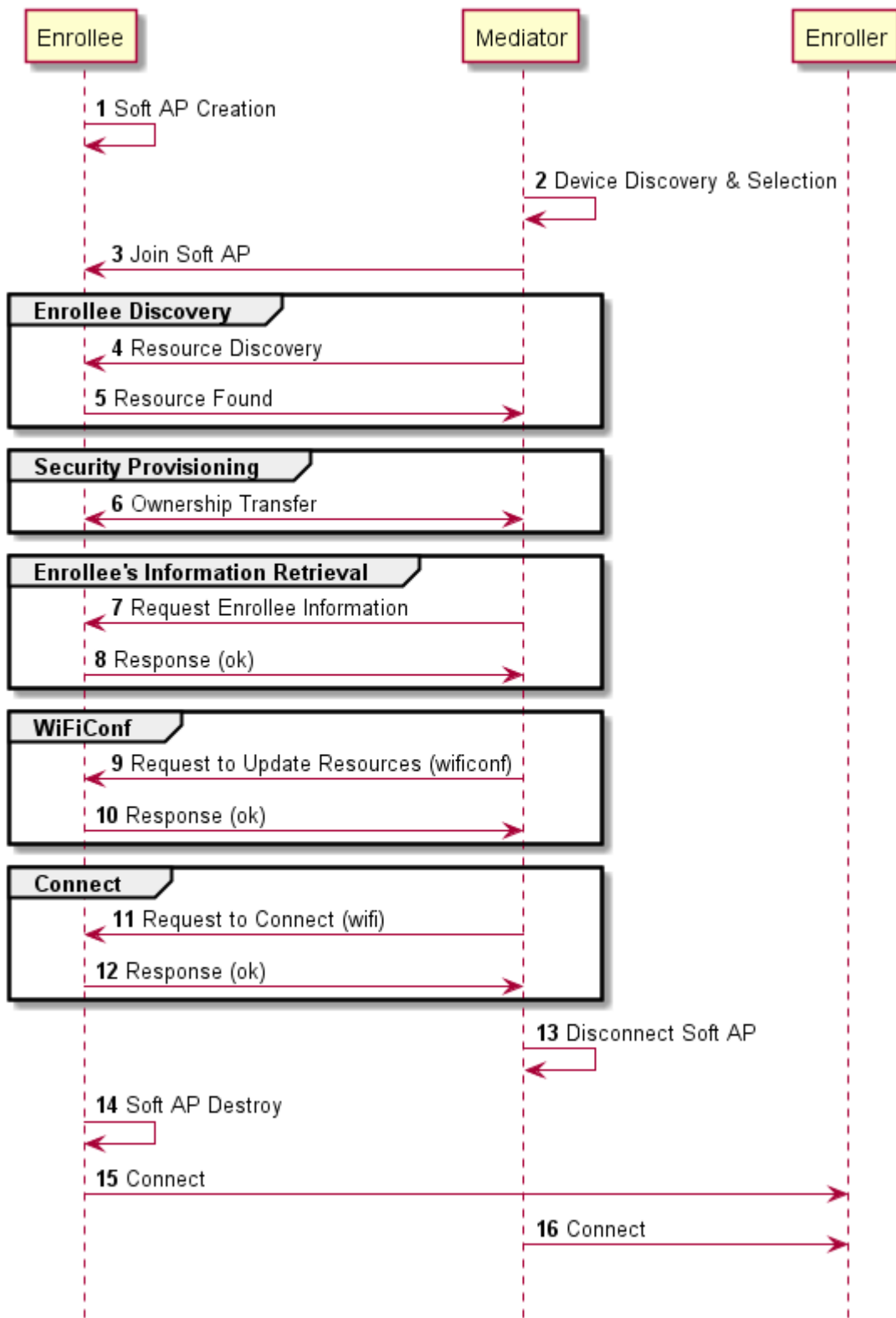


Figure 3 – Easy Setup Flow (Informative)

241

242

243

244 The example flow in Figure 1Figure 3 undergoes security provisioning (step 6) during Easy Setup.
 245 Alternatively, security provisioning can be done before Enrollee Discovery (steps 4 and 5) if
 246 preferred. Please refer to the ISO/IEC 30118-2:2018 for more information on the different scenarios.

247 **8.6 Easy Setup SSID Tags**

248 If using Wi-Fi as the connectivity between the Enrollee and the Mediator then the Enrollee's Soft
 249 AP SSID should contain exactly one of the following Easy Setup SSID tags:

- 250 – "OCF_"
 - 251 – Prefix tag that has to be at the beginning of the SSID.
 - 252 – Example: OCF_MySSID
- 253 – "_OCF"
 - 254 – Suffix tag that has to be at the end of the SSID.
 - 255 – Example: MySSID_OCF

256 These tags are case sensitive.

257 **8.7 Easy Setup Information Element**

258 **8.7.1 Overview**

259 If using Wi-Fi as the connectivity between the Enrollee and the Mediator then the Enrollee's Soft
 260 AP beacon should contain the Easy Setup Information Element. The information element provides
 261 additional information about the device such as a friendly name or device manufacturer for the
 262 mediator application. The mediator application can then use this information to provide a better
 263 user experience.

264 **8.7.2 OCF Device Information Element (IE)**

265 The Easy Setup Information Element has the structure shown in Figure 4

1 byte	1 byte	3 bytes	1 byte	<252 bytes
Type = 221	Length	CID = 6A 40 65	OCF IE Type = 0	Data

266 **Figure 4 – Easy Setup Information Element Definition**

- 267 – Type is a unique id allocated by the IEEE registrar to identify different information elements
 268 from each other. The Easy Setup Information Element shall have a Type value of 221 which is
 269 standard vendor specific information element.
- 270 – Length shall indicate the total size of CID, OCF IE Type, and Data in bytes.
- 271 – Company ID (CID) is a unique 24-bit identifier for a specific company or organization. The Easy
 272 Setup Information Element shall have a CID value of 6A 40 65.
- 273 – OCF IE Type is the identifier of the specific IE within OCF. The OCF IE Type shall be set to 0
 274 for Easy Setup.
- 275 – Data is a set of type-length-value (TLV) structures that represent the device information in Table
 276 1. The length of this field shall be less than 252 bytes.

277

278 Each TLV has the structure shown in Figure 5.

1 byte	1 byte	<250 bytes
Type	Length	Value

Figure 5 – Type-Length-Value Structure

279

- 280 – Type shall indicate the type of the field from .
- 281 – Table 7.
- 282 – Length shall indicate the length of the Value in bytes.
- 283 – Value shall represent the corresponding information for specific TLV type from .
- 284 – Table 7.

285 Data is a set of TLVs as defined in Table 7.

286

Table 7 – Easy Setup Information Element TLVs

Type	Length (bytes)	Value	Description of TLV	# of Occurrences in IE or IEC	Required
1	<65	Friendly name of the device	Device Friendly Name	1	Y
2	<27	Device Type	Device type/Class	>=1	Y
3	<65	Name of Device Manufacturer	Manufacturer Name	1	Y
4	<43	Language tag for strings	See IETF RFC 5646	1	Y
5	16	Protocol Independent ID in network byte order	See ISO/IEC 30118-1:2018	1	Y
101	<65	Device Type/Class	Device Type as string	>=0	N

287

288 The TLVs may be set in any order inside an IE or IEC. All strings shall be UTF-8 encoded and shall
289 not include a null terminator. All TLVs in .

290 Table 7 with a 'Required' value of 'Y' shall be included in the IE or IEC (if multiple IEs are required).
291 The value of each TLV shall meet the length requirements specified in Table 1.

292 **8.7.2.1 Device Friendly Name (Type 1)**

293 User readable string representing the friendly name of the device that is beaconing and ready to
294 undergo Easy Setup. This should match 'n' from oic.wk.d as defined in the ISO/IEC 30118-1:2018.

295 This string is in the same language specified in the type 4 TLV.

296 **8.7.2.2 Device Type (Type 2)**

297 Device type shall be the shortened form of 'Device Type' as specified in the ISO/IEC 30118-5:2018. For
298 example:

- 299 – 'Device Type' as specified in the ISO/IEC 30118-5:2018: "oic.d.airconditioner"
- 300 – 'Device Type' as specified in a type 2 TLV: "airconditioner"
- 301

302 In cases where the device supports multiple functions, several type 2 TLVs may be included to
303 represent each function of the device.

