

OCF Core Specification Extension WiFi Easy Setup

VERSION 2.0 | June 22, 2018



OPEN CONNECTIVITY
FOUNDATION®

CONTACT admin@openconnectivity.org
Copyright OCF © 2018. 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 © 2017-2018 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	1	Scope	7
23	2	Normative references	7
24	3	Terms, definitions, symbols and abbreviations	8
25	3.1	Terms and definitions	8
26	3.2	Conventions	8
27	3.3	Data types	8
28	4	Document conventions and organization	9
29	5	Overview	10
30	5.1	Introduction	10
31	5.2	Architecture	10
32	5.3	Example Scenario	10
33	6	Resource model	11
34	6.1	Introduction	11
35	6.2	EasySetup Resource	11
36	6.2.1	Overview	11
37	6.2.2	Resource	11
38	6.3	WiFiConf Resource Type	12
39	6.3.1	Introduction	12
40	6.3.2	Resource Type	12
41	6.4	DevConf Resource Type	13
42	6.4.1	Introduction	13
43	6.4.2	Resource Type	14
44	7	Network and connectivity	15
45	8	Functional interactions	16
46	8.1	Onboarding, Provisioning and Configuration	16
47	8.2	Resource discovery	16
48	8.3	Retrieving and Updating Easy Setup Resources	16
49	8.4	Error Handling	16
50	8.5	Example Easy Setup Flow	17
51	8.6	Easy Setup SSID Tags	19
52	8.7	Easy Setup Information Element	19
53	8.7.1	Overview	19
54	8.7.2	OCF Device Information Element (IE)	19
55	9	Security	22
56	Annex A (normative)	Resource Type definitions	23
57	A.1	List of Resource Type definitions	23
58	A.2	Easy Setup	23
59	A.2.1	Introduction	23
60	A.2.2	Example URI	23
61	A.2.3	Resource Type	23

62	A.2.4	RAML Definition.....	23
63	A.2.5	Property Definition.....	26
64	A.2.6	CRUDN behaviour.....	27
65	A.3	Wi-Fi Configuration Resource Baseline Interface.....	27
66	A.3.1	Introduction.....	27
67	A.3.2	Example URI.....	27
68	A.3.3	Resource Type.....	27
69	A.3.4	RAML Definition.....	27
70	A.3.5	Property Definition.....	31
71	A.3.6	CRUDN behaviour.....	31
72	A.4	Device Configuration.....	32
73	A.4.1	Introduction.....	32
74	A.4.2	Example URI.....	32
75	A.4.3	Resource Type.....	32
76	A.4.4	RAML Definition.....	32
77	A.4.5	Property Definition.....	33
78	A.4.6	CRUDN behaviour.....	34
79	Annex B (informative)	Swagger2.0 definitions.....	35
80	B.1	Device Configuration.....	35
81	B.1.1	Introduction.....	35
82	B.1.2	Example URI.....	35
83	B.1.3	Resource Type.....	35
84	B.1.4	Swagger2.0 Definition.....	35
85	B.1.5	Property Definition.....	37
86	B.1.6	CRUDN behaviour.....	37
87	B.2	Easy Setup.....	37
88	B.2.1	Introduction.....	37
89	B.2.2	Example URI.....	37
90	B.2.3	Resource Type.....	37
91	B.2.4	Swagger2.0 Definition.....	37
92	B.2.5	Property Definition.....	53
93	B.2.6	CRUDN behaviour.....	57
94	B.3	Wi-Fi Configuration.....	57
95	B.3.1	Introduction.....	57
96	B.3.2	Example URI.....	57
97	B.3.3	Resource Type.....	57
98	B.3.4	Swagger2.0 Definition.....	57
99	B.3.5	Property Definition.....	64
100	B.3.6	CRUDN behaviour.....	65
101			
102			

103
104
105
106
107
108
109

Figures

Figure 1. Easy Setup deployment architecture	10
--	----

Tables

110	
111	
112	Table 1. EasySetup Resource Type..... 11
113	Table 2. “oic.r.easyssetup” Resource Type definition..... 11
114	Table 3. WiFiConf Resource Type..... 13
115	Table 4. “oic.r.wificonf” Resource Type definition 13
116	Table 5. DevConf Resource Type 14
117	Table 6. “oic.r.devconf” Resource Type definition..... 14
118	Table 7 Easy Setup Information Element TLVs..... 20
119	Table 8. Alphabetized list of resources 23
120	Table 9 Easy Setup Collection Baseline Interface Property Definitions 26
121	Table 10 Easy Setup Collection Baseline Interface CRUDN operations 27
122	Table 11 Wi-Fi Configuration Resource Baseline Interface Property Definitions 31
123	Table 12 Wi-Fi Configuration Resource Baseline Interface CRUDN operations 31
124	Table 13 Device Configuration Property Definitions 33
125	Table 14 Device Configuration CRUDN operations 34
126	Table 15 The properties definitions of the resource..... 37
127	Table 16 The CRUDN operations of the resource..... 37
128	Table 17 The properties definitions of the resource..... 53
129	Table 18 The CRUDN operations of the resource..... 57
130	Table 19 The properties definitions of the resource..... 64
131	Table 20 The CRUDN operations of the resource..... 65
132	
133	

134 **1 Scope**

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

139 **2 Normative references**

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

143 OCF Core Specification, *Open Connectivity Foundation Core Specification*, Version 1.3.1

144 Available at: https://openconnectivity.org/specs/OCF_Core_Specification_v1.3.0.pdf

145 Latest version available at: https://openconnectivity.org/specs/OCF_Core_Specification.pdf

146 OCF Security Specification, *Open Connectivity Foundation Security Capabilities*, Version 1.3

147 Available at: https://openconnectivity.org/specs/OCF_Security_Specification_v1.3.0.pdf

148 Latest version available at https://openconnectivity.org/specs/OCF_Security_Specification.pdf

149 OCF Device Specification, *Open Connectivity Foundation Device Specification*, Version 1.3

150 Available at: https://openconnectivity.org/specs/OCF_Device_Specification_v1.3.0.pdf

151 Latest version available at: https://openconnectivity.org/specs/OCF_Device_Specification.pdf

152 IEEE 802.11:2016, IEEE Standard for Information technology—Telecommunications and
153 information exchange between systems Local and metropolitan area networks—Specific
154 requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY)
155 Specifications, December 2016

156 <https://standards.ieee.org/findstds/standard/802.11-2016.html>

157 IETF RFC 5646, *Tags for Identifying Languages*, September 2009

158 <https://www.rfc-editor.org/info/rfc5646>

159 IETF RFC 7159, *The JavaScript Object Notation (JSON) Data Interchange Format*, March 2014

160 <https://www.rfc-editor.org/info/rfc7159>

161 IETF RFC 7252, *The Constrained Application Protocol (CoAP)*, June 2014

162 <https://www.rfc-editor.org/info/rfc7252>

163 JSON Schema Validation, *JSON Schema: interactive and non-interactive validation*, January 2013

164 <http://json-schema.org/latest/json-schema-validation.html>

165 OpenAPI specification, *aka Swagger RESTful API Documentation Specification*, Version 2.0

166 <https://github.com/OAI/OpenAPI-Specification/blob/master/versions/2.0.md>

167

168 **3 Terms, definitions, symbols and abbreviations**

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

170 **3.1 Terms and definitions**

171 As defined in the OCF Core Specification with the following additions.

172 **3.1.1**

173 **Easy Setup Enrollment**

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

176 **3.1.2**

177 **Enrollee**

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

179 **3.1.3**

180 **Enroller**

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

182 **3.1.4**

183 **Mediator**

184 logical function that enables the Enrollee to connect to the target network (Enroller). The Mediator
185 transfers configuration information to the Enrollee. E.g. Mobile Phone

186 **3.1.5**

187 **Easy Setup**

188 Process of configuring an Enrollee using Mediator (by transferring of essential information to the
189 Enrollee).

190 **3.1.6**

191 **Soft AP**

192 Software Enabled Access Point hosted on the device which is not a dedicated Access Point.

193 **3.2 Conventions**

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

198 **3.3 Data types**

199 As defined in the OCF Core Specification.

200

201 **4 Document conventions and organization**

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

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

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

208 Recommended (or should)(S).

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

215 Allowed (may or allowed)(O).

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

219 DEPRECATED.

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

226 Conditionally allowed (CA)

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

229 Conditionally required (CR)

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

233

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

235 Words that are emphasized are printed in italic.

236

237 **5 Overview**

238 **5.1 Introduction**

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

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

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

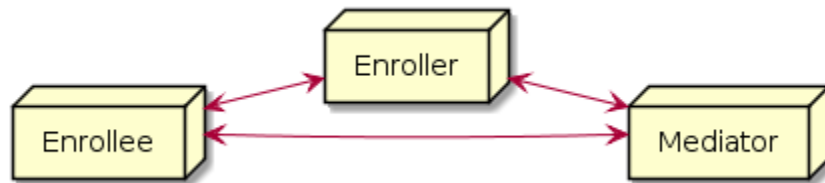
- 245 • Local network connection information, e.g. in case of Wi-Fi it will be Wi-Fi access point
246 information.
- 247 • Device Configuration: Additional Device configuration information.

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

249 **5.2 Architecture**

250 Figure 1 shows the deployment architectural approach.

251



252

253 **Figure 1. Easy Setup deployment architecture**

254 Easy Setup defines the following roles: Enrollee, Enroller, and Mediator. Please refer to Section
255 3.1 for definitions thereof.

256 **5.3 Example Scenario**

257 The following scenario presents a typical setup case.

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

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

266

267 **6 Resource model**

268 **6.1 Introduction**

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

- 270 1. EasySetup Resource Type
- 271 2. WiFiConf Resource Type
- 272 3. DevConf Resource Type

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

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



277
278 **Figure 2. Easy Setup Resource Types**

279 **6.2 EasySetup Resource**

280 **6.2.1 Overview**

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

283 **6.2.2 Resource**

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

285 **Table 1. EasySetup Resource Type**

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

286
287 Table 2. "oic.r.easysetup" Resource Type definition defines the details for the "oic.r.easysetup"
288 Resource Type.

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

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

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

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

- 291 • “ps” equal to 0.
- 292 • “lec” equal to 0.
- 293 • “cn” equal to an empty array.

294 6.3 WiFiConf Resource Type

295 6.3.1 Introduction

296 The WiFiConf Resource Type stores information to help an Enrollee to connect to an existing Wi-Fi AP.
297

298 6.3.2 Resource Type

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

300

Table 3. WiFiConf Resource Type

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

301

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

304

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

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

305

306 **6.4 DevConf Resource Type**

307 **6.4.1 Introduction**

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

309 **6.4.2 Resource Type**

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

311 **Table 5. DevConf Resource Type**

Example URI	Resource Type Title	Resource Type ID ("rt" value)	Interfaces	Description	Related Functional Interaction
/example/DevConfResURI	Dev Conf	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.	

312

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

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

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

316

317 **7 Network and connectivity**

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

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

321

322 **8 Functional interactions**

323 **8.1 Onboarding, Provisioning and Configuration**

324 The Mediator may be present as a standalone function or in conjunction with other functions or
325 services such as AMS as part of an OBT (Onboarding Tool); please refer to the OCF Security
326 Specification.

