

OCF Wi-Fi Easy Setup Specification

VERSION 2.0.3 | June 2019



OPEN CONNECTIVITY
FOUNDATION™

CONTACT admin@openconnectivity.org

Copyright Open Connectivity Foundation, Inc. © 2019
All Rights Reserved.

Legal Disclaimer

2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19

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

20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

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

61	A.2.2	Example URI	16
62	A.2.3	Resource type	16
63	A.2.4	OpenAPI 2.0 definition.....	16
64	A.2.5	Property definition	18
65	A.2.6	CRUDN behaviour	18
66	A.3	Easy Setup Collection	19
67	A.3.1	Introduction	19
68	A.3.2	Example URI	19
69	A.3.3	Resource type	19
70	A.3.4	OpenAPI 2.0 definition.....	19
71	A.3.5	Property definition	28
72	A.3.6	CRUDN behaviour	30
73	A.4	Wi-Fi Configuration	30
74	A.4.1	Introduction	30
75	A.4.2	Example URI	30
76	A.4.3	Resource type	30
77	A.4.4	OpenAPI 2.0 definition.....	30
78	A.4.5	Property definition	35
79	A.4.6	CRUDN behaviour	36
80			
81			

82
83
84
85
86
87
88
89
90
91

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

92		
93		
94	Table 1 – EasySetup Resource Type	5
95	Table 2 – "oic.r.easyssetup" Resource Type definition.....	6
96	Table 3 – WiFiConf Resource Type.....	7
97	Table 4 – "oic.r.wificonf" Resource Type definition.....	7
98	Table 5 – DevConf Resource Type	8
99	Table 6 – "oic.r.devconf" Resource Type definition	8
100	Table 7 – Easy Setup Information Element TLVs	13
101	Table A.1 – Alphabetized list of resources	16
102	Table A.2 – The Property definitions of the Resource with type "rt" = "oic.r.devconf".	18
103	Table A.3 – The CRUDN operations of the Resource with type "rt" = "oic.r.devconf".	18
104	Table A.4 – The Property definitions of the Resource with type "rt" = "oic.r.easyssetup,	
105	oic.wk.col".	28
106	Table A.5 – The CRUDN operations of the Resource with type "rt" = "oic.r.easyssetup,	
107	oic.wk.col".	30
108	Table A.6 – The Property definitions of the Resource with type "rt" = "oic.r.wificonf".	35
109	Table A.7 – The CRUDN operations of the Resource with type "rt" = "oic.r.wificonf".	36
110		
111		

112 **1 Scope**

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

116 **2 Normative references**

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

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

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

129 ISO/IEC 30118-5:2018 Information technology -- Open Connectivity Foundation (OCF)
130 Specification -- Part 5: Smart home device specification
131 <https://www.iso.org/standard/74242.html>
132 Latest version available at: https://openconnectivity.org/specs/OCF_Device_Specification.pdf

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

138 IETF RFC 5646, *Tags for Identifying Languages*, September 2009
139 <https://www.rfc-editor.org/info/rfc5646>

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

142

143 **3 Terms, definitions, and abbreviated terms**

144 **3.1 Terms and definitions**

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

147 ISO and IEC maintain terminological databases for use in standardization at the following
148 addresses:

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

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

151 **3.1.1**

152 **Easy Setup**

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

155 **3.1.2**

156 **Easy Setup Enrollment**

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

159 **3.1.3**

160 **Enrollee**

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

162 **3.1.4**

163 **Enroller**

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

165 **3.1.5**

166 **Mediator**

167 logical function that enables the Enrollee (3.1.3) to connect to the target network (i.e. Enroller
168 (3.1.4))

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

170 **3.2 Abbreviated terms**

171 **3.2.1**

172 **CID**

173 Company Identifier (ID)

174 **3.2.2**

175 **IE**

176 Information Element

177 **3.2.3**

178 **Soft AP**

179 Software Enabled Access Point

180 **3.2.4**

181 **TLV**

182 type-length-value

183 **4 Document conventions and organization**

184 **4.1 Conventions**

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

189 **4.2 Notation**

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

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

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

196 Recommended (or should)(S).

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

203 Allowed (may or allowed)(O).

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

207 DEPRECATED.

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

214 Conditionally allowed (CA)

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

217 Conditionally required (CR)

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

221

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

223 Words that are emphasized are printed in italic.

224 **5 Overview**

225 **5.1 Introduction**

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

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

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

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

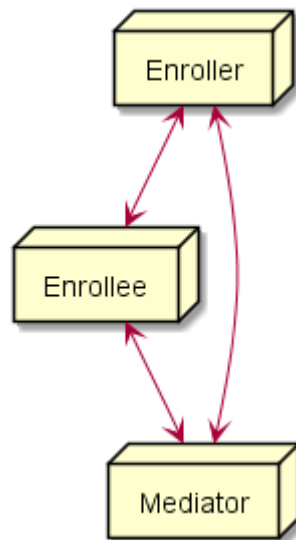
235 Easy Setup can be enhanced in future by incorporating other suitable technologies.

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

239 **5.2 Architecture**

240 Figure 1 shows the deployment architectural approach.

241



242

243

Figure 1 – Easy Setup deployment architecture

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

246 **5.3 Example Scenario**

247 The following scenario presents a typical setup case.

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

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

256 **6 Resource model**

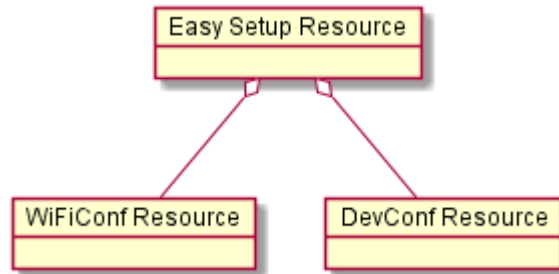
257 **6.1 Introduction**

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

- 259 1) EasySetup Resource Type
- 260 2) WiFiConf Resource Type
- 261 3) DevConf Resource Type

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

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



267
 268

Figure 2 – Easy Setup Resource Types

269 **6.2 EasySetup Resource**

270 **6.2.1 Overview**

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

273 **6.2.2 Resource**

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

275

Table 1 – EasySetup Resource Type

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

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

276

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

278

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.

279

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

- 281 – "ps" equal to 0.
- 282 – "lec" equal to 0.
- 283 – "cn" equal to an empty array.

284 6.3 WiFiConf Resource Type

285 6.3.1 Introduction

286 The WiFiConf Resource Type stores information to help an Enrollee to connect to an existing Wi-Fi AP.
287

288 6.3.2 Resource Type

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

290 **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

291
292 Table 4 defines the details for the "oic.r.wificonf" Resource Type.

293 **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")

294

295 6.4 DevConf Resource Type

296 6.4.1 Introduction

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

298 6.4.2 Resource Type

299 The DevConf Resource Type is as defined in Table 5

300

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

301

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

303

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.