304 If the device does not support any of the functions as specified in the ISO/IEC 30118-5:2018, at
305 least one type 101 TLV shall be included. Type 101 TLV contains a user readable string in the
306 same language specified in the type 4 TLV. (Ex: "Lock").

307 If the device supports more than one function, a mix of type 2 and type 101 TLVs may be used
308 depending on which functions are defined in the ISO/IEC 30118-5:2018.

309 **8.7.2.3 Device Manufacturer Name (Type 3)**

310 User readable string representing the manufacturer name of the device that is beaconing and ready
311 to undergo Easy Setup. This should match 'mnmn' from oic.wk.p as defined in the ISO/IEC 30118-
312 1:2018.

313 This string is in the same language specified in the type 4 TLV.

314 **8.7.2.4 Language Tag (Type 4)**

315 The language of all strings shall be specified in a type 4 TLV. The value of the type 4 TLV shall
316 contain a language tag as described in IETF RFC 5646 (Ex: "en-us"). If the actual length of the
317 language tag exceeds 42 bytes the manufacturer shall exclude subtags on the language tag until
318 it is less than 43 bytes.

319 Please see 8.7.2.8 for information on supporting multiple languages.

320 If an IE contains a TLV that is a string (i.e. type 1, type 3 or type 101), then a type 4 TLV
321 corresponding to the language of the string(s) shall also be present in the IE.

322 **8.7.2.5 Protocol Independent ID (Type 5)**

323 This shall match 'piid' from oic.wk.d as defined in the ISO/IEC 30118-1:2018.

324 The piid in the TLV shall be in network byte order.

325 **8.7.2.6 Multiple Information Elements**

326 Additional Easy Setup IEs may be present in the Soft AP beacon in the following situations:

- 327 – The total size of the TLVs is larger than the size of 'Data' as defined in an Easy Setup
328 Information Element.
- 329 – Support for multiple languages is necessary.

330 Two or more Easy Setup Information Elements are referred to as an Information Element Collection
331 (IEC).

332 **8.7.2.7 IEC for Large TLV Size Support**

333 If a TLV or set of TLVs will not fit into the current IE, a manufacturer may add additional Easy Setup
334 IEs to contain the TLV/s thereby creating or extending an IEC. The additional IE shall contain the
335 following fields as described in 8.7.2:

- 336 – Type
- 337 – Length
- 338 – CID
- 339 – OCF IE Type

340 If an IE contains a TLV that is a string (i.e. type 1, type 3 or type 101), then a type 4 TLV
341 corresponding to the language of the string(s) shall also be present in the IE.

342 **8.7.2.8 IEC for Multiple Language Support**

343 A manufacturer may include additional Easy Setup IEs to support multiple languages in the Soft
344 AP beacon. In the case that a manufacturer needs to provide device information in more than one
345 language, they shall include an additional copy of the IE/IEC for each additional language. Each
346 additional IE/IEC shall include all of the mandatory TLVs defined in 8.7.2.

347 **9 Security**

348 A Device shall meet the Wi-Fi Easy Setup security requirements specified in ISO/IEC 30118-2:2018.

Annex A(normative)

OpenAPI 2.0 specification definitions

A.1 List of Resource Type definitions

Table A.1 contains the list of defined resources in this document.

Table A.1 – Alphabetized list of resources

Friendly Name (informative)	Resource Type (rt)	Clause
Device Configuration	"oic.r.devconf"	A.1
Easy Setup	"oic.r.easyssetup"	A.2
Wi-Fi Configuration	"oic.r.wificonf"	A.3

A.2 Device Configuration

A.2.1 Introduction

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

A.2.2 Example URI

/example/DevConfResURI

A.2.3 Resource type

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

A.2.4 OpenAPI 2.0 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\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\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" : {
```

```

395     "get": {
396         "description": "Device configuration resource stores a preference of device settings
397 like\ndevice name. Vender-specific information can be added to the resource.\nRetrieve various
398 settings regarding to device-specific settings\n1. Device name (human-friendly name to be detected
399 by mediator during\n  easy setup)\n",
400         "parameters": [
401             {"$ref": "#/parameters/interface"}
402         ],
403         "responses": {
404             "200": {
405                 "description": "",
406                 "x-example":
407                 {
408                     "rt": ["oic.r.devconf"],
409                     "dn": "My Refrigerator"
410                 }
411             },
412             "schema": { "$ref": "#/definitions/DevConf" }
413         }
414     }
415 },
416 },
417 },
418 "parameters": {
419     "interface" : {
420         "in" : "query",
421         "name" : "if",
422         "type" : "string",
423         "enum" : ["oic.if.baseline", "oic.if.r"]
424     }
425 },
426 "definitions": {
427     "DevConf" : {
428         "properties": {
429             "rt" : {
430                 "description": "Resource Type of the Resource",
431                 "items": {
432                     "maxLength": 64,
433                     "type": "string"
434                 },
435                 "minItems": 1,
436                 "readOnly": true,
437                 "type": "array"
438             },
439             "n" : {
440                 "description": "Friendly name of the resource",
441                 "maxLength": 64,
442                 "readOnly": true,
443                 "type": "string"
444             },
445             "id" : {
446                 "description": "Instance ID of this specific resource",
447                 "maxLength": 64,
448                 "readOnly": true,
449                 "type": "string"
450             },
451             "if" : {
452                 "description": "The interface set supported by this resource",
453                 "items": {
454                     "enum": [
455                         "oic.if.baseline",
456                         "oic.if.ll",
457                         "oic.if.b",
458                         "oic.if.lb",
459                         "oic.if.rw",
460                         "oic.if.r",
461                         "oic.if.a",
462                         "oic.if.s"
463                     ],
464                     "type": "string"

```

```

465     },
466     "minItems": 1,
467     "readOnly": true,
468     "type": "array"
469   },
470   "dn": {
471     "oneOf": [
472       {
473         "type": "string",
474         "description": "Indicates a pre-configured device name in language indicated by 'dl'
475 in /oic/con; presented by enrollee device to mediator device during easy-setup process",
476         "pattern": "^.*$",
477         "readOnly": true
478       },
479       {
480         "type": "array",
481         "items": {
482           "type": "object",
483           "properties": {
484             "language": {
485               "$ref": "http://openconnectivityfoundation.github.io/core/schemas/oic.types-
486 schema.json#/definitions/language-tag",
487               "readOnly": true,
488               "description": "An RFC 5646 language tag."
489             },
490             "value": {
491               "type": "string",
492               "description": "Pre-configured device name in the indicated language.",
493               "pattern": "^.*$",
494               "readOnly": true
495             }
496           }
497         },
498         "minItems": 1,
499         "readOnly": true,
500         "description": "Localized device name."
501       }
502     ]
503   },
504 },
505 "type": "object",
506 "required": ["dn"]
507 }
508 }
509 }
510

```

511 A.2.5 Property definition

512 Table A.2 defines the Properties that are part of the ['oic.r.devconf'] Resource Type

513 **Table A.2 – The Property definitions of the Resource with type 'rt' = ['oic.r.devconf']**

Property name	Value type	Mandatory	Access mode	Description
rt	array: see schema	No	Read Only	Resource Type of the Resource
dn	multiple types: see schema	Yes	Read Write	
n	string	No	Read Only	Friendly name of the resource
if	array: see schema	No	Read Only	The interface set supported by this resource
id	string	No	Read Only	Instance ID of this specific resource

514 **A.2.6 CRUDN behaviour**

515 Table A.3 defines the CRUDN operations that are supported on the ['oic.r.devconf'] Resource Type

516 **Table A.3 – The CRUDN operations of the Resource with type 'rt' = ['oic.r.devconf']**

Create	Read	Update	Delete	Notify
	get			observe

517 **A.3 Easy Setup Collection Link List Interface**

518 **A.3.1 Introduction**

519 Easy Setup resource stores useful information including current status of
520 unboxing device and last error code which are produced in a process of
521 easy setup.

522 Note that, Easy Setup resource is a type of collection resource, which
523 contains links to WiFiConf, DevConf resources and may additionally contain
524 links to other resources.

526 **A.3.2 Example URI**

527 /EasySetupResURI

528 **A.3.3 Resource type**

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

530 **A.3.4 OpenAPI 2.0 definition**