327 **8.2 Resource discovery**

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

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

332 **8.3 Retrieving and Updating Easy Setup Resources**

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

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

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

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

345 For details of Easy Setup Resources refer to Section 6.

346 **8.4 Error Handling**

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

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

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

363 **8.5 Example Easy Setup Flow**

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

**Easy Setup Flow
(Informative)**

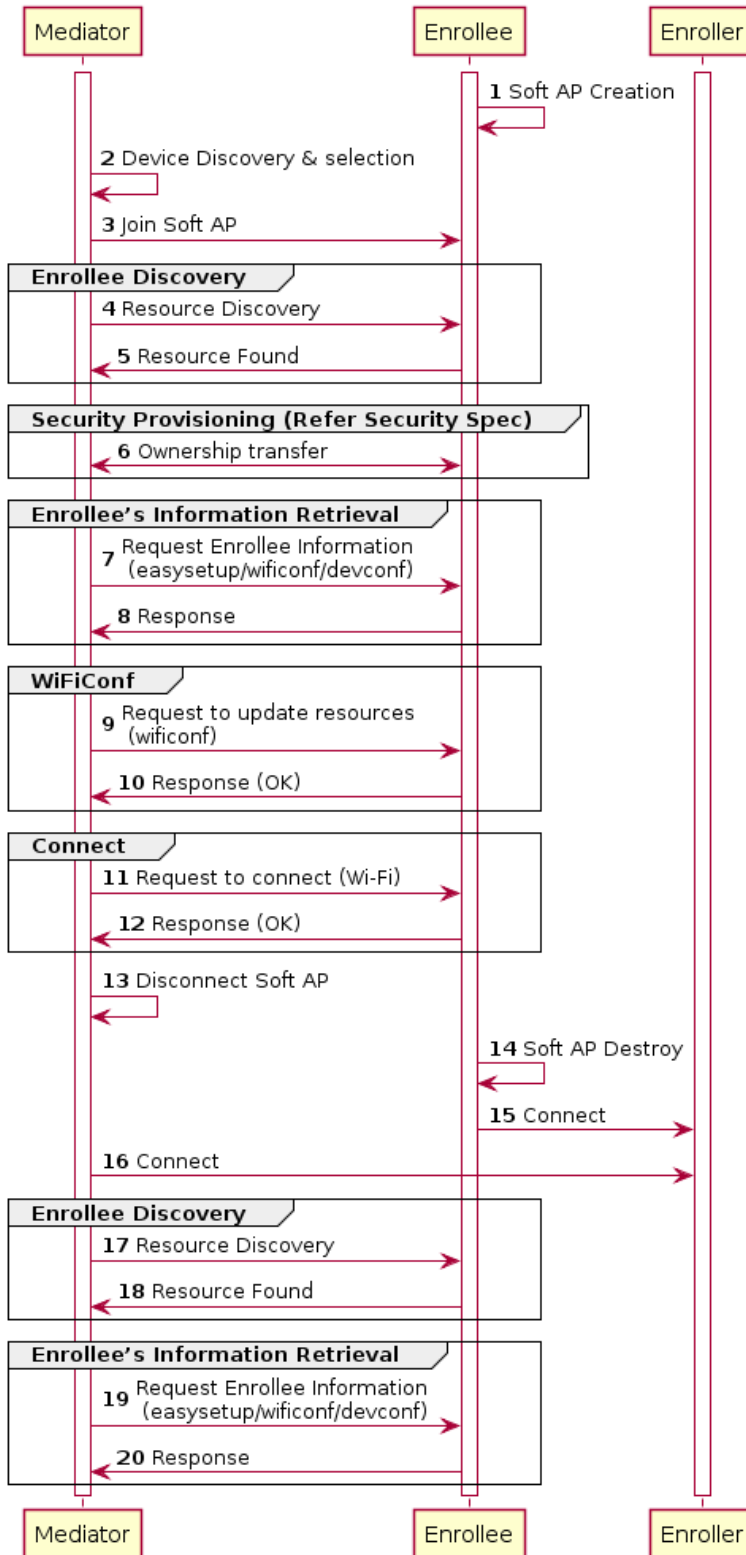


Figure 3. Easy Setup Flow (Informative)

367
368
369 The example flow above undergoes security provisioning (step 6) during Easy Setup.
370 Alternatively security provisioning can be done before Enrollee Discovery (steps 4 and 5) if
371 preferred. Please refer to the OCF Security Specification for more information on the different
372 scenarios.

373 **8.6 Easy Setup SSID Tags**

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

- 376 • “OCF_”
 - 377 ○ Prefix tag that has to be at the beginning of the SSID.
 - 378 ○ Example: OCF_MySSID
- 379 • “_OCF”
 - 380 ○ Suffix tag that has to be at the end of the SSID.
 - 381 ○ Example: MySSID_OCf

382 These tags are case sensitive.

383 **8.7 Easy Setup Information Element**

384 **8.7.1 Overview**

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

390 **8.7.2 OCF Device Information Element (IE)**

391 The Easy Setup Information Element has the following structure:

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

- 392
- 393 • Type is a unique id allocated by the IEEE registrar to identify different information elements
- 394 from each other. The Easy Setup Information Element shall have a Type value of 221 which is
- 395 standard vendor specific information element.
- 396 • Length shall indicate the total size of CID, OCF IE Type, and Data in bytes.
- 397 • Company ID (CID) is a unique 24-bit identifier for a specific company or organization. The Easy
- 398 Setup Information Element shall have a CID value of 6A 40 65.
- 399 • OCF IE Type is the identifier of the specific IE within OCF. The OCF IE Type shall be set to 0
- 400 for Easy Setup.
- 401 • Data is a set of type-length-value (TLV) structures that represent the device information in
- 402 Table 1. The length of this field shall be less than 252 bytes.

403
404 Each TLV has the following structure.

1 byte	1 byte	<250 bytes
Type	Length	Value

- 405
- 406 • Type shall indicate the type of the field from Table 7.
- 407 • Length shall indicate the length of the Value in bytes.
- 408 • Value shall represent the corresponding information for specific TLV type from Table 7.
- 409 Data is a set of TLVs as defined below:

410 **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 OCF Core Specification	1	Y
101	<65	Device Type/Class	Device Type as string	>=0	N

411 The TLVs may be set in any order inside an IE or IEC. All strings shall be UTF-8 encoded and shall not

412 include a null terminator. All TLVs in Table 7 with a 'Required' value of 'Y' shall be included in the IE or

413 IEC (if multiple IEs are required). The value of each TLV shall meet the length requirements specified in

414 Table 1.

415

416

417 **8.7.2.1 Device Friendly Name (Type 1)**

418 User readable string representing the friendly name of the device that is beaconing and ready to

419 undergo Easy Setup. This should match 'n' from oic.wk.d as defined in the OCF Core Specification.

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

421 **8.7.2.2 Device Type (Type 2)**

422 Device type shall be the shortened form of 'Device Type' as specified in the OCF Device Specification.

423 For example:

- 424
- 425 • 'Device Type' as specified in the OCF Device Specification: "oic.d.airconditioner"
- 426 • 'Device Type' as specified in a type 2 TLV: "airconditioner"

427

428 In cases where the device supports multiple functions, several type 2 TLVs may be included to represent

429 each function of the device.

430

431 If the device does not support any of the functions as specified in the OCF Device Specification, at least

432 one type 101 TLV shall be included. Type 101 TLV contains a user readable string in the same language

433 specified in the type 4 TLV. (Ex: "Lock").

434

435 If the device supports more than one function, a mix of type 2 and type 101 TLVs may be used depending

436 on which functions are defined in the OCF Device Specification.

437

438 **8.7.2.3 Device Manufacturer Name (Type 3)**

439 User readable string representing the manufacturer name of the device that is beaconing and ready
440 to undergo Easy Setup. This should match 'mnmn' from oic.wk.p as defined in the OCF Core
441 Specification.

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

443 **8.7.2.4 Language Tag (Type 4)**

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

448 Please see section 8.7.2.8 for information on supporting multiple languages.

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

451 **8.7.2.5 Protocol Independent ID (Type 5)**

452 This shall match 'piid' from oic.wk.d as defined in the OCF Core Specification.

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

454 **8.7.2.6 Multiple Information Elements**

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

- 456 • The total size of the TLVs is larger than the size of 'Data' as defined in an Easy Setup
457 Information Element.
- 458 • Support for multiple languages is necessary.

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

461 **8.7.2.7 IEC for Large TLV Size Support**

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

- 465 • Type
- 466 • Length
- 467 • CID
- 468 • OCF IE Type

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

471 **8.7.2.8 IEC for Multiple Language Support**

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

476

477

478 **9 Security**

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

Annex A (normative)

Resource Type definitions

A.1 List of Resource Type definitions

Table 8 contains the list of defined resources in this specification.

Table 8. Alphabetized list of resources

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

A.2 Easy Setup

A.2.1 Introduction

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

A.2.2 Example URI

/EasySetupResURI

A.2.3 Resource Type

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

A.2.4 RAML Definition

```
##RAML 0.8
title: Easy Setup Resource
version: v0.0.3-20170611

traits:
- interface-ll :
  queryParameters:
    if:
      enum: ["oic.if.ll"]
- interface-baseline :
  queryParameters:
    if:
      enum: ["oic.if.baseline"]
- interface-all :
  queryParameters:
    if:
      enum: ["oic.if.baseline", "oic.if.ll", "oic.if.b"]
```

```

515 - interface-batch :
516   queryParameters:
517     if:
518       enum: ["oic.if.b"]
519
520 /EasySetupResURI?if=oic.if.baseline:
521   description: |
522     Easy Setup resource stores useful information including current status of
523     unboxing device and last error code which are produced in a process of
524     easy setup.
525     Note that, Easy Setup resource is a type of collection resource, which
526     contains links to WiFiConf, DevConf resources and may additionally contain
527     links to other resources.
528
529   is : ['interface-baseline']
530   get:
531     description: |
532       Retrieve useful information during easy setup process :
533       1
534       A current status in easy setup process.
535       2
536       A last error code describing reason for failure occurred at the last
537       time.
538
539   responses :
540     200:
541       body:
542         application/json:
543           schema: /
544             {
545               "$schema": "http://json-schema.org/draft-04/schema#",
546               "description": "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
547 reserved.",
548               "id": "http://openconnectivityfoundation.github.io/core-
549 extensions/schemas/oic.r.easyssetup-schema.json#",
550               "definitions": {
551                 "oic.r.easyssetup": {
552                   "type": "object",
553                   "properties": {
554                     "rt": {
555                       "type": "array",
556                       "minItems": 2,
557                       "maxItems": 2,
558                       "uniqueItems": true,
559                       "items": {
560                         "enum": ["oic.r.easyssetup", "oic.wk.col"]
561                       }
562                     },
563                     "ps": {
564                       "type": "integer",
565                       "enum": [0, 1, 2, 3],
566                       "description": "Indicates the easy setup status of the device. (0: Need to
567 Setup, 1: Connecting to Enroller, 2: Connected to Enroller, 3: Failed to Connect to Enroller,
568 4~254: Reserved, 255: EOF)",
569                       "readOnly": true
570                     },
571                     "lec": {
572                       "type": "integer",
573                       "enum": [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 255],
574                       "description": "Indicates a failure reason (0: NO error, 1: A given SSID is
575 not found, 2: Wi-Fi's password is wrong, 3: IP address is not allocated, 4: No internet connection,
576 5: Timeout, 6: Wi-Fi Auth Type is not supported by the Enrollee, 7: Wi-Fi Encryption Type is not
577 supported by the Enrollee, 8: Wi-Fi Auth Type is wrong (failure while connecting to the Enroller),
578 9: Wi-Fi Encryption Type is wrong (failure while connecting to the Enroller), 10~254: Reserved,