304

305 **7 Network and connectivity**

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

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

309 **8 Functional interactions**

310 **8.1 Onboarding, Provisioning and Configuration**

311 The Mediator may be present as a standalone function or in conjunction with other functions or
312 services such as AMS as part of an OBT (Onboarding Tool); please refer to the ISO/IEC 30118-
313 2:2018.

314 **8.2 Resource discovery**

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

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

319 **8.3 Retrieving and Updating Easy Setup Resources**

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

323 Easy Setup Resources shall expose only secure Endpoints (e.g. CoAPS); see the ISO/IEC 30118-
324 1:2018, clause 10.

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

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

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

333 **8.4 Error Handling**

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

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

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

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

350 **8.5 Example Easy Setup Flow**

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

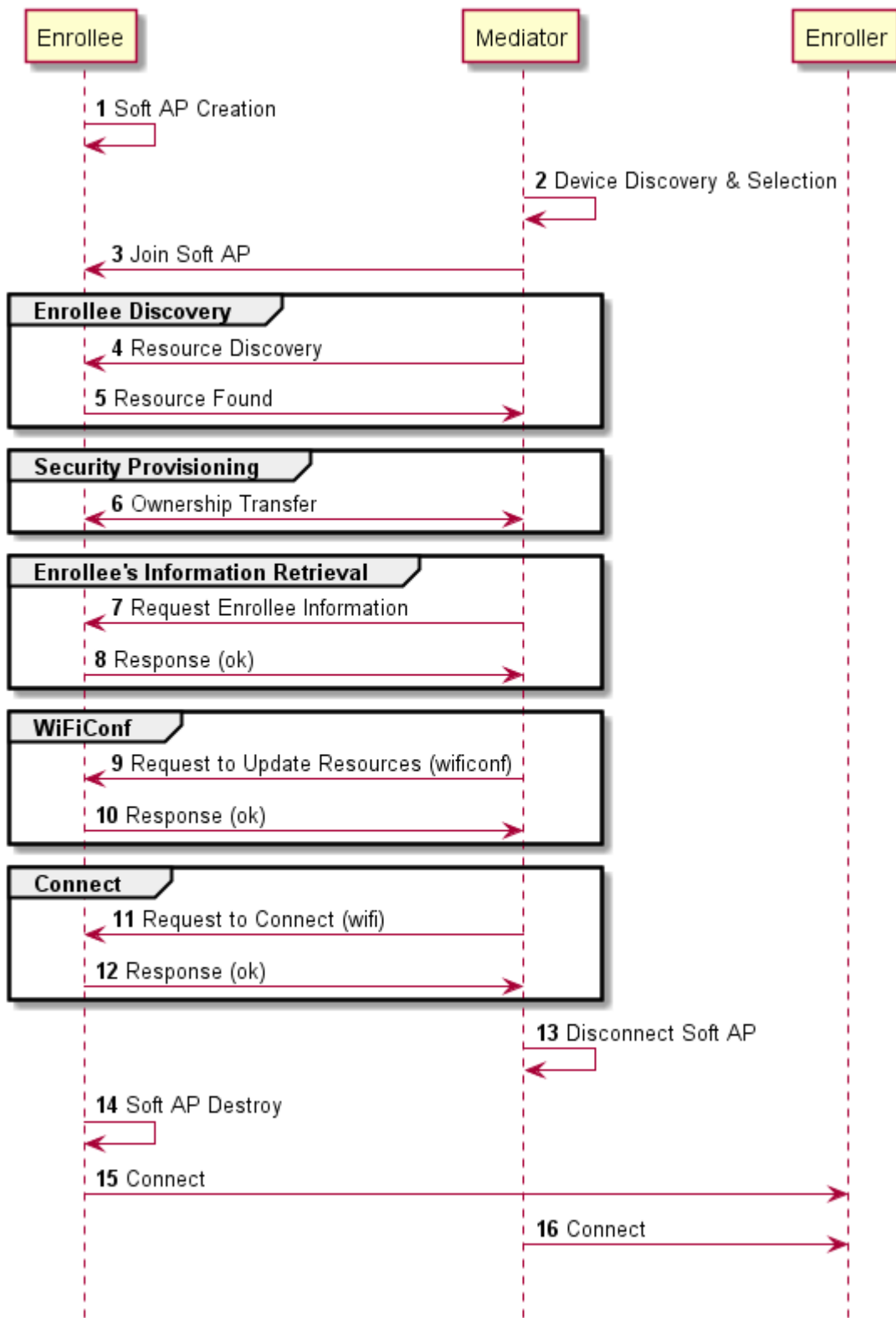


Figure 3 – Easy Setup Flow (Informative)

352

353

354

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

358 **8.6 Easy Setup SSID Tags**

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

- 361 – "OCF_"
 - 362 – Prefix tag that has to be at the beginning of the SSID.
 - 363 – Example: OCF_MySSID
- 364 – "_OCF"
 - 365 – Suffix tag that has to be at the end of the SSID.
 - 366 – Example: MySSID_OCF

367 These tags are case sensitive.

368 **8.7 Easy Setup Information Element**

369 **8.7.1 Overview**

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

375 **8.7.2 OCF Device Information Element (IE)**

376 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

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

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

388

389 Each TLV has the structure shown in Figure 5.

1 byte	1 byte	<250 bytes
Type	Length	Value

Figure 5 – Type-Length-Value Structure

- 390
- 391 – Type shall indicate the type of the field from Table 7.
 - 392 – Length shall indicate the length of the Value in bytes.
 - 393 – Value shall represent the corresponding information for specific TLV type from Table 7.
- 394 Data is a set of TLVs as defined in Table 7.

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

396

397 The TLVs may be set in any order inside an IE or IEC. All strings shall be UTF-8 encoded and shall
 398 not include a null terminator. All TLVs in Table 7 with a required value of "Y" shall be included in
 399 the IE or IEC (if multiple IEs are required). The value of each TLV shall meet the length
 400 requirements specified in Table 1.

401 **8.7.2.1 Device Friendly Name (Type 1)**

402 User readable string representing the friendly name of the device that is beaconing and ready to
 403 undergo Easy Setup. This should match "n" from "oic.wk.d" as defined in the ISO/IEC 30118-
 404 1:2018.

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

406 **8.7.2.2 Device Type (Type 2)**

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

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

411 In cases where the device supports multiple functions, several type 2 TLVs may be included to
 412 represent each function of the device.
 413

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

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

419 **8.7.2.3 Device Manufacturer Name (Type 3)**

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

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

424 **8.7.2.4 Language Tag (Type 4)**

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

429 Please see 8.7.2.8 for information on supporting multiple languages.

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

432 **8.7.2.5 Protocol Independent ID (Type 5)**

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

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

435 **8.7.2.6 Multiple Information Elements**

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

- 437 – The total size of the TLVs is larger than the size of Data as defined in an Easy Setup Information
438 Element.
- 439 – Support for multiple languages is necessary.