```
531 {  
532   "swagger": "2.0",  
533   "info": {  
534     "title": "Easy Setup Collection Link List Interface",  
535     "version": "v0.0.3-20170611",  
536     "license": {  
537       "name": "copyright 2016-2017 Open Connectivity Foundation, Inc. All rights reserved.",  
538       "x-description": "Redistribution and use in source and binary forms, with or without  
539 modification, are permitted provided that the following conditions are met:\n      1.  
540 Redistributions of source code must retain the above copyright notice, this list of conditions and  
541 the following disclaimer.\n      2. Redistributions in binary form must reproduce the above  
542 copyright notice, this list of conditions and the following disclaimer in the documentation and/or  
543 other materials provided with the distribution.\n\n      THIS SOFTWARE IS PROVIDED BY THE Open  
544 Connectivity Foundation, INC. \n      \"AS IS\" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT  
545 LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR  
546 WARRANTIES OF NON-INFRINGEMENT, ARE DISCLAIMED.\n\n      IN NO EVENT SHALL THE Open Connectivity  
547 Foundation, INC. OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY,  
548 OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR  
549 SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)\n\n      HOWEVER CAUSED AND ON  
550 ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR  
551 OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF  
552 SUCH DAMAGE.\n\n    }  
553   },  
554   "schemes": ["http"],  
555   "consumes": ["application/json"],  
556   "produces": ["application/json"],  
557   "paths": {  
558     "/EasySetupResURI?if=oic.if.ll" : {  
559       "get": {  
560         "description": "Easy Setup resource stores useful information including current status  
561 of\n\nunboxing device and last error code which are produced in a process of\n\neasy setup.\n\nNote that,  
562 Easy Setup resource is a type of collection resource, which\n\ncontains links to WiFiConf, DevConf  
563 resources and may additionally contain\n\nlinks to other resources.\n\nRetrieve useful information  
564 during easy setup process :\n\n1. A current status in easy setup process.\n\n2. A last error code  
565 describing reason for failure occurred at the last\n\n time.\n\n",  
566       "parameters": [  
567
```

```

568     {"$ref": "#/parameters/interface-11"}
569   ],
570   "responses": {
571     "200": {
572       "description": "",
573       "x-example":
574         [
575           {
576             "href": "/EasySetupResURI",
577             "rt": ["oic.r.easyssetup", "oic.wk.col"],
578             "if": ["oic.if.b"],
579             "p": {"bm": 3},
580             "eps": [
581               {"ep": "coaps://[fe80::bld6]:1111", "pri": 2}
582             ],
583             "rel": ["self", "item"]
584           },
585           {
586             "href": "/WiFiConfResURI",
587             "rt": ["oic.r.wificonf"],
588             "if": ["oic.if.baseline"],
589             "p": {"bm": 3},
590             "eps": [
591               {"ep": "coaps://[fe80::bld6]:1111", "pri": 2}
592             ]
593           },
594           {
595             "href": "/DevConfResURI",
596             "rt": ["oic.r.devconf"],
597             "if": ["oic.if.baseline"],
598             "p": {"bm": 3},
599             "eps": [
600               {"ep": "coaps://[fe80::bld6]:1111", "pri": 2}
601             ]
602           }
603         ]
604       ,
605       "schema": { "$ref": "#/definitions/slinks" }
606     }
607   }
608 },
609 "/EasySetupResURI?if=oic.if.b" : {
610   "get": {
611     "description": "Easy Setup resource stores useful information including current status
612 of\n\nboxing device and last error code which are produced in a process of\n\neasy setup.\n\nNote that,
613 Easy Setup resource is a type of collection resource, which\n\ncontains links to WiFiConf, DevConf
614 resources and may additionally contain\n\nlinks to other resources.\n\nRetrieve useful information
615 during easy setup process :\n\n1. A current status in easy setup process.\n\n2. A last error code
616 describing reason for failure occurred at the last\n\n time.\n\n",
617     "parameters": [
618       {"$ref": "#/parameters/interface-batch"}
619     ],
620     "responses": {
621       "200": {
622         "description": "",
623         "x-example":
624           [
625             {
626               "href": "/EasySetupResURI",
627               "rep": {
628                 "ps": 0,
629                 "lec": 0,
630                 "cn": [1]
631               }
632             },
633             {
634               "href": "/WiFiConfResURI",
635               "rep": {
636                 "swmt": ["A", "B", "G"],

```

```

638         "swf": ["2.4G", "5G"],
639         "tnn": "Home_AP_SSID",
640         "cd": "Home_AP_PWD",
641         "wat": "WPA2_PSK",
642         "wet": "AES"
643     }
644 },
645 {
646     "href": "/DevConfResURI",
647     "rep": {
648         "dn": "My Refrigerator"
649     }
650 }
651 ],
652 ,
653 "schema": { "$ref": "#/definitions/sbatch" }
654 }
655 },
656 },
657 "post": {
658     "description": "Able to deliver Wi-Fi, Device configuration and other
659 configuration\ninformation in a batch by utilizing 'batch' interface.\nIf you want to deliver Wi-Fi
660 and Device configuration information in a batch,\nyou can write all properties you want to send with
661 a 'batch' interface.\nThe below example is the case to send Easy Setup and Wi-Fi
662 configuration\n(i.e. connection type, target network, auth type information) in a batch.\n",
663     "parameters": [
664         { "$ref": "#/parameters/interface-batch" },
665         {
666             "name": "body",
667             "in": "body",
668             "required": true,
669             "schema": { "$ref": "#/definitions/sbatch-update" },
670             "x-example":
671             [
672                 {
673                     "href": "/EasySetupResURI",
674                     "rep": {
675                         "cn": [1]
676                     }
677                 },
678                 {
679                     "href": "/WiFiConfResURI",
680                     "rep": {
681                         "tnn": "Home_AP_SSID",
682                         "cd": "Home_AP_PWD",
683                         "wat": "WPA2_PSK",
684                         "wet": "AES"
685                     }
686                 }
687             ]
688         }
689     ],
690     "responses": {
691         "200": {
692             "description": "",
693             "x-example":
694             [
695                 {
696                     "href": "/EasySetupResURI",
697                     "rep": {
698                         "ps": 0,
699                         "lec": 0,
700                         "cn": [1]
701                     }
702                 },
703                 {
704                     "href": "/WiFiConfResURI",
705                     "rep": {
706                         "swmt": ["A", "B", "G"],
707                         "swf": ["2.4G", "5G"],

```

```

708         "tnn": "Home_AP_SSID",
709         "cd": "Home_AP_PWD",
710         "wat": "WPA2_PSK",
711         "wet": "AES"
712     }
713 },
714 {
715     "href": "/DevConfResURI",
716     "rep": {
717         "dn": "My Refrigerator"
718     }
719 }
720 ]
721 ,
722 "schema": { "$ref": "#/definitions/sbatch" }
723 }
724 }
725 },
726 },
727 "/EasySetupResURI?if=oic.if.baseline" : {
728     "get": {
729         "description": "Easy Setup resource stores useful information including current status
730 of\nnunboxing device and last error code which are produced in a process of\neasy setup.\nNote that,
731 Easy Setup resource is a type of collection resource, which\ncontains links to WiFiConf, DevConf
732 resources and may additionally contain\nlinks to other resources.\nRetrieve useful information
733 during easy setup process :\n 1. A current status in easy setup process.\n 2. A last error code
734 describing reason for failure occurred at the last\n      time.\n",
735         "parameters": [
736             { "$ref": "#/parameters/interface-baseline" }
737         ],
738         "responses": {
739             "200": {
740                 "description": "",
741                 "x-example":
742                 {
743                     "rt": ["oic.r.easyssetup", "oic.wk.col"],
744                     "if": ["oic.if.ll", "oic.if.baseline", "oic.if.b"],
745                     "ps": 0,
746                     "lec": 0,
747                     "cn": [1],
748                     "links": [
749                         {
750                             "href": "/EasySetupResURI",
751                             "rt": ["oic.r.easyssetup", "oic.wk.col"],
752                             "if": ["oic.if.b"],
753                             "p": {"bm": 3},
754                             "eps": [
755                                 {"ep": "coaps://[fe80::bld6]:1111", "pri": 2}
756                             ],
757                             "rel": ["self", "item"]
758                         },
759                         {
760                             "href": "/WiFiConfResURI",
761                             "rt": ["oic.r.wificonf"],
762                             "if": ["oic.if.baseline"],
763                             "p": {"bm": 3},
764                             "eps": [
765                                 {"ep": "coaps://[fe80::bld6]:1111", "pri": 2}
766                             ]
767                         },
768                         {
769                             "href": "/DevConfResURI",
770                             "rt": ["oic.r.devconf"],
771                             "if": ["oic.if.baseline"],
772                             "p": {"bm": 3},
773                             "eps": [
774                                 {"ep": "coaps://[fe80::bld6]:1111", "pri": 2}
775                             ]
776                         }
777                     ]
778                 }
779             }
780         }
781     }
782 }

```