```



```

579 255: Unknown error)",
580         "readOnly": true
581     },
582     "cn": {
583         "type": "array",
584         "description": "Indicates an array of connection types that trigger an
585 attempt to connect to the Enroller to start.",
586         "items": {
587             "type": "integer",
588             "description": "Connection type to attempt. (1 : Wi-Fi, 2 : other
589 entities / transports to be added in future (e.g. Connect to cloud / BLE))"
590         }
591     }
592 },
593 "required": ["ps", "lec", "cn"]
594 }
595 },
596 "type": "object",
597 "allof": [
598     { "$ref": "http://openconnectivityfoundation.github.io/core/schemas/oic.core-
599 schema.json#/definitions/oic.core"},
600     { "$ref":
601 "http://openconnectivityfoundation.github.io/core/schemas/oic.collection-
602 schema.json#/definitions/oic.collection.properties"},
603     { "$ref":
604 "http://openconnectivityfoundation.github.io/core/schemas/oic.collection-
605 schema.json#/definitions/oic.collection.links.arrayoflinks"},
606     { "$ref": "#/definitions/oic.r.easyssetup" }
607 ]
608 }
609
610 example: /
611 {
612     "rt" : ["oic.r.easyssetup", "oic.wk.col"],
613     "if" : ["oic.if.ll", "oic.if.baseline", "oic.if.b"],
614     "ps" : 0,
615     "lec": 0,
616     "cn": [1],
617     "links": [
618         {
619             "href": "/EasySetupResURI",
620             "rt": ["oic.r.easyssetup", "oic.wk.col"],
621             "if": ["oic.if.b"],
622             "p":{"bm":3},
623             "eps": [
624                 {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2}
625             ],
626             "rel":["self", "item"]
627         },
628         {
629             "href": "/WiFiConfResURI",
630             "rt": ["oic.r.wificonf"],
631             "if": ["oic.if.baseline"],
632             "p":{"bm":3},
633             "eps": [
634                 {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2}
635             ]
636         },
637         {
638             "href": "/DevConfResURI",
639             "rt": ["oic.r.devconf"],
640             "if": ["oic.if.baseline"],
641             "p":{"bm":3},
642             "eps": [
643                 {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2}
644             ]
645         }
646     ]
647 }
648

```

Table 9 Easy Setup Collection Baseline Interface Property Definitions

Property name	Value type	Mandatory	Access mode	Description
rt	array: see schema			
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)
cn	array: see schema	yes		Indicates an array of connection types that trigger an

				attempt to connect to the Enroller to start.
--	--	--	--	--

651 **A.2.6 CRUDN behaviour**

652 **Table 10 Easy Setup Collection Baseline Interface CRUDN operations**

Resource	Create	Read	Update	Delete	Notify
/EasySetupResURI		get			

653 **A.3 Wi-Fi Configuration Resource Baseline Interface**

654 **A.3.1 Introduction**

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

657 **A.3.2 Example URI**

658 /WiFiConfResURI

659 **A.3.3 Resource Type**

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

661 **A.3.4 RAML Definition**

```
662 #%RAML 0.8
663 title: Wi-Fi Configuration Resource
664 version: v0.0.3-20170611
665 traits:
666   - interface-rw :
667     queryParameters:
668       if:
669         enum: ["oic.if.rw"]
670   - interface-baseline :
671     queryParameters:
672       if:
673         enum: ["oic.if.baseline"]
674   - interface-all :
675     queryParameters:
676       if:
677         enum: ["oic.if.baseline", "oic.if.rw"]
678
679 /WiFiConfResURI?if=oic.if.baseline:
680   description: |
681     WiFiConf resource stores essential information to help an unboxing device
682     to connect to an existing Wi-Fi AP.
683
684   is : ['interface-baseline']
685   get:
686     description: |
687       Retrieve properties of WiFiConf resource.
688       The information includes :
689       1
690       Wi-Fi SSID and password
691       2
692       Wi-Fi Security type (i.e
693       auth type and encryption type)
694       3
695       Wi-Fi hardware capability (i.e
696       supported frequencies, modes,
```

```

697         auth types and encryption types)
698
699     responses :
700         200:
701             body:
702                 application/json:
703                 schema: /
704                 {
705                     "$schema": "http://json-schema.org/draft-04/schema#",
706                     "description": "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
reserved.",
707                     "id": "http://openconnectivityfoundation.github.io/core-
708 extensions/schemas/oic.r.wificonf-schema.json#",
709                     "definitions": {
710                         "oic.r.wificonf": {
711                             "type": "object",
712                             "properties": {
713                                 "swmt": {
714                                     "type": "array",
715                                     "description": "Indicates supported Wi-Fi mode types. It can be multiple",
716                                     "readOnly": true,
717                                     "items":
718                                     {
719                                         "type": "string",
720                                         "enum": ["A", "B", "G", "N", "AC"],
721                                         "description": "Supported Wi-Fi Mode Type."
722                                     }
723                                 },
724                                 "swf": {
725                                     "type": "array",
726                                     "description": "Indicates Supported Wi-Fi frequencies by the Enrollee. Can
727 be multiple. Valid values are ('2.4G', '5G')",
728                                     "readOnly": true,
729                                     "items":
730                                     {
731                                         "type": "string",
732                                         "pattern": "^(2\\.4|5)G$"
733                                     }
734                                 },
735                                 "tnn": {
736                                     "type": "string",
737                                     "description": "Indicates Target Network Name (SSID of Wi-Fi AP)",
738                                     "pattern": "^.*$"
739                                 },
740                                 "cd": {
741                                     "type": "string",
742                                     "description": "Indicates credential information of Wi-Fi AP",
743                                     "pattern": "^.*$"
744                                 },
745                                 "wat": {
746                                     "type": "string",
747                                     "enum": ["None", "WEP", "WPA_PSK", "WPA2_PSK"],
748                                     "description": "Indicates Wi-Fi Auth Type"
749                                 },
750                                 "wet": {
751                                     "type": "string",
752                                     "enum": ["None", "WEP_64", "WEP_128", "TKIP", "AES", "TKIP_AES"],
753                                     "description": "Indicates Wi-Fi Encryption Type"
754                                 },
755                                 "swat": {
756                                     "type": "array",
757                                     "description": "Indicates supported Wi-Fi Auth types. It can be multiple",
758                                     "readOnly": true,
759                                     "items":
760                                     {
761                                         "type": "string",
762                                         "enum": ["None", "WEP", "WPA_PSK", "WPA2_PSK"],
763                                         "description": "Indicates Wi-Fi Auth Type"
764                                     }
765                                 }
766                             }
767                         }
768                     }
769                 }

```

```

765         }
766     },
767     "swet": {
768         "type": "array",
769         "description": "Indicates supported Wi-Fi Encryption types. It can be
multiple",
770         "readOnly": true,
771         "items":
772         {
773             "type": "string",
774             "enum": ["None", "WEP_64", "WEP_128", "TKIP", "AES", "TKIP_AES"],
775             "description": "Indicates Wi-Fi Encryption Type"
776         }
777     }
778 },
779 },
780 "required":["swmt", "swf", "swat", "swet", "tnn", "wat", "wet"]
781 }
782 },
783 "type": "object",
784 "allof": [
785     { "$ref": "http://openconnectivityfoundation.github.io/core/schemas/oic.core-
schema.json#/definitions/oic.core"},
786     { "$ref": "#/definitions/oic.r.wificonf" }
787 ]
788 }
789 }
790
791 example: /
792 {
793     "rt": ["oic.r.wificonf"],
794     "swmt": ["A", "B", "G"],
795     "swf": ["2.4G", "5G"],
796     "tnn": "Home_AP_SSID",
797     "cd": "Home_AP_PWD",
798     "wat": "WPA2_PSK",
799     "wet": "TKIP",
800     "swat": ["WPA_PSK", "WPA2_PSK"],
801     "swet": ["TKIP", "AES", "TKIP_AES"]
802 }
803
804 post:
805     description: |
806         Deliver Wi-Fi AP's information for an unboxing device to connect to it.
807
808     body:
809         application/json:
810             schema: /
811             {
812                 "$schema": "http://json-schema.org/draft-v4/schema#",
813                 "description": "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
reserved.",
814                 "id": "http://openconnectivityfoundation.github.io/core-
extensions/schemas/oic.r.wificonf-update-schema.json#",
815                 "definitions": {
816                     "oic.r.wificonf": {
817                         "type": "object",
818                         "properties": {
819                             "tnn": {
820                                 "type": "string",
821                                 "description": "Indicates Target Network Name (SSID of Wi-Fi AP)",
822                                 "pattern": "^.*$"
823                             },
824                             "cd": {
825                                 "type": "string",
826                                 "description": "Indicates credential information of Wi-Fi AP",
827                                 "pattern": "^.*$"
828                             },
829                             "wat": {
830                                 "type": "string",
831                                 "description": "Indicates credential information of Wi-Fi AP",
832                                 "pattern": "^.*$"
833                             }
834                         }
835                     }
836                 }
837             }
838         }
839     }
840 }

```

```

832         "enum": ["None", "WEP", "WPA_PSK", "WPA2_PSK"],
833         "description": "Indicates Wi-Fi Auth Type"
834     },
835     "wet": {
836         "enum": ["None", "WEP_64", "WEP_128", "TKIP", "AES", "TKIP_AES"],
837         "description": "Indicates Wi-Fi Encryption Type"
838     }
839 },
840 "required":["tnn", "wat", "wet"]
841 }
842 },
843 "type": "object",
844 "allOf": [
845     { "$ref": "http://openconnectivityfoundation.github.io/core/schemas/oic.core-
846 schema.json#/definitions/oic.core"},
847     { "$ref": "#/definitions/oic.r.wificonf" }
848 ]
849 }
850
851 example: /
852 {
853     "tnn": "Home_AP_SSID",
854     "cd": "Home_AP_PWD",
855     "wat": "WPA2_PSK",
856     "wet": "AES"
857 }
858
859 responses :
860 200:
861     body:
862     application/json:
863         schema: /
864         {
865             "$schema": "http://json-schema.org/draft-v4/schema#",
866             "description": "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
867 reserved.",
868             "id": "http://openconnectivityfoundation.github.io/core-
869 extensions/schemas/oic.r.wificonf-update-schema.json#",
870             "definitions": {
871                 "oic.r.wificonf": {
872                     "type": "object",
873                     "properties": {
874                         "tnn": {
875                             "type": "string",
876                             "description": "Indicates Target Network Name (SSID of Wi-Fi AP)",
877                             "pattern": "^.*$"
878                         },
879                         "cd": {
880                             "type": "string",
881                             "description": "Indicates credential information of Wi-Fi AP",
882                             "pattern": "^.*$"
883                         },
884                         "wat": {
885                             "enum": ["None", "WEP", "WPA_PSK", "WPA2_PSK"],
886                             "description": "Indicates Wi-Fi Auth Type"
887                         },
888                         "wet": {
889                             "enum": ["None", "WEP_64", "WEP_128", "TKIP", "AES", "TKIP_AES"],
890                             "description": "Indicates Wi-Fi Encryption Type"
891                         }
892                     },
893                     "required":["tnn", "wat", "wet"]
894                 }
895             },
896             "type": "object",
897             "allOf": [
898                 { "$ref": "http://openconnectivityfoundation.github.io/core/schemas/oic.core-