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

442 **8.7.2.7 IEC for Large TLV Size Support**

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

- 446 – Type
- 447 – Length
- 448 – CID
- 449 – OCF IE Type

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

452 **8.7.2.8 IEC for Multiple Language Support**

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

457 **9 Security**

458 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.2
Easy Setup		"oic.r.easyssetup"	A.3
Wi-Fi Configuration		"oic.r.wificonf"	A.4

A.2 Device Configuration

A.2.1 Introduction

The Device configuration Resource stores Device settings such as the Device name. Vendor-specific information can be added to the Resource. The Device name is a human-friendly name read by a Mediator during easy setup.

A.2.2 Example URI

/example/DevConfResURI

A.2.3 Resource type

The Resource Type is defined as: "oic.r.devconf".

A.2.4 OpenAPI 2.0 definition

```
{
  "swagger": "2.0",
  "info": {
    "title": "Device Configuration",
    "version": "20190306",
    "license": {
      "name": "OCF Data Model License",
      "url":
"https://github.com/openconnectivityfoundation/core/blob/e28a9e0a92e17042ba3e83661e4c0fbce8bdc4ba/LI
CENSE.md",
      "x-copyright": "Copyright 2018-2019 Open Connectivity Foundation, Inc. All rights reserved."
    },
    "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
  },
  "schemes": ["http"],
  "consumes": ["application/json"],
  "produces": ["application/json"],
  "paths": {
    "/example/DevConfResURI" : {
      "get": {
        "description": "The Device configuration Resource stores Device settings such as the Device
name. Vendor-specific information can be added to the Resource.\nThe Device name is a human-friendly
name read by a Mediator during easy setup.\n",
        "parameters": [
          {"$ref": "#/parameters/interface"}
        ]
      }
    }
  }
}
```

```

502     ],
503     "responses": {
504         "200": {
505             "description" : "",
506             "x-example": {
507                 "rt": ["oic.r.devconf"],
508                 "dn" : "My Refrigerator"
509             },
510             "schema": { "$ref": "#/definitions/DevConf" }
511         }
512     }
513 }
514 }
515 },
516 "parameters": {
517     "interface" : {
518         "in" : "query",
519         "name" : "if",
520         "type" : "string",
521         "enum" : ["oic.if.r", "oic.if.baseline"]
522     }
523 },
524 "definitions": {
525     "DevConf" : {
526         "properties": {
527             "rt" : {
528                 "description": "Resource Type of the Resource",
529                 "items": {
530                     "enum": ["oic.r.devconf"],
531                     "maxLength": 64,
532                     "type": "string"
533                 },
534                 "minItems": 1,
535                 "readOnly": true,
536                 "uniqueItems": true,
537                 "type": "array"
538             },
539             "n" : {
540                 "$ref":
541 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
542 schema.json#/definitions/n"
543             },
544             "id" : {
545                 "$ref":
546 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
547 schema.json#/definitions/id"
548             },
549             "if" : {
550                 "description": "The OCF Interfaces supported by this Resource",
551                 "items": {
552                     "enum": [
553                         "oic.if.r",
554                         "oic.if.baseline"
555                     ],
556                     "type": "string",
557                     "maxLength": 64
558                 },
559                 "minItems": 2,
560                 "readOnly": true,
561                 "uniqueItems": true,
562                 "type": "array"
563             },
564             "dn": {
565                 "oneOf": [
566                     {
567                         "type": "string",
568                         "description": "Indicates a pre-configured Device name in language indicated by 'dl'
569 in /oic/con; presented by an Enrollee Device to a Mediator Device during the easy-setup process",
570                         "pattern": "^.*$",
571                         "readOnly": true

```

```

572     },
573     {
574         "type": "array",
575         "items": {
576             "type": "object",
577             "properties": {
578                 "language": {
579                     "$ref": "http://openconnectivityfoundation.github.io/core/schemas/oic.types-
580 schema.json#/definitions/language-tag",
581                     "readOnly": true,
582                     "description": "An RFC 5646 language tag."
583                 },
584                 "value": {
585                     "type": "string",
586                     "description": "Pre-configured Device name in the indicated language.",
587                     "pattern": "^.*$",
588                     "readOnly": true
589                 }
590             }
591         },
592         "minItems": 1,
593         "readOnly": true,
594         "description": "Localized device name."
595     }
596 ]
597 },
598 },
599 "type": "object",
600 "required": ["dn"]
601 }
602 }
603 }
604

```

605 A.2.5 Property definition

606 Table A.2 defines the Properties that are part of the "oic.r.devconf" Resource Type.

607 **Table A.2 – The Property definitions of the Resource with type "rt" = "oic.r.devconf".**

Property name	Value type	Mandatory	Access mode	Description
id	multiple types: see schema	No	Read Write	
n	multiple types: see schema	No	Read Write	
dn	multiple types: see schema	Yes	Read Write	
rt	array: see schema	No	Read Only	Resource Type of the Resource.
if	array: see schema	No	Read Only	The OCF Interfaces supported by this Resource.

608 A.2.6 CRUDN behaviour

609 Table A.3 defines the CRUDN operations that are supported on the "oic.r.devconf" Resource Type.

610 **Table A.3 – The CRUDN operations of the Resource with type "rt" = "oic.r.devconf".**

Create	Read	Update	Delete	Notify
	get			observe

611 **A.3 Easy Setup Collection**

612 **A.3.1 Introduction**

613 The Easy Setup Resource stores useful information including the current status of unboxing a
614 Device and the last error code which are produced in the process of easy setup.
615 Note that the Easy Setup Resource is a Collection Resource, which contains Links to WiFiConf,
616 and DevConf Resources and may additionally contain Links to other Resources.
617

