

OCF Core Specification Extension

WiFi Easy Setup

VERSION 1.3.1 | March 2018



OPEN CONNECTIVITY
FOUNDATION™

CONTACT admin@openconnectivity.org

Copyright Open Connectivity Foundation, Inc. © 2017.
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
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61

1	Scope	7
2	Normative references	7
3	Terms, definitions, symbols and abbreviations	8
3.1	Terms and definitions	8
3.2	Conventions	8
3.3	Data types	8
4	Document conventions and organization	9
5	Overview	10
5.1	Introduction	10
5.2	Architecture	10
5.3	Example Scenario	10
6	Resource model	11
6.1	Introduction	11
6.2	EasySetup Resource	11
6.2.1	Overview	11
6.2.2	Resource	11
6.3	WiFiConf Resource Type	12
6.3.1	Introduction	12
6.3.2	Resource Type	12
6.4	DevConf Resource Type	13
6.4.1	Introduction	13
6.4.2	Resource Type	14
7	Network and connectivity	15
8	Functional interactions	16
8.1	Onboarding, Provisioning and Configuration	16
8.2	Resource discovery	16
8.3	Retrieving and Updating Easy Setup Resources	16
8.4	Error Handling	16
8.5	Example Easy Setup Flow	17
9	Security	20
Annex A (normative)	Resource Type definitions	21
A.1	List of Resource Type definitions	21
A.2	Easy Setup Collection Baseline Interface	21
A.2.1	Introduction	21
A.2.2	Example URI	21
A.2.3	Resource Type	21
A.2.4	RAML Definition	21
A.2.5	Property Definition	23
A.2.6	CRUDN behaviour	24
A.3	Wi-Fi Configuration Resource Baseline Interface	25

62	A.3.1	Introduction	25
63	A.3.2	Example URI	25
64	A.3.3	Resource Type	25
65	A.3.4	RAML Definition	25
66	A.3.5	Property Definition	29
67	A.3.6	CRUDN behaviour	29
68	A.4	Device Configuration	29
69	A.4.1	Introduction	29
70	A.4.2	Example URI	29
71	A.4.3	Resource Type	29
72	A.4.4	RAML Definition	29
73	A.4.5	Property Definition	31
74	A.4.6	CRUDN behaviour	31
75	Annex B (informative)	Swagger2.0 definitions	32
76	B.1	Device Configuration	32
77	B.1.1	Introduction	32
78	B.1.2	Example URI	32
79	B.1.3	Resource Type	32
80	B.1.4	Swagger2.0 Definition	32
81	B.1.5	Property Definition	34
82	B.1.6	CRUDN behaviour	34
83	B.2	EasySetup Collection	34
84	B.2.1	Introduction	34
85	B.2.2	Example URI	34
86	B.2.3	Resource Type	34
87	B.2.4	Swagger2.0 Definition	34
88	B.2.5	Property Definition	45
89	B.2.6	CRUDN behaviour	48
90	B.3	Wi-Fi Configuration Resource	48
91	B.3.1	Introduction	48
92	B.3.2	Example URI	48
93	B.3.3	Resource Type	48
94	B.3.4	Swagger2.0 Definition	48
95	B.3.5	Property Definition	54
96	B.3.6	CRUDN behaviour	55
97			
98			

99
100
101
102
103
104
105

Figures

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

Tables

106
107
108
109
110
111
112
113
114
115
116

Table 1. EasySetup Resource Type.....	11
Table 2. “oic.r.easyssetup” Resource Type definition	11
Table 3. WiFiConf Resource Type	13
Table 4. “oic.r.wificonf” Resource Type definition	13
Table 5. DevConf Resource Type	14
Table 6. “oic.r.devconf” Resource Type definition	14
Table 7. Alphabetized list of resources	21