```

```

899 schema.json#/definitions/oic.core"},
900     { "$ref": "#/definitions/oic.r.wificonf" }
901   ]
902 }
903
904   example: /
905     {
906       "tnn": "Home_AP_SSID",
907       "cd": "Home_AP_PWD",
908       "wat": "WPA2_PSK",
909       "wet": "AES"
910     }
911

```

A.3.5 Property Definition

Table 11 Wi-Fi Configuration Resource Baseline Interface Property Definitions

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

A.3.6 CRUDN behaviour

Table 12 Wi-Fi Configuration Resource Baseline Interface CRUDN operations

Resource	Create	Read	Update	Delete	Notify
/WiFiConfResURI		get	post		

916 A.4 Device Configuration

917 A.4.1 Introduction

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

920 A.4.2 Example URI

921 /example/DevConfResURI

922 A.4.3 Resource Type

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

924 A.4.4 RAML Definition

925 `##RAML 0.8`

926 `title: Device Configuration Resource`

927 `version: v0.0.2-20170604`

928 `traits:`

929 `- interface :`

930 `queryParameters:`

931 `if:`

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

933

934 `/example/DevConfResURI:`

935 `description: |`

936 `Device configuration resource stores a preference of device settings like`

937 `device name`

938 `Vender-specific information can be added to the resource.`

939

940 `is : ['interface']`

941 `get:`

942 `description: |`

943 `Retrieve various settings regarding to device-specific settings`

944 `1`

945 `Device name (human-friendly name to be detected by mediator during`

946 `easy setup)`

947

948 `responses :`

949 `200:`

950 `body:`

951 `application/json:`

952 `schema: |`

953 `{`
954 `"$schema": "http://json-schema.org/draft-04/schema#",`
955 `"description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights`
956 `reserved.",`

957 `"id": "http://openconnectivityfoundation.github.io/core-`
958 `extensions/schemas/oic.r.devconf-schema.json#",`

959 `"definitions": {`

960 `"oic.r.devconf": {`

961 `"type": "object",`

962 `"oneOf": [`

963 `{`

964 `"properties": {`

965 `"dn": {`

966 `"type": "string",`

967 `"description": "Indicates a pre-configured device name in language`

968 `indicated by 'dl' in /oic/con; presented by enrollee device to mediator device during easy-setup`

969 `process",`

970 `"pattern": "^.*$",`


```

971         "readOnly": true
972     }
973 },
974     "required":["dn"]
975 },
976     {
977         "properties": {
978             "dn": {
979                 "type": "array",
980                 "items": {
981                     "type": "object",
982                     "properties": {
983                         "language": {
984                             "$ref":
985 "http://openconnectivityfoundation.github.io/core/schemas/oic.types-
986 schema.json#/definitions/language-tag",
987                             "readOnly": true,
988                             "description": "An RFC 5646 language tag."
989                         },
990                         "value": {
991                             "type": "string",
992                             "description": "Pre-configured device name in the indicated
993 language.",
994                             "pattern": "^.*$",
995                             "readOnly": true
996                         }
997                     }
998                 },
999                 "minItems" : 1,
1000                 "readOnly": true,
1001                 "description": "Localized device name."
1002             }
1003         },
1004         "required": ["dn"]
1005     }
1006 ]
1007 }
1008 },
1009     "type": "object",
1010     "allOf": [
1011         { "$ref": "http://openconnectivityfoundation.github.io/core/schemas/oic.core-
1012 schema.json#/definitions/oic.core"},
1013         { "$ref": "#/definitions/oic.r.devconf" }
1014     ]
1015 }
1016
1017     example: /
1018     {
1019         "rt": ["oic.r.devconf"],
1020         "dn" : "My Refrigerator"
1021     }
1022

```

A.4.5 Property Definition

Table 13 Device Configuration Property Definitions

Property name	Value type	Mandatory	Access mode	Description
dn	array: see schema	yes	Read Only	Localized device name.
value (dn)	string		Read Only	Pre-configured device name in the indicated language.
language (dn)	multiple types: see schema		Read Only	An RFC 5646 language tag.

1025 **A.4.6 CRUDN behaviour**

1026 **Table 14 Device Configuration CRUDN operations**

Resource	Create	Read	Update	Delete	Notify
/example/DevConfResURI		get			

1027

1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086

Annex B (informative)

Swagger2.0 definitions

B.1 Device Configuration

B.1.1 Introduction

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

B.1.2 Example URI

/example/DevConfResURI

B.1.3 Resource Type

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

B.1.4 Swagger2.0 Definition

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

```

1087         "schema": { "$ref": "#/definitions/DevConf" }
1088     }
1089 }
1090 }
1091 }
1092 },
1093 "parameters": {
1094     "interface" : {
1095         "in" : "query",
1096         "name" : "if",
1097         "type" : "string",
1098         "enum" : ["oic.if.baseline", "oic.if.r"]
1099     }
1100 },
1101 "definitions": {
1102     "DevConf" : {
1103         "properties": {
1104             "rt" :
1105                 {
1106                     "description": "Resource Type of the Resource",
1107                     "items": {
1108                         "maxLength": 64,
1109                         "type": "string"
1110                     },
1111                     "minItems": 1,
1112                     "readOnly": true,
1113                     "type": "array"
1114                 },
1115             "n" :
1116                 {
1117                     "description": "Friendly name of the resource",
1118                     "maxLength": 64,
1119                     "readOnly": true,
1120                     "type": "string"
1121                 },
1122             "id" :
1123                 {
1124                     "description": "Instance ID of this specific resource",
1125                     "maxLength": 64,
1126                     "readOnly": true,
1127                     "type": "string"
1128                 },
1129             "if" :
1130                 {
1131                     "description": "The interface set supported by this resource",
1132                     "items": {
1133                         "enum": [
1134                             "oic.if.baseline",
1135                             "oic.if.ll",
1136                             "oic.if.b",
1137                             "oic.if.lb",
1138                             "oic.if.rw",
1139                             "oic.if.r",
1140                             "oic.if.a",
1141                             "oic.if.s"
1142                         ],
1143                         "type": "string"
1144                     },
1145                     "minItems": 1,
1146                     "readOnly": true,
1147                     "type": "array"
1148                 }
1149             },
1150             "type" : "object"
1151         }
1152     }
1153 }
1154 }
1155 }
1156 }

```

1157 }
1158

1159 B.1.5 Property Definition

1160 **Table 15 The properties definitions of the resource**

Property name	Value type	Mandatory	Access mode	Description
if	array: see schema		Read Only	The interface set supported by this resource
n	string		Read Only	Friendly name of the resource
rt	array: see schema		Read Only	Resource Type of the Resource
id	string		Read Only	Instance ID of this specific resource

1161 B.1.6 CRUDN behaviour

1162 **Table 16 The CRUDN operations of the resource**

Resource	Create	Read	Update	Delete	Notify
/example/DevConfResURI		get			observe

1163 B.2 Easy Setup

1164 B.2.1 Introduction

1165 Easy Setup resource stores useful information including current status of
1166 unboxing device and last error code which are produced in a process of
1167 easy setup.

1168 Note that, Easy Setup resource is a type of collection resource, which
1169 contains links to WiFiConf, DevConf resources and may additionally contain
1170 links to other resources.

1171 B.2.2 Example URI

1172 /EasySetupResURI

1173 B.2.3 Resource Type

1174 The resource type (rt) is defined as: ['oic.r.easysetup'].

1175 B.2.4 Swagger2.0 Definition

```
1176 {  
1177   "swagger": "2.0",  
1178   "info": {  
1179     "title": "Easy Setup Collection Link List Interface",  
1180     "version": "v0.0.3-20170611",  
1181     "license": {  
1182       "name": "copyright 2016-2017 Open Connectivity Foundation, Inc. All rights reserved.",  
1183       "x-description": "Redistribution and use in source and binary forms, with or without  
1184 modification, are permitted provided that the following conditions are met:\n      1.  
1185 Redistributions of source code must retain the above copyright notice, this list of conditions and  
1186 the following disclaimer.\n      2. Redistributions in binary form must reproduce the above  
1187 copyright notice, this list of conditions and the following disclaimer in the documentation and/or  
1188 other materials provided with the distribution.\n      THIS SOFTWARE IS PROVIDED BY THE Open  
1189 Connectivity Foundation, INC. \\\n      AS IS\\n AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
```

```

1190 LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR
1191 WARRANTIES OF NON-INFRINGEMENT, ARE DISCLAIMED.\n          IN NO EVENT SHALL THE Open Connectivity
1192 Foundation, INC. OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
1193 EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
1194 OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)\n          HOWEVER CAUSED AND
1195 ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
1196 OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY
1197 OF SUCH DAMAGE.\n"
1198     }
1199   },
1200   "schemes": ["http"],
1201   "consumes": ["application/json"],
1202   "produces": ["application/json"],
1203   "paths": {
1204     "/EasySetupResURI?if=oic.if.ll" : {
1205       "get": {
1206         "description": "Easy Setup resource stores useful information including current status
1207 of\nunboxing device and last error code which are produced in a process of\neasy setup.\nNote that,
1208 Easy Setup resource is a type of collection resource, which\ncontains links to WiFiConf, DevConf
1209 resources and may additionally contain\nlinks to other resources.\nRetrieve useful information
1210 during easy setup process :\n1. A current status in easy setup process.\n2. A last error code
1211 describing reason for failure occurred at the last\n time.\n",
1212         "parameters": [
1213           { "$ref": "#/parameters/interface-ll" }
1214         ],
1215         "responses": {
1216           "200": {
1217             "description": "",
1218             "x-example":
1219               [
1220                 {
1221                   "href": "/EasySetupResURI",
1222                   "rt": ["oic.r.easyssetup", "oic.wk.col"],
1223                   "if": ["oic.if.b"],
1224                   "p": {"bm": 3},
1225                   "eps": [
1226                     {"ep": "coaps://[fe80::bld6]:1111", "pri": 2}
1227                   ],
1228                   "rel": ["self", "item"]
1229                 },
1230                 {
1231                   "href": "/WiFiConfResURI",
1232                   "rt": ["oic.r.wificonf"],
1233                   "if": ["oic.if.baseline"],
1234                   "p": {"bm": 3},
1235                   "eps": [
1236                     {"ep": "coaps://[fe80::bld6]:1111", "pri": 2}
1237                   ]
1238                 },
1239                 {
1240                   "href": "/DevConfResURI",
1241                   "rt": ["oic.r.devconf"],
1242                   "if": ["oic.if.baseline"],
1243                   "p": {"bm": 3},
1244                   "eps": [
1245                     {"ep": "coaps://[fe80::bld6]:1111", "pri": 2}
1246                   ]
1247                 }
1248               ]
1249         }
1250       },
1251       "schema": { "$ref": "#/definitions/slinks" }
1252     }
1253   }
1254 },
1255 "/EasySetupResURI?if=oic.if.b" : {
1256   "get": {
1257     "description": "Easy Setup resource stores useful information including current status
1258 of\nunboxing device and last error code which are produced in a process of\neasy setup.\nNote that,
1259 Easy Setup resource is a type of collection resource, which\ncontains links to WiFiConf, DevConf
1260 resources and may additionally contain\nlinks to other resources.\nRetrieve useful information