618 **A.3.2 Example URI**

619 /EasySetupResURI

620 **A.3.3 Resource type**

621 The Resource Type is defined as: "oic.r.easyssetup, oic.wk.col".

622 **A.3.4 OpenAPI 2.0 definition**

```
623 {
624   "swagger": "2.0",
625   "info": {
626     "title": "Easy Setup Collection",
627     "version": "20190327",
628     "license": {
629       "name": "OCF Data Model License",
630       "url":
631 "https://github.com/openconnectivityfoundation/core/blob/e28a9e0a92e17042ba3e83661e4c0fbce8bdc4ba/LI
632 CENSE.md",
633       "x-copyright": "Copyright 2016-2019 Open Connectivity Foundation, Inc. All rights reserved."
634     },
635     "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
636   },
637   "schemes": ["http"],
638   "consumes": ["application/json"],
639   "produces": ["application/json"],
640   "paths": {
641     "/EasySetupResURI?if=oic.if.ll" : {
642       "get": {
643         "description": "The Easy Setup Resource stores useful information including the current
644 status of unboxing a Device and the last error code which are produced in the process of easy
645 setup.\nNote that the Easy Setup Resource is a Collection Resource, which contains Links to
646 WiFiConf, and DevConf Resources and may additionally contain Links to other Resources.\n",
647         "parameters": [
648           {"$ref": "#/parameters/interface-all"}
649         ],
650         "responses": {
651           "200": {
652             "description": "",
653             "x-example":
654             [
655               {
656                 "href": "/EasySetupResURI",
657                 "rt": ["oic.r.easyssetup", "oic.wk.col"],
658                 "if": ["oic.if.b"],
659                 "p":{"bm":3},
660                 "eps": [
661                   {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2}
662                 ],
663                 "rel":["self", "item"]
664               },
665               {
666                 "href": "/WiFiConfResURI",
667                 "rt": ["oic.r.wificonf"],
668                 "if": ["oic.if.baseline"],
669                 "p":{"bm":3},
670                 "eps": [
```

```

671         {"ep": "coaps://[fe80::bld6]:1111", "pri": 2}
672     ]
673 },
674 {
675     "href": "/DevConfResURI",
676     "rt": ["oic.r.devconf"],
677     "if": ["oic.if.baseline"],
678     "p":{"bm":3},
679     "eps": [
680         {"ep": "coaps://[fe80::bld6]:1111", "pri": 2}
681     ]
682 }
683 ],
684 "schema": { "$ref": "#/definitions/slinks" }
685 }
686 }
687 }
688 },
689 "/EasySetupResURI?if=oic.if.b" : {
690     "get": {
691         "description": "The Easy Setup Resource stores useful information including the current
692 status of unboxing a Device and the last error code which are produced in the process of easy
693 setup.\nNote that the Easy Setup Resource is a Collection Resource, which contains Links to
694 Wi-FiConf, and DevConf Resources and may additionally contain Links to other Resources.\n",
695         "parameters": [
696             {"$ref": "#/parameters/interface-all"}
697         ],
698         "responses": {
699             "200": {
700                 "description" : "",
701                 "x-example":
702                 [
703                     {
704                         "href": "/EasySetupResURI",
705                         "rep":{
706                             "ps" : 0,
707                             "lec": 0,
708                             "cn": [1]
709                         }
710                     },
711                     {
712                         "href": "/Wi-FiConfResURI",
713                         "rep":{
714                             "swmt" : ["A", "B", "G"],
715                             "swf": ["2.4G", "5G"],
716                             "tnn": "Home_AP_SSID",
717                             "cd": "Home_AP_PWD",
718                             "wat": "WPA2_PSK",
719                             "wet": "AES",
720                             "swat": ["WPA_PSK", "WPA2_PSK"],
721                             "swet": ["TKIP", "AES", "TKIP_AES"]
722                         }
723                     },
724                     {
725                         "href": "/DevConfResURI",
726                         "rep":{
727                             "dn" : "My Refrigerator"
728                         }
729                     }
730                 ],
731                 "schema": { "$ref": "#/definitions/sbatch" }
732             }
733         }
734     },
735     "post": {
736         "description": "Able to deliver Wi-Fi, Device configuration and other
737 configuration\ninformation in a batch by utilizing 'batch' OCF Interface.\nIf you want to deliver
738 Wi-Fi and Device configuration information in a batch,\nyou can write all Properties you want to
739 send with a 'batch' OCF Interface.\nThe below example is the case to send Easy Setup and Wi-Fi
740 configuration\n(i.e. connection type, target network, auth type information) in a batch.\n",

```

```

741     "parameters": [
742         { "$ref": "#/parameters/interface-update" },
743     ],
744     "name": "body",
745     "in": "body",
746     "required": true,
747     "schema": { "$ref": "#/definitions/sbatch-update" },
748     "x-example":
749     [
750         {
751             "href": "/EasySetupResURI",
752             "rep": {
753                 "cn": [1]
754             }
755         },
756         {
757             "href": "/WiFiConfResURI",
758             "rep": {
759                 "tnn": "Home_AP_SSID",
760                 "cd": "Home_AP_PWD",
761                 "wat": "WPA2_PSK",
762                 "wet": "AES"
763             }
764         }
765     ]
766 ],
767 "responses": {
768     "200": {
769         "description": "",
770         "x-example":
771         [
772             {
773                 "href": "/EasySetupResURI",
774                 "rep": {
775                     "ps": 0,
776                     "lec": 0,
777                     "cn": [1]
778                 }
779             },
780             {
781                 "href": "/WiFiConfResURI",
782                 "rep": {
783                     "swmt": ["A", "B", "G"],
784                     "swf": ["2.4G", "5G"],
785                     "tnn": "Home_AP_SSID",
786                     "cd": "Home_AP_PWD",
787                     "wat": "WPA2_PSK",
788                     "wet": "AES",
789                     "swat": ["WPA_PSK", "WPA2_PSK"],
790                     "swet": ["TKIP", "AES", "TKIP_AES"]
791                 }
792             },
793             {
794                 "href": "/DevConfResURI",
795                 "rep": {
796                     "dn": "My Refrigerator"
797                 }
798             }
799         ],
800     "schema": { "$ref": "#/definitions/sbatch" }
801 }
802 }
803 }
804 },
805 "/EasySetupResURI?if=oic.if.baseline" : {
806     "get": {
807         "description": "The Easy Setup Resource stores useful information including the current
808 status of unboxing a Device and the last error code which are produced in the process of easy
809 setup.\nNote that the Easy Setup Resource is a Collection Resource, which contains Links to
810

```

```

811 WiFiConf, and DevConf Resources and may additionally contain Links to other Resources.\n",
812 "parameters": [
813   {"$ref": "#/parameters/interface-all"}
814 ],
815 "responses": {
816   "200": {
817     "description": "",
818     "x-example":
819     {
820       "rt": ["oic.r.easyssetup", "oic.wk.col"],
821       "if": ["oic.if.ll", "oic.if.baseline", "oic.if.b"],
822       "ps": 0,
823       "lec": 0,
824       "cn": [1],
825       "links": [
826         {
827           "href": "/EasySetupResURI",
828           "rt": ["oic.r.easyssetup", "oic.wk.col"],
829           "if": ["oic.if.b"],
830           "p":{"bm":3},
831           "eps": [
832             {"ep": "coaps://[fe80:b1d6]:1111", "pri": 2}
833           ],
834           "rel":["self", "item"]
835         },
836         {
837           "href": "/WiFiConfResURI",
838           "rt": ["oic.r.wificonf"],
839           "if": ["oic.if.baseline"],
840           "p":{"bm":3},
841           "eps": [
842             {"ep": "coaps://[fe80:b1d6]:1111", "pri": 2}
843           ]
844         },
845         {
846           "href": "/DevConfResURI",
847           "rt": ["oic.r.devconf"],
848           "if": ["oic.if.baseline"],
849           "p":{"bm":3},
850           "eps": [
851             {"ep": "coaps://[fe80:b1d6]:1111", "pri": 2}
852           ]
853         }
854       ]
855     },
856     "schema": { "$ref": "#/definitions/EasySetup" }
857   }
858 },
859 "post": {
860   "description": "Able to update connection type to attempt to connect to the Enroller to
861 start during while posting to /EasySetupResURI\nThe below example is the case to send Easy Setup
862 configuration\n(i.e. connection type) in a post.\n",
863   "parameters": [
864     {"$ref": "#/parameters/interface-update"},
865     {
866       "name": "body",
867       "in": "body",
868       "required": true,
869       "schema": { "$ref": "#/definitions/EasySetupUpdate" },
870       "x-example":
871       {
872         "cn": [1]
873       }
874     }
875   ],
876   "responses": {
877     "200": {
878       "description": "",
879       "x-example":