117 **1 Scope**

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

122 **2 Normative references**

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

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

127 Available at: https://openconnectivity.org/specs/OCF_Core_Specification_v1.3.1.pdf

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

129 OCF Security Specification, *Open Connectivity Foundation Security Capabilities*, Version 1.3.1

130 Available at: https://openconnectivity.org/specs/OCF_Security_Specification_v1.3.1.pdf

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

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

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

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

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

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

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

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

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

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

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

145

146 **3 Terms, definitions, symbols and abbreviations**

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

148 **3.1 Terms and definitions**

149 As defined in the OCF Core Specification with the following additions

150 **3.1.1.**

151 **Easy Setup Enrollment**

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

154 **3.1.2.**

155 **Enrollee**

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

157 **3.1.3.**

158 **Enroller**

159 Target network entity to which the Enrollee connects. E.g. Wi-Fi Access Point

160 **3.1.4.**

161 **Mediator**

162 Device (for example a Mobile Phone) that enables the Enrollee to connect to the target network
163 (Enroller). The Mediator transfers configuration information to the Enrollee.

164 **3.1.5.**

165 **Easy Setup**

166 Process of configuring an Enrollee to an Enroller using a Mediator (by the transferring of essential
167 information about the Enroller to the Enrollee).

168 **3.1.6.**

169 **Soft AP**

170 Software Enabled Access Point hosted on the Enrollee which is not a dedicated Access Point.

171 **3.2 Conventions**

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

176 **3.3 Data types**

177 As defined in the OCF Core Specification.

178

179 **4 Document conventions and organization**

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

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

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

186 Recommended (or should)(S).

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

193 Allowed (may or allowed)(O).

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

197 DEPRECATED.

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

204 Conditionally allowed (CA)

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

207 Conditionally required (CR)

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

211

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

213 Words that are emphasized are printed in italic.

214

215 **5 Overview**

216 **5.1 Introduction**

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

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

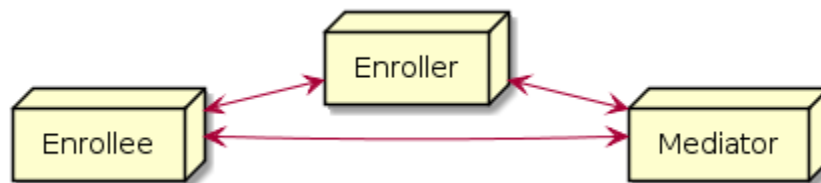
221 Specifically, this method allows the transferring of essential information to the new Device, which
222 includes:

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

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

227 **5.2 Architecture**

228 Figure 1 shows the deployment architectural approach.
229



230

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

232 Easy Setup defines the following roles: Enrollee, Enroller, and Mediator. Please refer to Section
233 3.1 for definitions thereof.

234 **5.3 Example Scenario**

235 The following scenario presents a typical setup case.

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

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

244

245 **6 Resource model**

246 **6.1 Introduction**

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

- 248 1. EasySetup Resource Type
- 249 2. WiFiConf Resource Type
- 250 3. DevConf Resource Type

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

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



255
256 **Figure 2. Easy Setup Resource Types**

257 **6.2 EasySetup Resource**

258 **6.2.1 Overview**

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

261 **6.2.2 Resource**

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

263 **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.	

264

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

267 **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.

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

- 269 • “ps” equal to 0.
- 270 • “lec” equal to 0.
- 271 • “cn” equal to an empty array.

272 **6.3 WiFiConf Resource Type**

273 **6.3.1 Introduction**

274 The WiFiConf Resource Type stores information to help an Enrollee to connect to an existing Wi-Fi AP.
275

276 **6.3.2 Resource Type**

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

Table 3. WiFiConf Resource Type

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

279

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

282

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. enrollee)
Credential	cd	string			RW	No	Credential information of Wi-Fi AP (Password used to connect to enrollee).
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")

283

284 6.4 DevConf Resource Type

285 6.4.1 Introduction

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

287 **6.4.2 Resource Type**

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

289 **Table 5. DevConf Resource Type**

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

290

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

293 **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.