```

```

1261 during easy setup process :\n1. A current status in easy setup process.\n2. A last error code
1262 describing reason for failure occurred at the last\n time.\n",
1263     "parameters": [
1264         { "$ref": "#/parameters/interface-batch" }
1265     ],
1266     "responses": {
1267         "200": {
1268             "description": "",
1269             "x-example":
1270                 [
1271                     {
1272                         "href": "/EasySetupResURI",
1273                         "rep": {
1274                             "ps": 0,
1275                             "lec": 0,
1276                             "cn": [1]
1277                         }
1278                     },
1279                     {
1280                         "href": "/WiFiConfResURI",
1281                         "rep": {
1282                             "swmt": ["A", "B", "G"],
1283                             "swf": ["2.4G", "5G"],
1284                             "tnn": "Home_AP_SSID",
1285                             "cd": "Home_AP_PWD",
1286                             "wat": "WPA2_PSK",
1287                             "wet": "AES"
1288                         }
1289                     },
1290                     {
1291                         "href": "/DevConfResURI",
1292                         "rep": {
1293                             "dn": "My Refrigerator"
1294                         }
1295                     }
1296                 ]
1297             ,
1298             "schema": { "$ref": "#/definitions/sbatch" }
1299         }
1300     }
1301 },
1302 "post": {
1303     "description": "Able to deliver Wi-Fi, Device configuration and other
1304 configuration\ninformation in a batch by utilizing 'batch' interface.\nIf you want to deliver Wi-Fi
1305 and Device configuration information in a batch,\nyou can write all properties you want to send
1306 with a 'batch' interface.\nThe below example is the case to send Easy Setup and Wi-Fi
1307 configuration\n(i.e. connection type, target network, auth type information) in a batch.\n",
1308     "parameters": [
1309         { "$ref": "#/parameters/interface-batch" },
1310         {
1311             "name": "body",
1312             "in": "body",
1313             "required": true,
1314             "schema": { "$ref": "#/definitions/sbatch-update" },
1315             "x-example":
1316                 [
1317                     {
1318                         "href": "/EasySetupResURI",
1319                         "rep": {
1320                             "cn": [1]
1321                         }
1322                     },
1323                     {
1324                         "href": "/WiFiConfResURI",
1325                         "rep": {
1326                             "tnn": "Home_AP_SSID",
1327                             "cd": "Home_AP_PWD",
1328                             "wat": "WPA2_PSK",
1329                             "wet": "AES"
1330                         }
1331                     }
1332                 ]
1333         }
1334     ]
1335 }

```

```

1332     ]
1333   }
1334 ],
1335 "responses": {
1336   "200": {
1337     "description": "",
1338     "x-example":
1339     [
1340       {
1341         "href": "/EasySetupResURI",
1342         "rep": {
1343           "ps": 0,
1344           "lec": 0,
1345           "cn": [1]
1346         }
1347       },
1348       {
1349         "href": "/WiFiConfResURI",
1350         "rep": {
1351           "swmt": ["A", "B", "G"],
1352           "swf": ["2.4G", "5G"],
1353           "tnn": "Home_AP_SSID",
1354           "cd": "Home_AP_PWD",
1355           "wat": "WPA2_PSK",
1356           "wet": "AES"
1357         }
1358       },
1359       {
1360         "href": "/DevConfResURI",
1361         "rep": {
1362           "dn": "My Refrigerator"
1363         }
1364       }
1365     ]
1366   },
1367   "schema": { "$ref": "#/definitions/sbatch" }
1368 }
1369 }
1370 },
1371 "/EasySetupResURI?if=oic.if.baseline" : {
1372 "get": {
1373   "description": "Easy Setup resource stores useful information including current status
1374 of\nunboxing device and last error code which are produced in a process of\neasy setup.\nNote that,
1375 Easy Setup resource is a type of collection resource, which\ncontains links to WiFiConf, DevConf
1376 resources and may additionally contain\nlinks to other resources.\nRetrieve useful information
1377 during easy setup process :\n 1. A current status in easy setup process.\n 2. A last error code
1378 describing reason for failure occurred at the last\n      time.\n",
1379   "parameters": [
1380     { "$ref": "#/parameters/interface-baseline" }
1381   ],
1382   "responses": {
1383     "200": {
1384       "description": "",
1385       "x-example":
1386       {
1387         "rt" : ["oic.r.easyssetup", "oic.wk.col"],
1388         "if" : ["oic.if.ll", "oic.if.baseline", "oic.if.b"],
1389         "ps" : 0,
1390         "lec": 0,
1391         "cn": [1],
1392         "links": [
1393           {
1394             "href": "/EasySetupResURI",
1395             "rt": ["oic.r.easyssetup", "oic.wk.col"],
1396             "if": ["oic.if.b"],
1397             "p":{"bm":3},
1398             "eps": [
1399               { "ep": "coaps://[fe80::b1d6]:1111", "pri": 2}
1400             ],
1401             "rel":["self", "item"]
1402           }

```



```

1403         },
1404         {
1405             "href": "/WiFiConfResURI",
1406             "rt": ["oic.r.wificonf"],
1407             "if": ["oic.if.baseline"],
1408             "p":{"bm":3},
1409             "eps": [
1410                 {"ep": "coaps://[fe80::bld6]:1111", "pri": 2}
1411             ]
1412         },
1413         {
1414             "href": "/DevConfResURI",
1415             "rt": ["oic.r.devconf"],
1416             "if": ["oic.if.baseline"],
1417             "p":{"bm":3},
1418             "eps": [
1419                 {"ep": "coaps://[fe80::bld6]:1111", "pri": 2}
1420             ]
1421         }
1422     ]
1423 }
1424 ,
1425 "schema": { "$ref": "#/definitions/EasySetup" }
1426 }
1427 }
1428 }
1429 }
1430 },
1431 "parameters": {
1432     "interface-ll" : {
1433         "in" : "query",
1434         "name" : "if",
1435         "type" : "string",
1436         "enum" : ["oic.if.ll"]
1437     },
1438     "interface-baseline" : {
1439         "in" : "query",
1440         "name" : "if",
1441         "type" : "string",
1442         "enum" : ["oic.if.baseline"]
1443     },
1444     "interface-all" : {
1445         "in" : "query",
1446         "name" : "if",
1447         "type" : "string",
1448         "enum" : ["oic.if.baseline", "oic.if.ll", "oic.if.b"]
1449     },
1450     "interface-batch" : {
1451         "in" : "query",
1452         "name" : "if",
1453         "type" : "string",
1454         "enum" : ["oic.if.b"]
1455     }
1456 },
1457 "definitions": {
1458     "slinks" : {
1459         "items" :
1460         {
1461             "properties": {
1462                 "anchor": {
1463                     "description": "This is used to override the context URI e.g. override the URI of the
1464 containing collection.",
1465                     "format": "uri",
1466                     "maxLength": 256,
1467                     "type": "string"
1468                 },
1469                 "di": {
1470                     "description": "The Device ID formatted according to IETF RFC 4122.",
1471                     "pattern": "^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-
1472 9]{12}$",
1473                     "type": "string"

```

```

1474     },
1475     "eps": {
1476       "description": "the Endpoint information of the target Resource",
1477       "items": {
1478         "properties": {
1479           "ep": {
1480             "description": "Transport Protocol Suite + Endpoint Locator",
1481             "format": "uri",
1482             "type": "string"
1483           },
1484           "pri": {
1485             "description": "The priority among multiple Endpoints",
1486             "minimum": 1,
1487             "type": "integer"
1488           }
1489         },
1490         "type": "object"
1491       },
1492       "type": "array"
1493     },
1494     "href": {
1495       "description": "This is the target URI, it can be specified as a Relative Reference or
1496 fully-qualified URI.",
1497       "format": "uri",
1498       "maxLength": 256,
1499       "type": "string"
1500     },
1501     "if": {
1502       "description": "The interface set supported by this resource",
1503       "items": {
1504         "enum": [
1505           "oic.if.baseline",
1506           "oic.if.ll",
1507           "oic.if.b",
1508           "oic.if.rw",
1509           "oic.if.r",
1510           "oic.if.a",
1511           "oic.if.s"
1512         ],
1513         "type": "string"
1514       },
1515       "minItems": 1,
1516       "type": "array"
1517     },
1518     "ins": {
1519       "description": "The instance identifier for this web link in an array of web links -
1520 used in collections",
1521       "type": "integer"
1522     },
1523     "p": {
1524       "description": "Specifies the framework policies on the Resource referenced by the
1525 target URI",
1526       "properties": {
1527         "bm": {
1528           "description": "Specifies the framework policies on the Resource referenced by the
1529 target URI for e.g. observable and discoverable",
1530           "type": "integer"
1531         }
1532       },
1533       "required": [
1534         "bm"
1535       ],
1536       "type": "object"
1537     },
1538     "rel": {
1539       "description": "The relation of the target URI referenced by the link to the context
1540 URI",
1541       "oneOf": [
1542         {
1543           "default": [
1544             "hosts"

```

```

1545         ],
1546         "items": {
1547             "maxLength": 64,
1548             "type": "string"
1549         },
1550         "minItems": 1,
1551         "type": "array"
1552     },
1553     {
1554         "default": "hosts",
1555         "maxLength": 64,
1556         "type": "string"
1557     }
1558 ]
1559 },
1560 "rt": {
1561     "description": "Resource Type of the Resource",
1562     "items": {
1563         "maxLength": 64,
1564         "type": "string"
1565     },
1566     "minItems": 1,
1567     "type": "array"
1568 },
1569 "title": {
1570     "description": "A title for the link relation. Can be used by the UI to provide a
1571 context.",
1572     "maxLength": 64,
1573     "type": "string"
1574 },
1575 "type": {
1576     "default": "application/cbor",
1577     "description": "A hint at the representation of the resource referenced by the target
1578 URI. This represents the media types that are used for both accepting and emitting.",
1579     "items": {
1580         "maxLength": 64,
1581         "type": "string"
1582     },
1583     "minItems": 1,
1584     "type": "array"
1585 }
1586 },
1587 "required": [
1588     "href",
1589     "rt",
1590     "if"
1591 ],
1592 "type": "object"
1593 }
1594
1595 , "type" :
1596     "array"
1597
1598 , "title" :
1599     "EasySetup Object Links List Schema (auto merged)"
1600
1601 }
1602
1603 , "sbatch" : {
1604     "title" :
1605         "Collection Batch Retrieve Format (auto merged)"
1606
1607 , "minItems" :
1608     1
1609
1610 , "items" :
1611     {
1612         "additionalProperties": true,
1613         "properties": {
1614             "href": {
1615                 "description": "URI of the target resource relative assuming the collection URI as