```

778         }
779         ,
780         "schema": { "$ref": "#/definitions/EasySetup" }
781     }
782 }
783 }
784 }
785 },
786 "parameters": {
787     "interface-ll" : {
788         "in" : "query",
789         "name" : "if",
790         "type" : "string",
791         "enum" : ["oic.if.ll"]
792     },
793     "interface-baseline" : {
794         "in" : "query",
795         "name" : "if",
796         "type" : "string",
797         "enum" : ["oic.if.baseline"]
798     },
799     "interface-all" : {
800         "in" : "query",
801         "name" : "if",
802         "type" : "string",
803         "enum" : ["oic.if.baseline", "oic.if.ll", "oic.if.b"]
804     },
805     "interface-batch" : {
806         "in" : "query",
807         "name" : "if",
808         "type" : "string",
809         "enum" : ["oic.if.b"]
810     }
811 },
812 "definitions": {
813     "slinks" : {
814         "items" :
815         {
816             "properties": {
817                 "anchor": {
818                     "description": "This is used to override the context URI e.g. override the URI of the
819 containing collection.",
820                     "format": "uri",
821                     "maxLength": 256,
822                     "type": "string"
823                 },
824                 "di": {
825                     "description": "The Device ID formatted according to IETF RFC 4122.",
826                     "pattern": "^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-
827 9]{12}$",
828                     "type": "string"
829                 },
830                 "eps": {
831                     "description": "the Endpoint information of the target Resource",
832                     "items": {
833                         "properties": {
834                             "ep": {
835                                 "description": "Transport Protocol Suite + Endpoint Locator",
836                                 "format": "uri",
837                                 "type": "string"
838                             },
839                             "pri": {
840                                 "description": "The priority among multiple Endpoints",
841                                 "minimum": 1,
842                                 "type": "integer"
843                             }
844                         },
845                         "type": "object"
846                     },
847                     "type": "array"

```

```

848     },
849     "href": {
850       "description": "This is the target URI, it can be specified as a Relative Reference or
851 fully-qualified URI.",
852       "format": "uri",
853       "maxLength": 256,
854       "type": "string"
855     },
856     "if": {
857       "description": "The interface set supported by this resource",
858       "items": {
859         "enum": [
860           "oic.if.baseline",
861           "oic.if.ll",
862           "oic.if.b",
863           "oic.if.rw",
864           "oic.if.r",
865           "oic.if.a",
866           "oic.if.s"
867         ],
868         "type": "string"
869       },
870       "minItems": 1,
871       "type": "array"
872     },
873     "ins": {
874       "description": "The instance identifier for this web link in an array of web links -
875 used in collections",
876       "type": "integer"
877     },
878     "p": {
879       "description": "Specifies the framework policies on the Resource referenced by the
880 target URI",
881       "properties": {
882         "bm": {
883           "description": "Specifies the framework policies on the Resource referenced by the
884 target URI for e.g. observable and discoverable",
885           "type": "integer"
886         }
887       },
888       "required": [
889         "bm"
890       ],
891       "type": "object"
892     },
893     "rel": {
894       "description": "The relation of the target URI referenced by the link to the context
895 URI",
896       "oneOf": [
897         {
898           "default": [
899             "hosts"
900           ],
901           "items": {
902             "maxLength": 64,
903             "type": "string"
904           },
905           "minItems": 1,
906           "type": "array"
907         },
908         {
909           "default": "hosts",
910           "maxLength": 64,
911           "type": "string"
912         }
913       ]
914     },
915     "rt": {
916       "description": "Resource Type of the Resource",
917       "items": {

```

```

918         "maxLength": 64,
919         "type": "string"
920     },
921     "minItems": 1,
922     "type": "array"
923 },
924 "title": {
925     "description": "A title for the link relation. Can be used by the UI to provide a
926 context.",
927     "maxLength": 64,
928     "type": "string"
929 },
930 "type": {
931     "default": "application/cbor",
932     "description": "A hint at the representation of the resource referenced by the target
933 URI. This represents the media types that are used for both accepting and emitting.",
934     "items": {
935         "maxLength": 64,
936         "type": "string"
937     },
938     "minItems": 1,
939     "type": "array"
940 }
941 },
942 "required": [
943     "href",
944     "rt",
945     "if"
946 ],
947 "type": "object"
948 },
949 "type" : "array",
950 "title" : "EasySetup Object Links List Schema (auto merged)"
951 },
952 "sbatch" : {
953     "title" : "Collection Batch Retrieve Format (auto merged)",
954     "minItems" : 1,
955     "items" : {
956         "additionalProperties": true,
957         "properties": {
958             "href": {
959 anchor",
960                 "description": "URI of the target resource relative assuming the collection URI as
961                 "format": "uri",
962                 "maxLength": 256,
963                 "type": "string"
964             },
965             "rep": {
966                 "oneOf": [
967                     {
968                         "description": "The response payload from a single resource",
969                         "type": "object"
970                     },
971                     {
972                         "description": " The response payload from a collection (batch) resource",
973                         "items": {
974                             "properties": {
975                                 "anchor": {
976 of the containing collection.",
977                                     "description": "This is used to override the context URI e.g. override the URI
978                                     "format": "uri",
979                                     "maxLength": 256,
980                                     "type": "string"
981                                 },
982                                 "di": {
983                                     "allOf": [
984                                         {
985                                             "description": "Format pattern according to IETF RFC 4122.",
986                                             "pattern": "^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-
987 [a-fA-F0-9]{12}$",

```

```

988         "type": "string"
989     },
990     {
991         "description": "The device ID"
992     }
993 ]
994 },
995 "eps": {
996     "description": "the Endpoint information of the target Resource",
997     "items": {
998         "properties": {
999             "ep": {
1000                 "description": "Transport Protocol Suite + Endpoint Locator",
1001                 "format": "uri",
1002                 "type": "string"
1003             },
1004             "pri": {
1005                 "description": "The priority among multiple Endpoints",
1006                 "minimum": 1,
1007                 "type": "integer"
1008             }
1009         },
1010         "type": "object"
1011     },
1012     "type": "array"
1013 },
1014 "href": {
1015     "description": "This is the target URI, it can be specified as a Relative
1016 Reference or fully-qualified URI.",
1017     "format": "uri",
1018     "maxLength": 256,
1019     "type": "string"
1020 },
1021 "if": {
1022     "description": "The interface set supported by this resource",
1023     "items": {
1024         "enum": [
1025             "oic.if.baseline",
1026             "oic.if.ll",
1027             "oic.if.b",
1028             "oic.if.rw",
1029             "oic.if.r",
1030             "oic.if.a",
1031             "oic.if.s"
1032         ],
1033         "type": "string"
1034     },
1035     "minItems": 1,
1036     "type": "array"
1037 },
1038 "ins": {
1039     "description": "The instance identifier for this web link in an array of web
1040 links - used in collections",
1041     "type": "integer"
1042 },
1043 "p": {
1044     "description": "Specifies the framework policies on the Resource referenced by
1045 the target URI",
1046     "properties": {
1047         "bm": {
1048             "description": "Specifies the framework policies on the Resource
1049 referenced by the target URI for e.g. observable and discoverable",
1050             "type": "integer"
1051         }
1052     },
1053     "required": [
1054         "bm"
1055     ],
1056     "type": "object"
1057 },

```