```

```

881     {
882       "rt" : ["oic.r.easyssetup", "oic.wk.col"],
883       "if" : ["oic.if.ll", "oic.if.baseline", "oic.if.b"],
884       "ps" : 0,
885       "lec" : 0,
886       "cn" : [1],
887       "links" : [
888         {
889           "href" : "/EasySetupResURI",
890           "rt" : ["oic.r.easyssetup", "oic.wk.col"],
891           "if" : ["oic.if.b", "oic.if.ll", "oic.if.baseline"],
892           "p":{"bm":3},
893           "eps" : [
894             {"ep" : "coaps://[fe80::b1d6]:1111", "pri" : 2}
895           ],
896           "rel":["self", "item"]
897         },
898         {
899           "href" : "/WiFiConfResURI",
900           "rt" : ["oic.r.wificonf"],
901           "if" : ["oic.if.rw", "oic.if.baseline"],
902           "p":{"bm":3},
903           "eps" : [
904             {"ep" : "coaps://[fe80::b1d6]:1111", "pri" : 2}
905           ]
906         },
907         {
908           "href" : "/DevConfResURI",
909           "rt" : ["oic.r.devconf"],
910           "if" : ["oic.if.r", "oic.if.baseline"],
911           "p":{"bm":3},
912           "eps" : [
913             {"ep" : "coaps://[fe80::b1d6]:1111", "pri" : 2}
914           ]
915         }
916       ]
917     },
918     "schema": { "$ref": "#/definitions/EasySetup" }
919   }
920 }
921 }
922 }
923 },
924 "parameters": {
925   "interface-all" : {
926     "in" : "query",
927     "name" : "if",
928     "type" : "string",
929     "enum" : ["oic.if.ll", "oic.if.b", "oic.if.baseline"]
930   },
931   "interface-update" : {
932     "in" : "query",
933     "name" : "if",
934     "type" : "string",
935     "enum" : ["oic.if.b", "oic.if.baseline"]
936   }
937 },
938 "definitions": {
939   "oic.oic-link": {
940     "type": "object",
941     "properties": {
942       "anchor": {
943         "$ref":
944 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
945 schema.json#/definitions/anchor"
946       },
947       "di": {
948         "$ref":
949 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
950 schema.json#/definitions/di"

```

```

951     },
952     "eps": {
953       "$ref":
954       "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
955       schema.json#/definitions/eps"
956     },
957     "href": {
958       "$ref":
959       "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
960       schema.json#/definitions/href"
961     },
962     "ins": {
963       "$ref":
964       "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
965       schema.json#/definitions/ins"
966     },
967     "p": {
968       "$ref":
969       "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
970       schema.json#/definitions/p"
971     },
972     "rel": {
973       "$ref":
974       "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
975       schema.json#/definitions/rel_array"
976     },
977     "title": {
978       "$ref":
979       "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
980       schema.json#/definitions/title"
981     },
982     "type": {
983       "$ref":
984       "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
985       schema.json#/definitions/type"
986     },
987     "if": {
988       "description": "The OCF Interfaces supported by the target Resource",
989       "items": {
990         "enum": [
991           "oic.if.baseline",
992           "oic.if.ll",
993           "oic.if.b",
994           "oic.if.r",
995           "oic.if.rw"
996         ],
997         "type": "string",
998         "maxLength": 64
999       },
1000     "minItems": 1,
1001     "uniqueItems": true,
1002     "type": "array"
1003   },
1004   "rt": {
1005     "description": "Resource Type of the target Resource",
1006     "items": {
1007       "maxLength": 64,
1008       "type": "string"
1009     },
1010     "minItems": 1,
1011     "uniqueItems": true,
1012     "type": "array"
1013   }
1014 },
1015 "required": [
1016   "href",
1017   "rt",
1018   "if"
1019 ]
1020 },

```

```

1021     "slinks" : {
1022         "type": "array",
1023         "items": {
1024             "$ref": "#/definitions/oic.oic-link"
1025         }
1026     },
1027     "sbatch" : {
1028         "minItems" : 1,
1029         "items" : {
1030             "additionalProperties": true,
1031             "properties": {
1032                 "href": {
1033                     "$ref":
1034 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
1035 schema.json#/definitions/href"
1036                 },
1037                 "rep": {
1038                     "description": "The response payload from a single Resource",
1039                     "type": "object",
1040                     "anyOf": [
1041                         {
1042                             "$ref": "#/definitions/EasySetup"
1043                         },
1044                         {
1045                             "$ref": "https://openconnectivityfoundation.github.io/core-
1046 extensions/swagger2.0/oic.r.wificonf.swagger.json#/definitions/WiFiConf"
1047                         },
1048                         {
1049                             "$ref": "https://openconnectivityfoundation.github.io/core-
1050 extensions/swagger2.0/oic.r.devconf.swagger.json#/definitions/DevConf"
1051                         }
1052                     ]
1053                 }
1054             },
1055             "required": [
1056                 "href",
1057                 "rep"
1058             ],
1059             "type": "object"
1060         },
1061         "type" : "array"
1062     },
1063     "sbatch-update" : {
1064         "minItems" : 1,
1065         "items" : {
1066             "additionalProperties": true,
1067             "description": "Array of Resource representations to apply to the batch Collection, using
1068 href to indicate which resource(s) in the batch to update. If the href Property is empty,
1069 effectively making the URI reference to the Collection itself, the representation is to be applied
1070 to all Resources in the batch",
1071             "properties": {
1072                 "href": {
1073                     "$ref":
1074 "https://openconnectivityfoundation.github.io/core/schemas/oic.links.properties.core-
1075 schema.json#/definitions/href"
1076                 },
1077                 "rep": {
1078                     "description": "The response payload from a single Resource",
1079                     "type": "object",
1080                     "anyOf": [
1081                         {
1082                             "$ref": "#/definitions/EasySetupUpdate"
1083                         },
1084                         {
1085                             "$ref": "https://openconnectivityfoundation.github.io/core-
1086 extensions/swagger2.0/oic.r.wificonf.swagger.json#/definitions/WiFiConfUpdate"
1087                         }
1088                     ]
1089                 }
1090             },