```

```

1616 anchor",
1617     "format": "uri",
1618     "maxLength": 256,
1619     "type": "string"
1620 },
1621 "rep": {
1622     "oneOf": [
1623         {
1624             "description": "The response payload from a single resource",
1625             "type": "object"
1626         },
1627         {
1628             "description": " The response payload from a collection (batch) resource",
1629             "items": {
1630                 "properties": {
1631                     "anchor": {
1632                         "description": "This is used to override the context URI e.g. override the
1633 URI of the containing collection.",
1634                         "format": "uri",
1635                         "maxLength": 256,
1636                         "type": "string"
1637                     },
1638                     "di": {
1639                         "allOf": [
1640                             {
1641                                 "description": "Format pattern according to IETF RFC 4122.",
1642                                 "pattern": "^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-
1643 [a-fA-F0-9]{12}$",
1644                                 "type": "string"
1645                             },
1646                             {
1647                                 "description": "The device ID"
1648                             }
1649                         ]
1650                     },
1651                     "eps": {
1652                         "description": "the Endpoint information of the target Resource",
1653                         "items": {
1654                             "properties": {
1655                                 "ep": {
1656                                     "description": "Transport Protocol Suite + Endpoint Locator",
1657                                     "format": "uri",
1658                                     "type": "string"
1659                                 },
1660                                 "pri": {
1661                                     "description": "The priority among multiple Endpoints",
1662                                     "minimum": 1,
1663                                     "type": "integer"
1664                                 }
1665                             },
1666                             "type": "object"
1667                         },
1668                         "type": "array"
1669                     },
1670                     "href": {
1671                         "description": "This is the target URI, it can be specified as a Relative
1672 Reference or fully-qualified URI.",
1673                         "format": "uri",
1674                         "maxLength": 256,
1675                         "type": "string"
1676                     },
1677                     "if": {
1678                         "description": "The interface set supported by this resource",
1679                         "items": {
1680                             "enum": [
1681                                 "oic.if.baseline",
1682                                 "oic.if.ll",
1683                                 "oic.if.b",
1684                                 "oic.if.rw",
1685                                 "oic.if.r",
1686                                 "oic.if.a",

```

```

1687         "oic.if.s"
1688     ],
1689     "type": "string"
1690 },
1691     "minItems": 1,
1692     "type": "array"
1693 },
1694     "ins": {
1695         "description": "The instance identifier for this web link in an array of web
1696 links - used in collections",
1697         "type": "integer"
1698     },
1699     "p": {
1700         "description": "Specifies the framework policies on the Resource referenced
1701 by the target URI",
1702         "properties": {
1703             "bm": {
1704                 "description": "Specifies the framework policies on the Resource
1705 referenced by the target URI for e.g. observable and discoverable",
1706                 "type": "integer"
1707             }
1708         },
1709         "required": [
1710             "bm"
1711         ],
1712         "type": "object"
1713     },
1714     "rel": {
1715         "description": "The relation of the target URI referenced by the link to the
1716 context URI",
1717         "oneOf": [
1718             {
1719                 "default": [
1720                     "hosts"
1721                 ],
1722                 "items": {
1723                     "maxLength": 64,
1724                     "type": "string"
1725                 },
1726                 "minItems": 1,
1727                 "type": "array"
1728             },
1729             {
1730                 "default": "hosts",
1731                 "maxLength": 64,
1732                 "type": "string"
1733             }
1734         ]
1735     },
1736     "rt": {
1737         "description": "Resource Type of the Resource",
1738         "items": {
1739             "maxLength": 64,
1740             "type": "string"
1741         },
1742         "minItems": 1,
1743         "type": "array"
1744     },
1745     "title": {
1746         "description": "A title for the link relation. Can be used by the UI to
1747 provide a context.",
1748         "maxLength": 64,
1749         "type": "string"
1750     },
1751     "type": {
1752         "default": "application/cbor",
1753         "description": "A hint at the representation of the resource referenced by
1754 the target URI. This represents the media types that are used for both accepting and emitting.",
1755         "items": {
1756             "maxLength": 64,
1757             "type": "string"

```

```

1758         },
1759         "minItems": 1,
1760         "type": "array"
1761     }
1762 },
1763     "required": [
1764         "href",
1765         "rt",
1766         "if"
1767     ],
1768     "type": "object"
1769 },
1770 "type": "array"
1771 }
1772 ]
1773 }
1774 },
1775 "required": [
1776     "href",
1777     "rep"
1778 ],
1779 "type": "object"
1780 }
1781 , "type" :
1782     "array"
1783 }
1784 ,
1785 "sbatch-update" : {
1786     "title" :
1787         "Collection Batch Update Format (auto merged)"
1788     , "minItems" :
1789         1
1790     , "items" :
1791         {
1792             "$ref": "#/definitions/oic.batch-update.item"
1793         }
1794     , "type" :
1795         "array"
1796 }
1797 ,
1798 "EasySetup" : {
1799     "properties": {
1800         "rt" :
1801             {
1802                 "items": {
1803                     "enum": [
1804                         "oic.r.easyssetup",
1805                         "oic.wk.col"
1806                     ]
1807                 },
1808                 "maxItems": 2,
1809                 "minItems": 2,
1810                 "type": "array",
1811                 "uniqueItems": true
1812             },
1813     "ps" :
1814         {
1815             "description": "Indicates the easy setup status of the device. (0: Need to Setup, 1:
1816 Connecting to Enroller, 2: Connected to Enroller, 3: Failed to Connect to Enroller, 4~254:
1817 Reserved, 255: EOF)",
1818             "enum": [
1819                 0,
1820                 1,
1821                 2,

```

```

1829     3
1830   ],
1831   "readOnly": true,
1832   "type": "integer"
1833 },
1834
1835   "lec" :
1836   {
1837     "description": "Indicates a failure reason (0: NO error, 1: A given SSID is not found, 2:
1838 Wi-Fi's password is wrong, 3: IP address is not allocated, 4: No internet connection, 5: Timeout,
1839 6: Wi-Fi Auth Type is not supported by the Enrollee, 7: Wi-Fi Encryption Type is not supported by
1840 the Enrollee, 8: Wi-Fi Auth Type is wrong (failure while connecting to the Enroller), 9: Wi-Fi
1841 Encryption Type is wrong (failure while connecting to the Enroller), 10~254: Reserved, 255: Unknown
1842 error)",
1843     "enum": [
1844       0,
1845       1,
1846       2,
1847       3,
1848       4,
1849       5,
1850       6,
1851       7,
1852       8,
1853       9,
1854       255
1855     ],
1856     "readOnly": true,
1857     "type": "integer"
1858   },
1859
1860   "cn" :
1861   {
1862     "description": "Indicates an array of connection types that trigger an attempt to connect
1863 to the Enroller to start.",
1864     "items": {
1865       "description": "Connection type to attempt. (1 : Wi-Fi, 2 : other entities / transports
1866 to be added in future (e.g. Connect to cloud / BLE))",
1867       "type": "integer"
1868     },
1869     "type": "array"
1870   },
1871
1872   "links" :
1873   {
1874     "description": "A set of simple or individual OIC Links.",
1875     "items": {
1876       "properties": {
1877         "anchor": {
1878           "description": "This is used to override the context URI e.g. override the URI of
1879 the containing collection.",
1880           "format": "uri",
1881           "maxLength": 256,
1882           "type": "string"
1883         },
1884         "di": {
1885           "description": "The Device ID formatted according to IETF RFC 4122.",
1886           "pattern": "^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-
1887 9]{12}$",
1888           "type": "string"
1889         },
1890         "eps": {
1891           "description": "the Endpoint information of the target Resource",
1892           "items": {
1893             "properties": {
1894               "ep": {
1895                 "description": "Transport Protocol Suite + Endpoint Locator",
1896                 "format": "uri",
1897                 "type": "string"
1898               },
1899               "pri": {

```

```

1900         "description": "The priority among multiple Endpoints",
1901         "minimum": 1,
1902         "type": "integer"
1903     },
1904     },
1905     "type": "object"
1906 },
1907 "type": "array"
1908 },
1909 "href": {
1910     "description": "This is the target URI, it can be specified as a Relative Reference
1911 or fully-qualified URI.",
1912     "format": "uri",
1913     "maxLength": 256,
1914     "type": "string"
1915 },
1916 "if": {
1917     "description": "The interface set supported by this resource",
1918     "items": {
1919         "enum": [
1920             "oic.if.baseline",
1921             "oic.if.ll",
1922             "oic.if.b",
1923             "oic.if.rw",
1924             "oic.if.r",
1925             "oic.if.a",
1926             "oic.if.s"
1927         ],
1928         "type": "string"
1929     },
1930     "minItems": 1,
1931     "type": "array"
1932 },
1933 "ins": {
1934     "description": "The instance identifier for this web link in an array of web links
1935 - used in collections",
1936     "type": "integer"
1937 },
1938 "p": {
1939     "description": "Specifies the framework policies on the Resource referenced by the
1940 target URI",
1941     "properties": {
1942         "bm": {
1943             "description": "Specifies the framework policies on the Resource referenced by
1944 the target URI for e.g. observable and discoverable",
1945             "type": "integer"
1946         }
1947     },
1948     "required": [
1949         "bm"
1950     ],
1951     "type": "object"
1952 },
1953 "rel": {
1954     "description": "The relation of the target URI referenced by the link to the
1955 context URI",
1956     "oneOf": [
1957         {
1958             "default": [
1959                 "hosts"
1960             ],
1961             "items": {
1962                 "maxLength": 64,
1963                 "type": "string"
1964             },
1965             "minItems": 1,
1966             "type": "array"
1967         },
1968         {
1969             "default": "hosts",
1970             "maxLength": 64,

```



```

1971         "type": "string"
1972     }
1973 ]
1974 },
1975 "rt": {
1976     "description": "Resource Type of the Resource",
1977     "items": {
1978         "maxLength": 64,
1979         "type": "string"
1980     },
1981     "minItems": 1,
1982     "type": "array"
1983 },
1984 "title": {
1985     "description": "A title for the link relation. Can be used by the UI to provide a
1986 context.",
1987     "maxLength": 64,
1988     "type": "string"
1989 },
1990 "type": {
1991     "default": "application/cbor",
1992     "description": "A hint at the representation of the resource referenced by the
1993 target URI. This represents the media types that are used for both accepting and emitting.",
1994     "items": {
1995         "maxLength": 64,
1996         "type": "string"
1997     },
1998     "minItems": 1,
1999     "type": "array"
2000 }
2001 },
2002 "required": [
2003     "href",
2004     "rt",
2005     "if"
2006 ],
2007 "type": "object"
2008 },
2009 "type": "array"
2010 },
2011
2012 "n" :
2013 {
2014     "description": "Friendly name of the resource",
2015     "maxLength": 64,
2016     "readOnly": true,
2017     "type": "string"
2018 },
2019
2020 "rts" :
2021 {
2022     "description": "Resource Type of the Resource",
2023     "items": {
2024         "maxLength": 64,
2025         "type": "string"
2026     },
2027     "minItems": 1,
2028     "readOnly": true,
2029     "type": "array"
2030 },
2031
2032 "id" :
2033 {
2034     "description": "Instance ID of this specific resource",
2035     "maxLength": 64,
2036     "readOnly": true,
2037     "type": "string"
2038 },
2039
2040 "rts-m" :
2041 {