```

1058         "rel": {
1059             "description": "The relation of the target URI referenced by the link to the
1060 context URI",
1061             "oneOf": [
1062                 {
1063                     "default": [
1064                         "hosts"
1065                     ],
1066                     "items": {
1067                         "maxLength": 64,
1068                         "type": "string"
1069                     },
1070                     "minItems": 1,
1071                     "type": "array"
1072                 },
1073                 {
1074                     "default": "hosts",
1075                     "maxLength": 64,
1076                     "type": "string"
1077                 }
1078             ]
1079         },
1080         "rt": {
1081             "description": "Resource Type of the Resource",
1082             "items": {
1083                 "maxLength": 64,
1084                 "type": "string"
1085             },
1086             "minItems": 1,
1087             "type": "array"
1088         },
1089         "title": {
1090             "description": "A title for the link relation. Can be used by the UI to
1091 provide a context.",
1092             "maxLength": 64,
1093             "type": "string"
1094         },
1095         "type": {
1096             "default": "application/cbor",
1097             "description": "A hint at the representation of the resource referenced by the
1098 target URI. This represents the media types that are used for both accepting and emitting.",
1099             "items": {
1100                 "maxLength": 64,
1101                 "type": "string"
1102             },
1103             "minItems": 1,
1104             "type": "array"
1105         }
1106     },
1107     "required": [
1108         "href",
1109         "rt",
1110         "if"
1111     ],
1112     "type": "object"
1113 },
1114 "type": "array"
1115 }
1116 ]
1117 }
1118 },
1119 "required": [
1120     "href",
1121     "rep"
1122 ],
1123 "type": "object"
1124 },
1125 "type": "array"
1126 },
1127

```

```

1128     "sbatch-update" : {
1129         "title" : "Collection Batch Update Format (auto merged)",
1130         "minItems" : 1,
1131         "items" : { "$ref": "#/definitions/oic.batch-update.item" },
1132         "type" : "array"
1133     },
1134     "EasySetup" : {
1135         "properties": {
1136             "rt" : {
1137                 "items": {
1138                     "enum": [
1139                         "oic.r.easysetup",
1140                         "oic.wk.col"
1141                     ]
1142                 },
1143                 "maxItems": 2,
1144                 "minItems": 2,
1145                 "type": "array",
1146                 "uniqueItems": true
1147             },
1148             "ps" : {
1149                 "description": "Indicates the easy setup status of the device. (0: Need to Setup, 1:
1150 Connecting to Enroller, 2: Connected to Enroller, 3: Failed to Connect to Enroller, 4~254: Reserved,
1151 255: EOF)",
1152                 "enum": [
1153                     0,
1154                     1,
1155                     2,
1156                     3
1157                 ],
1158                 "readOnly": true,
1159                 "type": "integer"
1160             },
1161             "lec" : {
1162                 "description": "Indicates a failure reason (0: NO error, 1: A given SSID is not found, 2:
1163 Wi-Fi's password is wrong, 3: IP address is not allocated, 4: No internet connection, 5: Timeout, 6:
1164 Wi-Fi Auth Type is not supported by the Enrollee, 7: Wi-Fi Encryption Type is not supported by the
1165 Enrollee, 8: Wi-Fi Auth Type is wrong (failure while connecting to the Enroller), 9: Wi-Fi
1166 Encryption Type is wrong (failure while connecting to the Enroller), 10~254: Reserved, 255: Unknown
1167 error)",
1168                 "enum": [
1169                     0,
1170                     1,
1171                     2,
1172                     3,
1173                     4,
1174                     5,
1175                     6,
1176                     7,
1177                     8,
1178                     9,
1179                     255
1180                 ],
1181                 "readOnly": true,
1182                 "type": "integer"
1183             },
1184             "cn" : {
1185                 "description": "Indicates an array of connection types that trigger an attempt to connect
1186 to the Enroller to start.",
1187                 "items": {
1188                     "description": "Connection type to attempt. (1 : Wi-Fi, 2 : other entities / transports
1189 to be added in future (e.g. Connect to cloud / BLE))",
1190                     "type": "integer"
1191                 },
1192                 "type": "array"
1193             },
1194             "links" : {
1195                 "description": "A set of simple or individual OIC Links.",
1196                 "items": {
1197                     "properties": {

```

```

1198         "anchor": {
1199             "description": "This is used to override the context URI e.g. override the URI of
1200 the containing collection.",
1201             "format": "uri",
1202             "maxLength": 256,
1203             "type": "string"
1204         },
1205         "di": {
1206             "description": "The Device ID formatted according to IETF RFC 4122.",
1207             "pattern": "^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-
1208 9]{12}$",
1209             "type": "string"
1210         },
1211         "eps": {
1212             "description": "the Endpoint information of the target Resource",
1213             "items": {
1214                 "properties": {
1215                     "ep": {
1216                         "description": "Transport Protocol Suite + Endpoint Locator",
1217                         "format": "uri",
1218                         "type": "string"
1219                     },
1220                     "pri": {
1221                         "description": "The priority among multiple Endpoints",
1222                         "minimum": 1,
1223                         "type": "integer"
1224                     }
1225                 },
1226                 "type": "object"
1227             },
1228             "type": "array"
1229         },
1230         "href": {
1231             "description": "This is the target URI, it can be specified as a Relative Reference
1232 or fully-qualified URI.",
1233             "format": "uri",
1234             "maxLength": 256,
1235             "type": "string"
1236         },
1237         "if": {
1238             "description": "The interface set supported by this resource",
1239             "items": {
1240                 "enum": [
1241                     "oic.if.baseline",
1242                     "oic.if.ll",
1243                     "oic.if.b",
1244                     "oic.if.rw",
1245                     "oic.if.r",
1246                     "oic.if.a",
1247                     "oic.if.s"
1248                 ],
1249                 "type": "string"
1250             },
1251             "minItems": 1,
1252             "type": "array"
1253         },
1254         "ins": {
1255             "description": "The instance identifier for this web link in an array of web links -
1256 used in collections",
1257             "type": "integer"
1258         },
1259         "p": {
1260             "description": "Specifies the framework policies on the Resource referenced by the
1261 target URI",
1262             "properties": {
1263                 "bm": {
1264                     "description": "Specifies the framework policies on the Resource referenced by
1265 the target URI for e.g. observable and discoverable",
1266                     "type": "integer"
1267                 }
1268             }
1269         }

```

```

1268     },
1269     "required": [
1270         "bm"
1271     ],
1272     "type": "object"
1273 },
1274 "rel": {
1275     "description": "The relation of the target URI referenced by the link to the context
1276 URI",
1277     "oneOf": [
1278         {
1279             "default": [
1280                 "hosts"
1281             ],
1282             "items": {
1283                 "maxLength": 64,
1284                 "type": "string"
1285             },
1286             "minItems": 1,
1287             "type": "array"
1288         },
1289         {
1290             "default": "hosts",
1291             "maxLength": 64,
1292             "type": "string"
1293         }
1294     ],
1295 },
1296 "rt": {
1297     "description": "Resource Type of the Resource",
1298     "items": {
1299         "maxLength": 64,
1300         "type": "string"
1301     },
1302     "minItems": 1,
1303     "type": "array"
1304 },
1305 "title": {
1306     "description": "A title for the link relation. Can be used by the UI to provide a
1307 context.",
1308     "maxLength": 64,
1309     "type": "string"
1310 },
1311 "type": {
1312     "default": "application/cbor",
1313     "description": "A hint at the representation of the resource referenced by the
1314 target URI. This represents the media types that are used for both accepting and emitting.",
1315     "items": {
1316         "maxLength": 64,
1317         "type": "string"
1318     },
1319     "minItems": 1,
1320     "type": "array"
1321 }
1322 },
1323 "required": [
1324     "href",
1325     "rt",
1326     "if"
1327 ],
1328 "type": "object"
1329 },
1330 "type": "array"
1331 },
1332 "n" : {
1333     "description": "Friendly name of the resource",
1334     "maxLength": 64,
1335     "readOnly": true,
1336     "type": "string"
1337 },

```