```

```

1091     "required": [
1092         "href",
1093         "rep"
1094     ],
1095     "type": "object"
1096 },
1097 "type": "array"
1098 },
1099 "EasySetup" : {
1100     "properties": {
1101         "n" : {
1102             "$ref":
1103 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
1104 schema.json#/definitions/n"
1105         },
1106         "rts" : {
1107             "description": "Resource Type of the Resources within the Collection",
1108             "items": {
1109                 "maxLength": 64,
1110                 "type": "string"
1111             },
1112             "minItems": 1,
1113             "uniqueItems": true,
1114             "readOnly": true,
1115             "type": "array"
1116         },
1117         "id" : {
1118             "$ref":
1119 "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
1120 schema.json#/definitions/id"
1121         },
1122         "rts-m" : {
1123             "description": "Resource Type of the mandatory Resources within the Collection",
1124             "items": {
1125                 "maxLength": 64,
1126                 "type": "string"
1127             },
1128             "minItems": 1,
1129             "uniqueItems": true,
1130             "readOnly": true,
1131             "type": "array"
1132         },
1133         "if" : {
1134             "description": "The OCF Interfaces supported by this Resource",
1135             "items": {
1136                 "enum": [
1137                     "oic.if.ll",
1138                     "oic.if.baseline",
1139                     "oic.if.b"
1140                 ],
1141                 "type": "string",
1142                 "maxLength": 64
1143             },
1144             "minItems": 2,
1145             "uniqueItems": true,
1146             "readOnly": true,
1147             "type": "array"
1148         },
1149         "rt" : {
1150             "items": {
1151                 "enum": [
1152                     "oic.r.easyssetup",
1153                     "oic.wk.col"
1154                 ],
1155                 "type": "string",
1156                 "maxLength": 64
1157             },
1158             "minItems": 2,
1159             "type": "array",
1160             "uniqueItems": true

```



```

1161     },
1162     "ps" : {
1163         "description": "Indicates the easy setup status of the Device. (0: Need to Setup, 1:
1164 Connecting to Enroller, 2: Connected to Enroller, 3: Failed to Connect to Enroller, 4~254: Reserved,
1165 255: EOF)",
1166         "enum": [
1167             0,
1168             1,
1169             2,
1170             3
1171         ],
1172         "readOnly": true,
1173         "type": "integer"
1174     },
1175     "lec" : {
1176         "description": "Indicates a failure reason (0: NO error, 1: A given SSID is not found, 2:
1177 Wi-Fi's password is wrong, 3: IP address is not allocated, 4: No internet connection, 5: Timeout, 6:
1178 Wi-Fi Auth Type is not supported by the Enrollee, 7: Wi-Fi Encryption Type is not supported by the
1179 Enrollee, 8: Wi-Fi Auth Type is wrong (failure while connecting to the Enroller), 9: Wi-Fi
1180 Encryption Type is wrong (failure while connecting to the Enroller), 10~254: Reserved, 255: Unknown
1181 error)",
1182         "enum": [
1183             0,
1184             1,
1185             2,
1186             3,
1187             4,
1188             5,
1189             6,
1190             7,
1191             8,
1192             9,
1193             255
1194         ],
1195         "readOnly": true,
1196         "type": "integer"
1197     },
1198     "cn" : {
1199         "description": "Indicates an array of connection types that trigger an attempt to connect
1200 to the Enroller to start.",
1201         "items": {
1202             "description": "Connection type to attempt. (1 : Wi-Fi, 2 : other entities / transports
1203 to be added in future (e.g. Connect to cloud / BLE))",
1204             "type": "integer"
1205         },
1206         "type": "array"
1207     },
1208     "links" : {
1209         "type": "array",
1210         "description": "A set of OCF Links.",
1211         "items": {
1212             "$ref": "#/definitions/oic.oic-link"
1213         }
1214     },
1215 },
1216 "type" : "object",
1217 "required": ["ps","lec","cn"]
1218 },
1219 "EasySetupUpdate" : {
1220     "additionalProperties": true,
1221     "description": "Update to writeable values in EasySetupResURI",
1222     "properties": {
1223         "cn" : {
1224             "description": "Indicates an array of connection types that trigger an attempt to connect
1225 to the Enroller to start.",
1226             "items": {
1227                 "description": "Connection type to attempt. (1 : Wi-Fi, 2 : other entities / transports
1228 to be added in future (e.g. Connect to cloud / BLE))",
1229                 "type": "integer"
1230             },

```

```

1231         "type": "array"
1232     },
1233 },
1234     "required": [
1235         "cn"
1236     ],
1237     "type": "object"
1238 }
1239 }
1240 }
1241

```

1242 A.3.5 Property definition

1243 Table A.4 defines the Properties that are part of the "oic.r.easyssetup, oic.wk.col" Resource Type.

1244 **Table A.4 – The Property definitions of the Resource with type "rt" = "oic.r.easyssetup,**
1245 **oic.wk.col".**

Property name	Value type	Mandatory	Access mode	Description
rep	object: see schema	Yes	Read Write	The response payload from a single Resource.
href	multiple types: see schema	Yes	Read Write	
rep	object: see schema	Yes	Read Write	The response payload from a single Resource.
href	multiple types: see schema	Yes	Read Write	
links	array: see schema	No	Read Write	A set of OCF Links.
rts-m	array: see schema	No	Read Only	Resource Type of the mandatory Resources within the Collection.
n	multiple types: see schema	No	Read Write	
if	array: see schema	No	Read Only	The OCF Interfaces supported by this Resource.
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).
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).
rt	array: see schema	No	Read Write	
rts	array: see schema	No	Read Only	Resource Type of the Resources within the Collection.
cn	array: see schema	Yes	Read Write	Indicates an array of connection types that trigger an attempt to connect to the Enroller to start.
id	multiple types: see schema	No	Read Write	
rt	array: see schema	Yes	Read Write	Resource Type of the target Resource.
href	multiple types: see schema	Yes	Read Write	
if	array: see schema	Yes	Read Write	The OCF Interfaces supported by the target Resource.
type	multiple types: see schema	No	Read Write	
p	multiple types: see schema	No	Read Write	
ins	multiple types: see schema	No	Read Write	
title	multiple types: see schema	No	Read Write	
anchor	multiple types: see schema	No	Read Write	
rel	multiple types: see schema	No	Read Write	
eps	multiple types: see schema	No	Read Write	
di	multiple types: see schema	No	Read Write	
cn	array: see schema	Yes	Read Write	Indicates an array of connection types that trigger an attempt to connect to the Enroller to start.