```

```

2042         "description": "Resource Type of the Resource",
2043         "items": {
2044             "maxLength": 64,
2045             "type": "string"
2046         },
2047         "minItems": 1,
2048         "readOnly": true,
2049         "type": "array"
2050     },
2051     "if" :
2052     {
2053         "description": "The interface set supported by this resource",
2054         "items": {
2055             "enum": [
2056                 "oic.if.baseline",
2057                 "oic.if.ll",
2058                 "oic.if.b",
2059                 "oic.if.lb",
2060                 "oic.if.rw",
2061                 "oic.if.r",
2062                 "oic.if.a",
2063                 "oic.if.s"
2064             ],
2065             "type": "string"
2066         },
2067         "minItems": 1,
2068         "readOnly": true,
2069         "type": "array"
2070     }
2071     },
2072     },
2073     },
2074     "type" : "object"
2075 }
2076 , "oic.batch-update.item" :
2077 {
2078     "additionalProperties": true,
2079     "description": "array of resource representations to apply to the batch collection, using
2080 href to indicate which resource(s) in the batch to update. If the href property is empty,
2081 effectively making the URI reference to the collection itself, the representation is to be applied
2082 to all resources in the batch",
2083     "properties": {
2084         "href": {
2085             "description": "URI of the target resource relative assuming the collection URI as
2086 anchor",
2087             "format": "uri",
2088             "maxLength": 256,
2089             "type": "string"
2090         },
2091         "rep": {
2092             "oneOf": [
2093                 {
2094                     "description": "The response payload from a single resource",
2095                     "type": "object"
2096                 },
2097                 {
2098                     "description": " The response payload from a collection (batch) resource",
2099                     "items": {
2100                         "$ref": "#/definitions/oic.oic-link"
2101                     },
2102                     "type": "array"
2103                 }
2104             ]
2105         }
2106     },
2107     "required": [
2108         "href",
2109         "rep"
2110     ],
2111     "type": "object"
2112 }

```

```

2113
2114     , "oic.oic-link" :
2115       {
2116         "properties": {
2117           "anchor": {
2118             "description": "This is used to override the context URI e.g. override the URI of the
2119 containing collection.",
2120             "format": "uri",
2121             "maxLength": 256,
2122             "type": "string"
2123           },
2124           "di": {
2125             "allOf": [
2126               {
2127                 "$ref": "#/definitions/uuid"
2128               },
2129               {
2130                 "description": "The device ID"
2131               }
2132             ]
2133           },
2134           "eps": {
2135             "description": "the Endpoint information of the target Resource",
2136             "items": {
2137               "properties": {
2138                 "ep": {
2139                   "description": "Transport Protocol Suite + Endpoint Locator",
2140                   "format": "uri",
2141                   "type": "string"
2142                 },
2143                 "pri": {
2144                   "description": "The priority among multiple Endpoints",
2145                   "minimum": 1,
2146                   "type": "integer"
2147                 }
2148               },
2149               "type": "object"
2150             },
2151             "type": "array"
2152           },
2153           "href": {
2154             "description": "This is the target URI, it can be specified as a Relative Reference or
2155 fully-qualified URI.",
2156             "format": "uri",
2157             "maxLength": 256,
2158             "type": "string"
2159           },
2160           "if": {
2161             "description": "The interface set supported by this resource",
2162             "items": {
2163               "enum": [
2164                 "oic.if.baseline",
2165                 "oic.if.ll",
2166                 "oic.if.b",
2167                 "oic.if.rw",
2168                 "oic.if.r",
2169                 "oic.if.a",
2170                 "oic.if.s"
2171               ],
2172               "type": "string"
2173             },
2174             "minItems": 1,
2175             "type": "array"
2176           },
2177           "ins": {
2178             "description": "The instance identifier for this web link in an array of web links - used
2179 in collections",
2180             "type": "integer"
2181           },
2182           "p": {
2183             "description": "Specifies the framework policies on the Resource referenced by the target

```

```

2184 URI",
2185     "properties": {
2186         "bm": {
2187             "description": "Specifies the framework policies on the Resource referenced by the
2188 target URI for e.g. observable and discoverable",
2189             "type": "integer"
2190         }
2191     },
2192     "required": [
2193         "bm"
2194     ],
2195     "type": "object"
2196 },
2197 "rel": {
2198     "description": "The relation of the target URI referenced by the link to the context
2199 URI",
2200     "oneOf": [
2201         {
2202             "default": [
2203                 "hosts"
2204             ],
2205             "items": {
2206                 "maxLength": 64,
2207                 "type": "string"
2208             },
2209             "minItems": 1,
2210             "type": "array"
2211         },
2212         {
2213             "default": "hosts",
2214             "maxLength": 64,
2215             "type": "string"
2216         }
2217     ]
2218 },
2219 "rt": {
2220     "description": "Resource Type of the Resource",
2221     "items": {
2222         "maxLength": 64,
2223         "type": "string"
2224     },
2225     "minItems": 1,
2226     "type": "array"
2227 },
2228 "title": {
2229     "description": "A title for the link relation. Can be used by the UI to provide a
2230 context.",
2231     "maxLength": 64,
2232     "type": "string"
2233 },
2234 "type": {
2235     "default": "application/cbor",
2236     "description": "A hint at the representation of the resource referenced by the target
2237 URI. This represents the media types that are used for both accepting and emitting.",
2238     "items": {
2239         "maxLength": 64,
2240         "type": "string"
2241     },
2242     "minItems": 1,
2243     "type": "array"
2244 }
2245 },
2246 "required": [
2247     "href",
2248     "rt",
2249     "if"
2250 ],
2251 "type": "object"
2252 }
2253
2254 , "uuid" :

```

```

2255     {
2256       "description": "Format pattern according to IETF RFC 4122.",
2257       "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}$",
2258       "type": "string"
2259     }
2260
2261     , "oic.wk.col-batch-update" :
2262     {
2263       "description": "array of resource representations to apply to the batch collection, using
2264 href to indicate which resource(s) in the batch to update. If the href property is empty,
2265 effectively making the URI reference to the collection itself, the representation is to be applied
2266 to all resources in the batch",
2267       "items": {
2268         "$ref": "#/definitions/oic.batch-update.item"
2269       },
2270       "minItems": 1,
2271       "type": "array"
2272     }
2273   }
2274 }
2275 }
2276

```

B.2.5 Property Definition

Table 17 The properties definitions of the resource

Property name	Value type	Mandatory	Access mode	Description
rts	array: see schema	No	Read Only	Resource Type of the Resource
n	string	No	Read Only	Friendly name of the resource
cn	array: see schema	No	Read Write	Indicates an array of connection types that trigger an attempt to connect to the Enroller to start.
id	string	No	Read Only	Instance ID of this specific resource
lec	integer	No	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)
ps	integer	No	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)
if	array: schema see	Yes	Read Only	The interface set supported by this resource
links	array: schema see	No	Read Write	A set of simple or individual OIC Links.
rts-m	array: schema see	No	Read Only	Resource Type of the Resource
rt	array: schema see	Yes	Read Write	
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	
ins	integer	No	Read Write	

				The instance identifier for this web link in an array of web links - used in collections
rel	multiple types: see schema	No	Read Write	The relation of the target URI referenced by the link to the context URI
href	string	Yes	Read Write	This is the target URI, it can be specified as a Relative Reference or fully-qualified URI.
eps	array: see schema	No	Read Write	the Endpoint information of the target Resource
rt	array: see schema	Yes	Read Write	Resource Type of the Resource
p	object: see schema	No	Read Write	Specifies the framework policies on the Resource referenced by the target URI
title	string	No	Read Write	A title for the link relation. Can be used by the UI to provide a context.
di	multiple types: see schema	No	Read Write	
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.

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

				The Device ID formatted according to IETF RFC 4122.
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.
if	array: see schema	Yes	Read Write	The interface set supported by this resource
anchor	string	No	Read Write	This is used to override the context URI e.g. override the URI of the containing collection.
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	

2279 **B.2.6 CRUDN behaviour**

2280 **Table 18 The CRUDN operations of the resource**

Resource	Create	Read	Update	Delete	Notify
/EasySetupResURI		get			observe

2281 **B.3 Wi-Fi Configuration**

2282 **B.3.1 Introduction**

2283 WiFiConf resource stores essential information to help an unboxing device
2284 to connect to an existing Wi-Fi AP.

2285 **B.3.2 Example URI**

2286 /WiFiConfResURI

2287 **B.3.3 Resource Type**

2288 The resource type (rt) is defined as: ['oic.r.wificonf'].

2289 **B.3.4 Swagger2.0 Definition**

2290 {
2291 "swagger": "2.0",

```

2292     "info": {
2293         "title": "Wi-Fi Configuration Resource Baseline Interface",
2294         "version": "v0.0.3-20170611",
2295         "license": {
2296             "name": "copyright 2016-2017 Open Connectivity Foundation, Inc. All rights reserved.",
2297             "x-description": "Redistribution and use in source and binary forms, with or without
2298 modification, are permitted provided that the following conditions are met:\n      1.
2299 Redistributions of source code must retain the above copyright notice, this list of conditions and
2300 the following disclaimer.\n      2. Redistributions in binary form must reproduce the above
2301 copyright notice, this list of conditions and the following disclaimer in the documentation and/or
2302 other materials provided with the distribution.\n\n      THIS SOFTWARE IS PROVIDED BY THE Open
2303 Connectivity Foundation, INC. \AS IS\ AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
2304 LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR
2305 WARRANTIES OF NON-INFRINGEMENT, ARE DISCLAIMED.\n\n      IN NO EVENT SHALL THE Open Connectivity
2306 Foundation, INC. OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
2307 EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
2308 OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)\n\n      HOWEVER CAUSED AND
2309 ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
2310 OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY
2311 OF SUCH DAMAGE.\n"
2312         }
2313     },
2314     "schemes": ["http"],
2315     "consumes": ["application/json"],
2316     "produces": ["application/json"],
2317     "paths": {
2318         "/WiFiConfResURI?if=oic.if.baseline" : {
2319             "get": {
2320                 "description": "WiFiConf resource stores essential information to help an unboxing
2321 device\nto connect to an existing Wi-Fi AP.\nRetrieve properties of WiFiConf resource.\nThe
2322 information includes :\n1. Wi-Fi SSID and password\n2. Wi-Fi Security type (i.e. auth type and
2323 encryption type)\n3. Wi-Fi hardware capability (i.e. supported frequencies, modes,\n auth types
2324 and encryption types)\n",
2325                 "parameters": [
2326                     { "$ref": "#/parameters/interface-baseline" }
2327                 ],
2328                 "responses": {
2329                     "200": {
2330                         "description": "",
2331                         "x-example":
2332                         {
2333                             "rt": ["oic.r.wificonf"],
2334                             "swmt": ["A", "B", "G"],
2335                             "swf": ["2.4G", "5G"],
2336                             "tnn": "Home_AP_SSID",
2337                             "cd": "Home_AP_PWD",
2338                             "wat": "WPA2_PSK",
2339                             "wet": "TKIP",
2340                             "swat": ["WPA2_PSK", "WPA2_PSK"],
2341                             "swet": ["TKIP", "AES", "TKIP_AES"]
2342                         }
2343                     },
2344                     "schema": { "$ref": "#/definitions/WiFiConf" }
2345                 }
2346             },
2347         },
2348         "post": {
2349             "description": "Deliver Wi-Fi AP's information for an unboxing device to connect to it.\n",
2350             "parameters": [
2351                 { "$ref": "#/parameters/interface-baseline" },
2352                 {
2353                     "name": "body",
2354                     "in": "body",
2355                     "required": true,
2356                     "schema": { "$ref": "#/definitions/WiFiConfUpdate" },
2357                     "x-example":
2358                     {
2359                         "tnn": "Home_AP_SSID",
2360                         "cd": "Home_AP_PWD",
2361                         "wat": "WPA2_PSK",
2362                         "wet": "AES"