```

1338     "rts" : {
1339         "description": "Resource Type of the Resource",
1340         "items": {
1341             "maxLength": 64,
1342             "type": "string"
1343         },
1344         "minItems": 1,
1345         "readOnly": true,
1346         "type": "array"
1347     },
1348     "id" : {
1349         "description": "Instance ID of this specific resource",
1350         "maxLength": 64,
1351         "readOnly": true,
1352         "type": "string"
1353     },
1354     "rts-m" : {
1355         "description": "Resource Type of the Resource",
1356         "items": {
1357             "maxLength": 64,
1358             "type": "string"
1359         },
1360         "minItems": 1,
1361         "readOnly": true,
1362         "type": "array"
1363     },
1364     "if" : {
1365         "description": "The interface set supported by this resource",
1366         "items": {
1367             "enum": [
1368                 "oic.if.baseline",
1369                 "oic.if.ll",
1370                 "oic.if.b",
1371                 "oic.if.lb",
1372                 "oic.if.rw",
1373                 "oic.if.r",
1374                 "oic.if.a",
1375                 "oic.if.s"
1376             ],
1377             "type": "string"
1378         },
1379         "minItems": 1,
1380         "readOnly": true,
1381         "type": "array"
1382     }
1383 },
1384 "type" : "object",
1385 "required": ["ps", "lec", "cn"]
1386 },
1387 "oic.batch-update.item" : {
1388     "additionalProperties": true,
1389     "description": "array of resource representations to apply to the batch collection, using href
1390 to indicate which resource(s) in the batch to update. If the href property is empty, effectively
1391 making the URI reference to the collection itself, the representation is to be applied to all
1392 resources in the batch",
1393     "properties": {
1394         "href": {
1395             "description": "URI of the target resource relative assuming the collection URI as
1396 anchor",
1397             "format": "uri",
1398             "maxLength": 256,
1399             "type": "string"
1400         },
1401         "rep": {
1402             "oneOf": [
1403                 {
1404                     "description": "The response payload from a single resource",
1405                     "type": "object"
1406                 },
1407                 {

```

```

1408         "description": " The response payload from a collection (batch) resource",
1409         "items": {
1410             "$ref": "#/definitions/oic.oic-link"
1411         },
1412         "type": "array"
1413     }
1414 ]
1415 }
1416 },
1417 "required": [
1418     "href",
1419     "rep"
1420 ],
1421 "type": "object"
1422 },
1423 "oic.oic-link" : {
1424     "properties": {
1425         "anchor": {
1426             "description": "This is used to override the context URI e.g. override the URI of the
1427 containing collection.",
1428             "format": "uri",
1429             "maxLength": 256,
1430             "type": "string"
1431         },
1432         "di": {
1433             "allOf": [
1434                 {
1435                     "$ref": "#/definitions/uuid"
1436                 },
1437                 {
1438                     "description": "The device ID"
1439                 }
1440             ]
1441         },
1442         "eps": {
1443             "description": "the Endpoint information of the target Resource",
1444             "items": {
1445                 "properties": {
1446                     "ep": {
1447                         "description": "Transport Protocol Suite + Endpoint Locator",
1448                         "format": "uri",
1449                         "type": "string"
1450                     },
1451                     "pri": {
1452                         "description": "The priority among multiple Endpoints",
1453                         "minimum": 1,
1454                         "type": "integer"
1455                     }
1456                 },
1457                 "type": "object"
1458             },
1459             "type": "array"
1460         },
1461         "href": {
1462             "description": "This is the target URI, it can be specified as a Relative Reference or
1463 fully-qualified URI.",
1464             "format": "uri",
1465             "maxLength": 256,
1466             "type": "string"
1467         },
1468         "if": {
1469             "description": "The interface set supported by this resource",
1470             "items": {
1471                 "enum": [
1472                     "oic.if.baseline",
1473                     "oic.if.ll",
1474                     "oic.if.b",
1475                     "oic.if.rw",
1476                     "oic.if.r",
1477                     "oic.if.a",

```

```

1478         "oic.if.s"
1479     ],
1480     "type": "string"
1481 },
1482     "minItems": 1,
1483     "type": "array"
1484 },
1485     "ins": {
1486         "description": "The instance identifier for this web link in an array of web links - used
1487 in collections",
1488         "type": "integer"
1489     },
1490     "p": {
1491         "description": "Specifies the framework policies on the Resource referenced by the target
1492 URI",
1493         "properties": {
1494             "bm": {
1495                 "description": "Specifies the framework policies on the Resource referenced by the
1496 target URI for e.g. observable and discoverable",
1497                 "type": "integer"
1498             }
1499         },
1500         "required": [
1501             "bm"
1502         ],
1503         "type": "object"
1504     },
1505     "rel": {
1506         "description": "The relation of the target URI referenced by the link to the context URI",
1507         "oneOf": [
1508             {
1509                 "default": [
1510                     "hosts"
1511                 ],
1512                 "items": {
1513                     "maxLength": 64,
1514                     "type": "string"
1515                 },
1516                 "minItems": 1,
1517                 "type": "array"
1518             },
1519             {
1520                 "default": "hosts",
1521                 "maxLength": 64,
1522                 "type": "string"
1523             }
1524         ]
1525     },
1526     "rt": {
1527         "description": "Resource Type of the Resource",
1528         "items": {
1529             "maxLength": 64,
1530             "type": "string"
1531         },
1532         "minItems": 1,
1533         "type": "array"
1534     },
1535     "title": {
1536         "description": "A title for the link relation. Can be used by the UI to provide a
1537 context.",
1538         "maxLength": 64,
1539         "type": "string"
1540     },
1541     "type": {
1542         "default": "application/cbor",
1543         "description": "A hint at the representation of the resource referenced by the target URI.
1544 This represents the media types that are used for both accepting and emitting.",
1545         "items": {
1546             "maxLength": 64,
1547             "type": "string"

```