1246 **A.3.6 CRUDN behaviour**

1247 Table A.5 defines the CRUDN operations that are supported on the "oic.r.easyssetup, oic.wk.col"
1248 Resource Type.

1249 **Table A.5 – The CRUDN operations of the Resource with type "rt" = "oic.r.easyssetup,**
1250 **oic.wk.col".**

Create	Read	Update	Delete	Notify
	get	post		observe

1251 **A.4 Wi-Fi Configuration**

1252 **A.4.1 Introduction**

1253 WiFiConf Resource stores essential information to help an unboxing Device
1254 to connect to an existing Wi-Fi AP.
1255

1256 **A.4.2 Example URI**

1257 /WiFiConfResURI

1258 **A.4.3 Resource type**

1259 The Resource Type is defined as: "oic.r.wificonf".

1260 **A.4.4 OpenAPI 2.0 definition**

```

1261 {
1262   "swagger": "2.0",
1263   "info": {
1264     "title": "Wi-Fi Configuration",
1265     "version": "20190327",
1266     "license": {
1267       "name": "OCF Data Model License",
1268       "url":
1269 "https://github.com/openconnectivityfoundation/core/blob/e28a9e0a92e17042ba3e83661e4c0fbce8bdc4ba/LI
1270 CENSE.md",
1271       "x-copyright": "Copyright 2018-2019 Open Connectivity Foundation, Inc. All rights reserved."
1272     },
1273     "termsOfService": "https://openconnectivityfoundation.github.io/core/DISCLAIMER.md"
1274   },
1275   "schemes": ["http"],
1276   "consumes": ["application/json"],
1277   "produces": ["application/json"],
1278   "paths": {
1279     "/WiFiConfResURI?if=oic.if.rw" : {
1280       "get": {
1281         "description": "The WiFiConf Resource stores essential information to help an unboxing
1282 Device connect to an existing Wi-Fi AP.\n",
1283         "parameters": [
1284           {"$ref": "#/parameters/interface-all"}
1285         ],
1286         "responses": {
1287           "200": {
1288             "description": "",
1289             "x-example":
1290             {
1291               "tnn": "Home_AP_SSID",
1292               "swmt": ["A", "B", "G"],
1293               "swf": ["2.4G", "5G"],
1294               "cd": "Home_AP_PWD",
1295               "wat": "WPA2_PSK",
1296               "wet": "AES",
1297               "swat": ["WPA_PSK", "WPA2_PSK"],

```

```

1298         "swet": ["TKIP", "AES", "TKIP_AES"]
1299     },
1300     "schema": { "$ref": "#/definitions/WiFiConf" }
1301 }
1302 },
1303 },
1304 "post": {
1305     "description": "Deliver Wi-Fi AP's information for an unboxing Device to connect to it.\n",
1306     "parameters": [
1307         { "$ref": "#/parameters/interface-all" },
1308         {
1309             "name": "body",
1310             "in": "body",
1311             "required": true,
1312             "schema": { "$ref": "#/definitions/WiFiConfUpdate" },
1313             "x-example":
1314                 {
1315                     "tnn": "Home_AP_SSID",
1316                     "cd": "Home_AP_PWD",
1317                     "wat": "WPA2_PSK",
1318                     "wet": "AES"
1319                 }
1320         }
1321     ],
1322     "responses": {
1323         "200": {
1324             "description": "",
1325             "x-example":
1326                 {
1327                     "tnn": "Home_AP_SSID",
1328                     "swmt": ["A", "B", "G"],
1329                     "swf": ["2.4G", "5G"],
1330                     "cd": "Home_AP_PWD",
1331                     "wat": "WPA2_PSK",
1332                     "wet": "AES",
1333                     "swat": ["WPA_PSK", "WPA2_PSK"],
1334                     "swet": ["TKIP", "AES", "TKIP_AES"]
1335                 },
1336             "schema": { "$ref": "#/definitions/WiFiConf" }
1337         }
1338     }
1339 },
1340 },
1341 "/WiFiConfResURI?if=oic.if.baseline" : {
1342     "get": {
1343         "description": "WiFiConf Resource stores essential information to help an unboxing
1344 Device\nto connect to an existing Wi-Fi AP.\n",
1345         "parameters": [
1346             { "$ref": "#/parameters/interface-all" }
1347         ],
1348         "responses": {
1349             "200": {
1350                 "description": "",
1351                 "x-example":
1352                     {
1353                         "rt": ["oic.r.wificonf"],
1354                         "if": ["oic.if.rw", "oic.if.baseline"],
1355                         "swmt": ["A", "B", "G"],
1356                         "swf": ["2.4G", "5G"],
1357                         "tnn": "Home_AP_SSID",
1358                         "cd": "Home_AP_PWD",
1359                         "wat": "WPA2_PSK",
1360                         "wet": "TKIP",
1361                         "swat": ["WPA_PSK", "WPA2_PSK"],
1362                         "swet": ["TKIP", "AES", "TKIP_AES"]
1363                     },
1364                 "schema": { "$ref": "#/definitions/WiFiConf" }
1365             }
1366         }
1367     },

```

```

1368     "post": {
1369         "description": "Deliver Wi-Fi AP's information for an unboxing device to connect to it.\n",
1370         "parameters": [
1371             { "$ref": "#/parameters/interface-all"},
1372             {
1373                 "name": "body",
1374                 "in": "body",
1375                 "required": true,
1376                 "schema": { "$ref": "#/definitions/WiFiConfUpdate" },
1377                 "x-example":
1378                     {
1379                         "tnn": "Home_AP_SSID",
1380                         "cd": "Home_AP_PWD",
1381                         "wat": "WPA2_PSK",
1382                         "wet": "AES"
1383                     }
1384             }
1385         ],
1386         "responses": {
1387             "200": {
1388                 "description": "",
1389                 "x-example":
1390                     {
1391                         "rt": ["oic.r.wificonf"],
1392                         "if": ["oic.if.rw", "oic.if.baseline"],
1393                         "tnn": "Home_AP_SSID",
1394                         "swmt": ["A", "B", "G"],
1395                         "swf": ["2.4G", "5G"],
1396                         "cd": "Home_AP_PWD",
1397                         "wat": "WPA2_PSK",
1398                         "wet": "AES",
1399                         "swat": ["WPA_PSK", "WPA2_PSK"],
1400                         "swet": ["TKIP", "AES", "TKIP_AES"]
1401                     },
1402                 "schema": { "$ref": "#/definitions/WiFiConf" }
1403             }
1404         }
1405     }
1406 },
1407 "parameters": {
1408     "interface-all": {
1409         "in": "query",
1410         "name": "if",
1411         "type": "string",
1412         "enum": ["oic.if.rw", "oic.if.baseline"]
1413     }
1414 },
1415 "definitions": {
1416     "WiFiConf": {
1417         "properties": {
1418             "rt": {
1419                 "description": "Resource Type of the Resource",
1420                 "items": {
1421                     "enum": ["oic.r.wificonf"],
1422                     "type": "string",
1423                     "maxLength": 64
1424                 },
1425                 "minItems": 1,
1426                 "uniqueItems": true,
1427                 "readOnly": true,
1428                 "type": "array"
1429             },
1430             "tnn": {
1431                 "description": "Indicates Target Network Name (SSID of Wi-Fi AP)",
1432                 "pattern": "^.*$",
1433                 "type": "string"
1434             },
1435             "swmt": {
1436                 "description": "Indicates supported Wi-Fi mode types. It can be multiple",
1437