```

```

2363     }
2364   },
2365   "responses": {
2366     "200": {
2367       "description": "",
2368       "x-example":
2369         {
2370           "tnn": "Home_AP_SSID",
2371           "cd": "Home_AP_PWD",
2372           "wat": "WPA2_PSK",
2373           "wet": "AES"
2374         }
2375     },
2376     "schema": { "$ref": "#/definitions/WiFiConfUpdate" }
2377   }
2378 },
2379 },
2380 },
2381 },
2382 "/WiFiConfResURI?if=oic.if.rw" : {
2383   "get": {
2384     "description": "WiFiConf resource stores essential information to help an unboxing
2385 device\nto connect to an existing Wi-Fi AP.\nRetrieve properties of WiFiConf resource that can be
2386 updated by a client.\n",
2387     "parameters": [
2388       { "$ref": "#/parameters/interface-rw" }
2389     ],
2390     "responses": {
2391       "200": {
2392         "description": "",
2393         "x-example":
2394           {
2395             "tnn": "Home_AP_SSID",
2396             "cd": "Home_AP_PWD",
2397             "wat": "WPA2_PSK",
2398             "wet": "AES"
2399           }
2400       },
2401       "schema": { "$ref": "#/definitions/WiFiConfUpdate" }
2402     }
2403   }
2404 },
2405 "post": {
2406   "description": "Deliver Wi-Fi AP's information for an unboxing device to connect to it.\n",
2407   "parameters": [
2408     { "$ref": "#/parameters/interface-rw" },
2409     {
2410       "name": "body",
2411       "in": "body",
2412       "required": true,
2413       "schema": { "$ref": "#/definitions/WiFiConfUpdate" },
2414       "x-example":
2415         {
2416           "tnn": "Home_AP_SSID",
2417           "cd": "Home_AP_PWD",
2418           "wat": "WPA2_PSK",
2419           "wet": "AES"
2420         }
2421     }
2422   ],
2423   "responses": {
2424     "200": {
2425       "description": "",
2426       "x-example":
2427         {
2428           "tnn": "Home_AP_SSID",
2429           "cd": "Home_AP_PWD",
2430           "wat": "WPA2_PSK",
2431           "wet": "AES"
2432         }
2433     },

```

```

2434         "schema": { "$ref": "#/definitions/WiFiConfUpdate" }
2435     }
2436 }
2437 }
2438 }
2439 },
2440 "parameters": {
2441     "interface-rw" : {
2442         "in" : "query",
2443         "name" : "if",
2444         "type" : "string",
2445         "enum" : ["oic.if.rw"]
2446     },
2447     "interface-baseline" : {
2448         "in" : "query",
2449         "name" : "if",
2450         "type" : "string",
2451         "enum" : ["oic.if.baseline"]
2452     },
2453     "interface-all" : {
2454         "in" : "query",
2455         "name" : "if",
2456         "type" : "string",
2457         "enum" : ["oic.if.baseline", "oic.if.rw"]
2458     }
2459 },
2460 "definitions": {
2461     "WiFiConf" : {
2462         "properties": {
2463             "rt" :
2464                 {
2465                     "description": "Resource Type of the Resource",
2466                     "items": {
2467                         "maxLength": 64,
2468                         "type": "string"
2469                     },
2470                     "minItems": 1,
2471                     "readOnly": true,
2472                     "type": "array"
2473                 },
2474             "tnn" :
2475                 {
2476                     "description": "Indicates Target Network Name (SSID of Wi-Fi AP)",
2477                     "pattern": "^.*$",
2478                     "type": "string"
2479                 },
2480             "swmt" :
2481                 {
2482                     "description": "Indicates supported Wi-Fi mode types. It can be multiple",
2483                     "items": {
2484                         "description": "Supported Wi-Fi Mode Type.",
2485                         "enum": [
2486                             "A",
2487                             "B",
2488                             "G",
2489                             "N",
2490                             "AC"
2491                         ],
2492                         "type": "string"
2493                     },
2494                     "readOnly": true,
2495                     "type": "array"
2496                 },
2497             "wat" :
2498                 {
2499                     "description": "Indicates Wi-Fi Auth Type",
2500                     "enum": [
2501                         "None",

```

```

2505         "WEP",
2506         "WPA_PSK",
2507         "WPA2_PSK"
2508     ],
2509     "type": "string"
2510 },
2511
2512     "n" :
2513     {
2514         "description": "Friendly name of the resource",
2515         "maxLength": 64,
2516         "readOnly": true,
2517         "type": "string"
2518     },
2519
2520     "swat" :
2521     {
2522         "description": "Indicates supported Wi-Fi Auth types. It can be multiple",
2523         "items": {
2524             "description": "Indicates Wi-Fi Auth Type",
2525             "enum": [
2526                 "None",
2527                 "WEP",
2528                 "WPA_PSK",
2529                 "WPA2_PSK"
2530             ],
2531             "type": "string"
2532         },
2533         "readOnly": true,
2534         "type": "array"
2535     },
2536
2537     "swf" :
2538     {
2539         "description": "Indicates Supported Wi-Fi frequencies by the Enrollee. Can be multiple.
2540 Valid values are ('2.4G', '5G')",
2541         "items": {
2542             "pattern": "^(2\\.4|5)G$",
2543             "type": "string"
2544         },
2545         "readOnly": true,
2546         "type": "array"
2547     },
2548
2549     "swet" :
2550     {
2551         "description": "Indicates supported Wi-Fi Encryption types. It can be multiple",
2552         "items": {
2553             "description": "Indicates Wi-Fi Encryption Type",
2554             "enum": [
2555                 "None",
2556                 "WEP_64",
2557                 "WEP_128",
2558                 "TKIP",
2559                 "AES",
2560                 "TKIP_AES"
2561             ],
2562             "type": "string"
2563         },
2564         "readOnly": true,
2565         "type": "array"
2566     },
2567
2568     "wet" :
2569     {
2570         "description": "Indicates Wi-Fi Encryption Type",
2571         "enum": [
2572             "None",
2573             "WEP_64",
2574             "WEP_128",
2575             "TKIP",

```

```

2576         "AES",
2577         "TKIP_AES"
2578     ],
2579     "type": "string"
2580 },
2581
2582     "cd" :
2583     {
2584         "description": "Indicates credential information of Wi-Fi AP",
2585         "pattern": "^.*$",
2586         "type": "string"
2587     },
2588
2589     "id" :
2590     {
2591         "description": "Instance ID of this specific resource",
2592         "maxLength": 64,
2593         "readOnly": true,
2594         "type": "string"
2595     },
2596
2597     "if" :
2598     {
2599         "description": "The interface set supported by this resource",
2600         "items": {
2601             "enum": [
2602                 "oic.if.baseline",
2603                 "oic.if.ll",
2604                 "oic.if.b",
2605                 "oic.if.lb",
2606                 "oic.if.rw",
2607                 "oic.if.r",
2608                 "oic.if.a",
2609                 "oic.if.s"
2610             ],
2611             "type": "string"
2612         },
2613         "minItems": 1,
2614         "readOnly": true,
2615         "type": "array"
2616     }
2617
2618     }, "type" : "object"
2619 }
2620
2621 ,
2622 "WiFiConfUpdate" : {
2623     "properties": {
2624         "rt" :
2625         {
2626             "description": "Resource Type of the Resource",
2627             "items": {
2628                 "maxLength": 64,
2629                 "type": "string"
2630             },
2631             "minItems": 1,
2632             "readOnly": true,
2633             "type": "array"
2634         },
2635
2636         "wat" :
2637         {
2638             "description": "Indicates Wi-Fi Auth Type",
2639             "enum": [
2640                 "None",
2641                 "WEP",
2642                 "WPA_PSK",
2643                 "WPA2_PSK"
2644             ]
2645         },
2646

```

```

2647     "n" :
2648         {
2649             "description": "Friendly name of the resource",
2650             "maxLength": 64,
2651             "readOnly": true,
2652             "type": "string"
2653         },
2654
2655     "cd" :
2656         {
2657             "description": "Indicates credential information of Wi-Fi AP",
2658             "pattern": "^.*$",
2659             "type": "string"
2660         },
2661
2662     "wet" :
2663         {
2664             "description": "Indicates Wi-Fi Encryption Type",
2665             "enum": [
2666                 "None",
2667                 "WEP_64",
2668                 "WEP_128",
2669                 "TKIP",
2670                 "AES",
2671                 "TKIP_AES"
2672             ]
2673         },
2674
2675     "tnn" :
2676         {
2677             "description": "Indicates Target Network Name (SSID of Wi-Fi AP)",
2678             "pattern": "^.*$",
2679             "type": "string"
2680         },
2681
2682     "id" :
2683         {
2684             "description": "Instance ID of this specific resource",
2685             "maxLength": 64,
2686             "readOnly": true,
2687             "type": "string"
2688         },
2689
2690     "if" :
2691         {
2692             "description": "The interface set supported by this resource",
2693             "items": {
2694                 "enum": [
2695                     "oic.if.baseline",
2696                     "oic.if.ll",
2697                     "oic.if.b",
2698                     "oic.if.lb",
2699                     "oic.if.rw",
2700                     "oic.if.r",
2701                     "oic.if.a",
2702                     "oic.if.s"
2703                 ],
2704                 "type": "string"
2705             },
2706             "minItems": 1,
2707             "readOnly": true,
2708             "type": "array"
2709         }
2710     },
2711     "type" : "object"
2712 }
2713 }
2714 }
2715 }
2716 }

```

Table 19 The properties definitions of the resource

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

n	string		Read Only	Friendly name of the resource
cd	string		Read Write	Indicates credential information of Wi-Fi AP
if	array: see schema		Read Only	The interface set supported by this resource
tnn	string		Read Write	Indicates Target Network Name (SSID of Wi-Fi AP)
wat	multiple types: see schema		Read Write	Indicates Wi-Fi Auth Type
wet	multiple types: see schema		Read Write	Indicates Wi-Fi Encryption Type
rt	array: see schema		Read Only	Resource Type of the Resource
id	string		Read Only	Instance ID of this specific resource
n	string		Read Only	Friendly name of the resource

2719

B.3.6 CRUDN behaviour

2720

Table 20 The CRUDN operations of the resource

Resource	Create	Read	Update	Delete	Notify
/WiFiConfResURI		get	post		observe

2721