```

1548     },
1549     "minItems": 1,
1550     "type": "array"
1551   }
1552 },
1553 "required": [
1554   "href",
1555   "rt",
1556   "if"
1557 ],
1558 "type": "object"
1559 },
1560 "uuid" : {
1561   "description": "Format pattern according to IETF RFC 4122.",
1562   "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}$",
1563   "type": "string"
1564 },
1565 "oic.wk.col-batch-update" : {
1566   "description": "array of resource representations to apply to the batch collection, using href
1567 to indicate which resource(s) in the batch to update. If the href property is empty, effectively
1568 making the URI reference to the collection itself, the representation is to be applied to all
1569 resources in the batch",
1570   "items": {
1571     "$ref": "#/definitions/oic.batch-update.item"
1572   },
1573   "minItems": 1,
1574   "type": "array"
1575 }
1576 }
1577 }
1578

```

1579 A.3.5 Property definition

1580 Table A.4 defines the Properties that are part of the ['oic.r.easyssetup', 'oic.wk.col'] Resource Type

1581 **Table A.4 – The Property definitions of the Resource with type 'rt' = ['oic.r.easyssetup',**
1582 **'oic.wk.col']**

Property name	Value type	Mandatory	Access mode	Description
p	object: see schema	No	Read Write	Specifies the framework policies on the Resource referenced by the target URI
anchor	string	No	Read Write	This is used to override the context URI e.g. override the URI of the containing collection.
rel	multiple types: see schema	No	Read Write	The relation of the target URI referenced by the link to the context URI
title	string	No	Read Write	A title for the link relation. Can be used by the UI to provide a context.

href	string	Yes	Read Write	This is the target URI, it can be specified as a Relative Reference or fully-qualified URI.
if	array: schema see	Yes	Read Write	The interface set supported by this resource
ins	integer	No	Read Write	The instance identifier for this web link in an array of web links - used in collections
rt	array: schema see	Yes	Read Write	Resource Type of the Resource
type	array: schema see	No	Read Write	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.
di	string	No	Read Write	The Device ID formatted according to IETF RFC 4122.
eps	array: schema see	No	Read Write	the Endpoint information of the target Resource
p	object: schema see	No	Read Write	Specifies the framework policies on the Resource referenced by the target URI
anchor	string	No	Read Write	This is used to override the context URI e.g. override the URI of the containing collection.
rel	multiple types: see schema	No	Read Write	The relation of the target URI referenced by the link to the context URI

title	string	No	Read Write	A title for the link relation. Can be used by the UI to provide a context.
href	string	Yes	Read Write	This is the target URI, it can be specified as a Relative Reference or fully-qualified URI.
if	array: see schema	Yes	Read Write	The interface set supported by this resource
ins	integer	No	Read Write	The instance identifier for this web link in an array of web links - used in collections
rt	array: see schema	Yes	Read Write	Resource Type of the Resource
type	array: see schema	No	Read Write	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.
di	multiple types: see schema	No	Read Write	
eps	array: see schema	No	Read Write	the Endpoint information of the target Resource
href	string	Yes	Read Write	URI of the target resource relative assuming the collection URI as anchor
rep	multiple types: see schema	Yes	Read Write	
rt	array: see schema	No	Read Write	
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)
n	string	No	Read Only	Friendly name of the resource
rts-m	array: see schema	No	Read Only	Resource Type of the Resource
links	array: see schema	No	Read Write	A set of simple or individual OIC Links.
if	array: see schema	No	Read Only	The interface set supported by this resource
id	string	No	Read Only	Instance ID of this specific resource
cn	array: see schema	Yes	Read Write	Indicates an array of connection types that trigger an attempt to connect to the Enroller to start.
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)
rts	array: see schema	No	Read Only	Resource Type of the Resource
href	string	Yes	Read Write	URI of the target resource relative assuming the collection URI as anchor
rep	multiple types: see schema	Yes	Read Write	

1583 **A.3.6 CRUDN behaviour**

1584 Table A.5 defines the CRUDN operations that are supported on the ['oic.r.easyssetup', 'oic.wk.col']
 1585 Resource Type

1586 **Table A.5 – The CRUDN operations of the Resource with type 'rt' = ['oic.r.easyssetup',**
 1587 **'oic.wk.col']**

Create	Read	Update	Delete	Notify
	get			observe

1588 **A.4 Wi-Fi Configuration Resource Baseline Interface**

1589 **A.4.1 Introduction**

1590 WiFiConf resource stores essential information to help an unboxing device
 1591 to connect to an existing Wi-Fi AP.

1593 **A.4.2 Example URI**

1594 /WiFiConfResURI

1595 **A.4.3 Resource type**

1596 The resource type (rt) is defined as: ['oic.r.wificonf'].

1597 **A.4.4 OpenAPI 2.0 definition**

```

1598 {
1599   "swagger": "2.0",
1600   "info": {
1601     "title": "Wi-Fi Configuration Resource Baseline Interface",
1602     "version": "v0.0.3-20170611",
1603     "license": {
1604       "name": "copyright 2016-2019 Open Connectivity Foundation, Inc. All rights reserved.",
1605       "x-description": "Redistribution and use in source and binary forms, with or without
1606 modification, are permitted provided that the following conditions are met:\n      1.
1607 Redistributions of source code must retain the above copyright notice, this list of conditions and
1608 the following disclaimer.\n      2. Redistributions in binary form must reproduce the above
1609 copyright notice, this list of conditions and the following disclaimer in the documentation and/or
1610 other materials provided with the distribution.\n\n      THIS SOFTWARE IS PROVIDED BY THE Open
1611 Connectivity Foundation, INC. \\"AS IS\\" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
1612 LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR
1613 WARRANTIES OF NON-INFRINGEMENT, ARE DISCLAIMED.\n      IN NO EVENT SHALL THE Open Connectivity
1614 Foundation, INC. OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY,
1615 OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
1616 SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)\n      HOWEVER CAUSED AND ON
1617 ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
1618 OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
  
```

```

1619 SUCH DAMAGE.\n"
1620 }
1621 },
1622 "schemes": ["http"],
1623 "consumes": ["application/json"],
1624 "produces": ["application/json"],
1625 "paths": {
1626   "/WiFiConfResURI?if=oic.if.baseline" : {
1627     "get": {
1628       "description": "WiFiConf resource stores essential information to help an unboxing
1629 device\nto connect to an existing Wi-Fi AP.\nRetrieve properties of WiFiConf resource.\nThe
1630 information includes :\n1. Wi-Fi SSID and password\n2. Wi-Fi Security type (i.e. auth type and
1631 encryption type)\n3. Wi-Fi hardware capability (i.e. supported frequencies, modes,\n auth types and
1632 encryption types)\n",
1633       "parameters": [
1634         {"$ref": "#/parameters/interface-baseline"}
1635       ],
1636       "responses": {
1637         "200": {
1638           "description": "",
1639           "x-example":
1640             {
1641               "rt": ["oic.r.wificonf"],
1642               "swmt": ["A", "B", "G"],
1643               "swf": ["2.4G", "5G"],
1644               "tnn": "Home_AP_SSID",
1645               "cd": "Home_AP_PWD",
1646               "wat": "WPA2_PSK",
1647               "wet": "TKIP",
1648               "swat": ["WPA_PSK", "WPA2_PSK"],
1649               "swet": ["TKIP", "AES", "TKIP_AES"]
1650             },
1651           "schema": { "$ref": "#/definitions/WiFiConf" }
1652         }
1653       }
1654     },
1655     "post": {
1656       "description": "Deliver Wi-Fi AP's information for an unboxing device to connect to it.\n",
1657       "parameters": [
1658         {"$ref": "#/parameters/interface-baseline"},
1659         {
1660           "name": "body",
1661           "in": "body",
1662           "required": true,
1663           "schema": { "$ref": "#/definitions/WiFiConfUpdate" },
1664           "x-example":
1665             {
1666               "tnn": "Home_AP_SSID",
1667               "cd": "Home_AP_PWD",
1668               "wat": "WPA2_PSK",
1669               "wet": "AES"
1670             }
1671         }
1672       ],
1673       "responses": {
1674         "200": {
1675           "description": "",
1676           "x-example":
1677             {
1678               "tnn": "Home_AP_SSID",
1679               "cd": "Home_AP_PWD",
1680               "wat": "WPA2_PSK",
1681               "wet": "AES"
1682             },
1683           "schema": { "$ref": "#/definitions/WiFiConfUpdate" }
1684         }
1685       }
1686     }
1687   },
1688   "/WiFiConfResURI?if=oic.if.rw" : {

```

```

1689     "get": {
1690         "description": "WiFiConf resource stores essential information to help an unboxing
1691 device\nto connect to an existing Wi-Fi AP.\nRetrieve properties of WiFiConf resource that can be
1692 updated by a client.\n",
1693         "parameters": [
1694             { "$ref": "#/parameters/interface-rw" }
1695         ],
1696         "responses": {
1697             "200": {
1698                 "description": "",
1699                 "x-example":
1700                 {
1701                     "tnn": "Home_AP_SSID",
1702                     "cd": "Home_AP_PWD",
1703                     "wat": "WPA2_PSK",
1704                     "wet": "AES"
1705                 }
1706             },
1707             "schema": { "$ref": "#/definitions/WiFiConfUpdate" }
1708         }
1709     },
1710 },
1711 "post": {
1712     "description": "Deliver Wi-Fi AP's information for an unboxing device to connect to it.\n",
1713     "parameters": [
1714         { "$ref": "#/parameters/interface-rw" },
1715         {
1716             "name": "body",
1717             "in": "body",
1718             "required": true,
1719             "schema": { "$ref": "#/definitions/WiFiConfUpdate" },
1720             "x-example":
1721             {
1722                 "tnn": "Home_AP_SSID",
1723                 "cd": "Home_AP_PWD",
1724                 "wat": "WPA2_PSK",
1725                 "wet": "AES"
1726             }
1727         }
1728     ],
1729     "responses": {
1730         "200": {
1731             "description": "",
1732             "x-example":
1733             {
1734                 "tnn": "Home_AP_SSID",
1735                 "cd": "Home_AP_PWD",
1736                 "wat": "WPA2_PSK",
1737                 "wet": "AES"
1738             },
1739             "schema": { "$ref": "#/definitions/WiFiConfUpdate" }
1740         }
1741     }
1742 },
1743 },
1744 },
1745 "parameters": {
1746     "interface-rw" : {
1747         "in" : "query",
1748         "name" : "if",
1749         "type" : "string",
1750         "enum" : ["oic.if.rw"]
1751     },
1752     "interface-baseline" : {
1753         "in" : "query",
1754         "name" : "if",
1755         "type" : "string",
1756         "enum" : ["oic.if.baseline"]
1757     },
1758     "interface-all" : {

```