```

```

1438     "items": {
1439         "description": "Supported Wi-Fi Mode Type.",
1440         "enum": [
1441             "A",
1442             "B",
1443             "G",
1444             "N",
1445             "AC"
1446         ],
1447         "type": "string"
1448     },
1449     "readOnly": true,
1450     "type": "array"
1451 },
1452 "wat" : {
1453     "description": "Indicates Wi-Fi Auth Type",
1454     "enum": [
1455         "None",
1456         "WEP",
1457         "WPA_PSK",
1458         "WPA2_PSK"
1459     ],
1460     "type": "string"
1461 },
1462 "n" : {
1463     "$ref":
1464     "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
1465     schema.json#/definitions/n"
1466 },
1467 "swat" : {
1468     "description": "Indicates supported Wi-Fi Auth types. It can be multiple",
1469     "items": {
1470         "description": "Indicates Wi-Fi Auth Type",
1471         "enum": [
1472             "None",
1473             "WEP",
1474             "WPA_PSK",
1475             "WPA2_PSK"
1476         ],
1477         "type": "string"
1478     },
1479     "readOnly": true,
1480     "type": "array"
1481 },
1482 "swf" : {
1483     "description": "Indicates Supported Wi-Fi frequencies by the Enrollee. Can be multiple.
1484 Valid values are ('2.4G', '5G')",
1485     "items": {
1486         "pattern": "^(2\\.4|5)G$",
1487         "type": "string"
1488     },
1489     "readOnly": true,
1490     "type": "array"
1491 },
1492 "swet" : {
1493     "description": "Indicates supported Wi-Fi Encryption types. It can be multiple",
1494     "items": {
1495         "description": "Indicates Wi-Fi Encryption Type",
1496         "enum": [
1497             "None",
1498             "WEP_64",
1499             "WEP_128",
1500             "TKIP",
1501             "AES",
1502             "TKIP_AES"
1503         ],
1504         "type": "string"
1505     },
1506     "readOnly": true,
1507     "type": "array"

```

```

1508     },
1509     "wet" : {
1510         "description": "Indicates Wi-Fi Encryption Type",
1511         "enum": [
1512             "None",
1513             "WEP_64",
1514             "WEP_128",
1515             "TKIP",
1516             "AES",
1517             "TKIP_AES"
1518         ],
1519         "type": "string"
1520     },
1521     "cd" : {
1522         "description": "Indicates credential information of Wi-Fi AP",
1523         "pattern": "^.*$",
1524         "type": "string"
1525     },
1526     "id" : {
1527         "$ref":
1528         "https://openconnectivityfoundation.github.io/core/schemas/oic.common.properties.core-
1529         schema.json#/definitions/id"
1530     },
1531     "if" : {
1532         "description": "The OCF Interfaces supported by this Resource",
1533         "items": {
1534             "enum": [
1535                 "oic.if.rw",
1536                 "oic.if.baseline"
1537             ],
1538             "type": "string",
1539             "maxLength": 64
1540         },
1541         "minItems": 2,
1542         "uniqueItems": true,
1543         "readOnly": true,
1544         "type": "array"
1545     }
1546 },
1547 "type" : "object",
1548 "required":["swmt", "swf", "swat", "swet", "tnn", "wat", "wet"]
1549 },
1550 "WiFiConfUpdate" : {
1551     "properties": {
1552         "wat" : {
1553             "description": "Indicates Wi-Fi Auth Type",
1554             "enum": [
1555                 "None",
1556                 "WEP",
1557                 "WPA_PSK",
1558                 "WPA2_PSK"
1559             ]
1560         },
1561         "cd" : {
1562             "description": "Indicates credential information of Wi-Fi AP",
1563             "pattern": "^.*$",
1564             "type": "string"
1565         },
1566         "wet" : {
1567             "description": "Indicates Wi-Fi Encryption Type",
1568             "enum": [
1569                 "None",
1570                 "WEP_64",
1571                 "WEP_128",
1572                 "TKIP",
1573                 "AES",
1574                 "TKIP_AES"
1575             ]
1576         },
1577         "tnn" : {

```



```

1578         "description": "Indicates Target Network Name (SSID of Wi-Fi AP)",
1579         "pattern": "^.*$",
1580         "type": "string"
1581     }
1582 },
1583     "type" : "object",
1584     "required":["tnn", "wat", "wet"]
1585 }
1586 }
1587 }
1588

```

1589 **A.4.5 Property definition**

1590 Table A.6 defines the Properties that are part of the "oic.r.wificonf" Resource Type.

1591 **Table A.6 – The Property definitions of the Resource with type "rt" = "oic.r.wificonf".**

Property name	Value type	Mandatory	Access mode	Description
if	array: see schema	No	Read Only	The OCF Interfaces supported by this Resource.
cd	string	No	Read Write	Indicates credential information of Wi-Fi AP.
wat	string	Yes	Read Write	Indicates Wi-Fi Auth Type.
swat	array: see schema	Yes	Read Only	Indicates supported Wi-Fi Auth types. It can be multiple.
tnn	string	Yes	Read Write	Indicates Target Network Name (SSID of Wi-Fi AP).
wet	string	Yes	Read Write	Indicates Wi-Fi Encryption Type.
id	multiple types: see schema	No	Read Write	
rt	array: see schema	No	Read Only	Resource Type of the Resource.
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").
n	multiple types: see schema	No	Read Write	
swet	array: see schema	Yes	Read Only	Indicates supported Wi-Fi Encryption types. It can be multiple.
wat	multiple types: see schema	Yes	Read Write	Indicates Wi-Fi Auth Type.

cd	string	No	Read Write	Indicates credential information of Wi-Fi AP.
tnn	string	Yes	Read Write	Indicates Target Network Name (SSID of Wi-Fi AP).
wet	multiple types: see schema	Yes	Read Write	Indicates Wi-Fi Encryption Type.

1592 **A.4.6 CRUDN behaviour**

1593 Table A.7 defines the CRUDN operations that are supported on the "oic.r.wificonf" Resource Type.

1594 **Table A.7 – The CRUDN operations of the Resource with type "rt" = "oic.r.wificonf".**

Create	Read	Update	Delete	Notify
	get	post		observe

1595