```

1759     "in" : "query",
1760     "name" : "if",
1761     "type" : "string",
1762     "enum" : ["oic.if.baseline", "oic.if.rw"]
1763   }
1764 },
1765 "definitions": {
1766   "WiFiConf" : {
1767     "properties": {
1768       "rt" : {
1769         "description": "Resource Type of the Resource",
1770         "items": {
1771           "maxLength": 64,
1772           "type": "string"
1773         },
1774         "minItems": 1,
1775         "readOnly": true,
1776         "type": "array"
1777       },
1778       "tnn" : {
1779         "description": "Indicates Target Network Name (SSID of Wi-Fi AP)",
1780         "pattern": "^.*$",
1781         "type": "string"
1782       },
1783       "swmt" : {
1784         "description": "Indicates supported Wi-Fi mode types. It can be multiple",
1785         "items": {
1786           "description": "Supported Wi-Fi Mode Type.",
1787           "enum": [
1788             "A",
1789             "B",
1790             "G",
1791             "N",
1792             "AC"
1793           ],
1794           "type": "string"
1795         },
1796         "readOnly": true,
1797         "type": "array"
1798       },
1799       "wat" : {
1800         "description": "Indicates Wi-Fi Auth Type",
1801         "enum": [
1802           "None",
1803           "WEP",
1804           "WPA_PSK",
1805           "WPA2_PSK"
1806         ],
1807         "type": "string"
1808       },
1809       "n" : {
1810         "description": "Friendly name of the resource",
1811         "maxLength": 64,
1812         "readOnly": true,
1813         "type": "string"
1814       },
1815       "swat" : {
1816         "description": "Indicates supported Wi-Fi Auth types. It can be multiple",
1817         "items": {
1818           "description": "Indicates Wi-Fi Auth Type",
1819           "enum": [
1820             "None",
1821             "WEP",
1822             "WPA_PSK",
1823             "WPA2_PSK"
1824           ],
1825           "type": "string"
1826         },
1827         "readOnly": true,
1828         "type": "array"

```

```

1829     },
1830     "swf" : {
1831         "description": "Indicates Supported Wi-Fi frequencies by the Enrollee. Can be multiple.
1832 Valid values are ('2.4G', '5G')",
1833         "items": {
1834             "pattern": "^(2\\.4|5)G$",
1835             "type": "string"
1836         },
1837         "readOnly": true,
1838         "type": "array"
1839     },
1840     "swet" : {
1841         "description": "Indicates supported Wi-Fi Encryption types. It can be multiple",
1842         "items": {
1843             "description": "Indicates Wi-Fi Encryption Type",
1844             "enum": [
1845                 "None",
1846                 "WEP_64",
1847                 "WEP_128",
1848                 "TKIP",
1849                 "AES",
1850                 "TKIP_AES"
1851             ],
1852             "type": "string"
1853         },
1854         "readOnly": true,
1855         "type": "array"
1856     },
1857     "wet" : {
1858         "description": "Indicates Wi-Fi Encryption Type",
1859         "enum": [
1860             "None",
1861             "WEP_64",
1862             "WEP_128",
1863             "TKIP",
1864             "AES",
1865             "TKIP_AES"
1866         ],
1867         "type": "string"
1868     },
1869     "cd" : {
1870         "description": "Indicates credential information of Wi-Fi AP",
1871         "pattern": "^.*$",
1872         "type": "string"
1873     },
1874     "id" : {
1875         "description": "Instance ID of this specific resource",
1876         "maxLength": 64,
1877         "readOnly": true,
1878         "type": "string"
1879     },
1880     "if" : {
1881         "description": "The interface set supported by this resource",
1882         "items": {
1883             "enum": [
1884                 "oic.if.baseline",
1885                 "oic.if.ll",
1886                 "oic.if.b",
1887                 "oic.if.lb",
1888                 "oic.if.rw",
1889                 "oic.if.r",
1890                 "oic.if.a",
1891                 "oic.if.s"
1892             ],
1893             "type": "string"
1894         },
1895         "minItems": 1,
1896         "readOnly": true,
1897         "type": "array"
1898     }

```



```

1899     },
1900     "type" : "object",
1901     "required":["swmt", "swf", "swat", "swet", "tnn", "wat", "wet"]
1902 },
1903 "WiFiConfUpdate" : {
1904   "properties": {
1905     "wat" : {
1906       "description": "Indicates Wi-Fi Auth Type",
1907       "enum": [
1908         "None",
1909         "WEP",
1910         "WPA_PSK",
1911         "WPA2_PSK"
1912       ]
1913     },
1914     "cd" : {
1915       "description": "Indicates credential information of Wi-Fi AP",
1916       "pattern": "^.*$",
1917       "type": "string"
1918     },
1919     "wet" : {
1920       "description": "Indicates Wi-Fi Encryption Type",
1921       "enum": [
1922         "None",
1923         "WEP_64",
1924         "WEP_128",
1925         "TKIP",
1926         "AES",
1927         "TKIP_AES"
1928       ]
1929     },
1930     "tnn" : {
1931       "description": "Indicates Target Network Name (SSID of Wi-Fi AP)",
1932       "pattern": "^.*$",
1933       "type": "string"
1934     }
1935   },
1936   "type" : "object",
1937   "required":["tnn", "wat", "wet"]
1938 }
1939 }
1940 }
1941

```

1942 A.4.5 Property definition

1943 Table A.6 defines the Properties that are part of the ['oic.r.wificonf'] Resource Type

1944 **Table A.6 – The Property definitions of the Resource with type 'rt' = ['oic.r.wificonf']**

Property name	Value type	Mandatory	Access mode	Description
swmt	array: see schema	Yes	Read Only	Indicates supported Wi-Fi mode types. It can be multiple
swf	array: see schema	Yes	Read Only	Indicates Supported Wi-Fi frequencies by the Enrollee. Can be multiple. Valid values are ('2.4G', '5G')
cd	string	No	Read Write	Indicates credential information of Wi-Fi AP

wet	string	Yes	Read Write	Indicates Wi-Fi Encryption Type
swat	array: schema see	Yes	Read Only	Indicates supported Wi-Fi Auth types. It can be multiple
if	array: schema see	No	Read Only	The interface set supported by this resource
swet	array: schema see	Yes	Read Only	Indicates supported Wi-Fi Encryption types. It can be multiple
rt	array: schema see	No	Read Only	Resource Type of the Resource
n	string	No	Read Only	Friendly name of the resource
wat	string	Yes	Read Write	Indicates Wi-Fi Auth Type
id	string	No	Read Only	Instance ID of this specific resource
tnn	string	Yes	Read Write	Indicates Target Network Name (SSID of Wi-Fi AP)
cd	string	No	Read Write	Indicates credential information of Wi-Fi AP
wet	multiple types: see schema	Yes	Read Write	Indicates Wi-Fi Encryption Type
wat	multiple types: see schema	Yes	Read Write	Indicates Wi-Fi Auth Type
tnn	string	Yes	Read Write	Indicates Target Network Name (SSID of Wi-Fi AP)

1945 **A.4.6 CRUDN behaviour**

1946 Table A.7 defines the CRUDN operations that are supported on the ['oic.r.wificonf'] Resource Type

1947 **Table A.7 – The CRUDN operations of the Resource with type 'rt' = ['oic.r.wificonf']**

Create	Read	Update	Delete	Notify
	get	post		observe

1948

1949