294

295 **7 Network and connectivity**

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

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

299

300 **8 Functional interactions**

301 **8.1 Onboarding, Provisioning and Configuration**

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

305

306 **8.2 Resource discovery**

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

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

311 **8.3 Retrieving and Updating Easy Setup Resources**

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

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

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

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

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

325 **8.4 Error Handling**

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

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

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

342 **8.5 Example Easy Setup Flow**

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

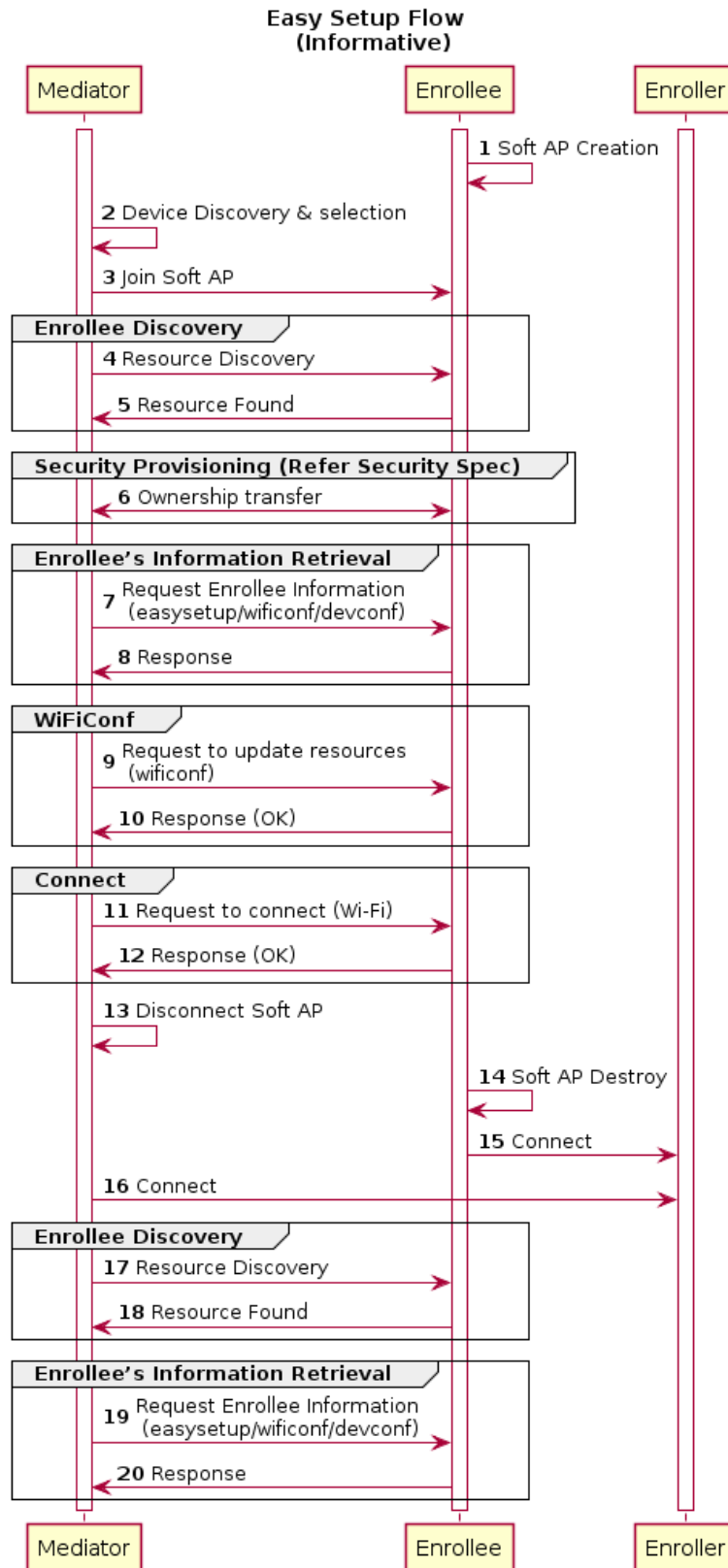


Figure 3. Easy Setup Flow (Informative)

346
347
348
349
350
351

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

352 **9 Security**

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

Annex A (normative)

Resource Type definitions

A.1 List of Resource Type definitions

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

Table 7. Alphabetized list of resources

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

A.2 Easy Setup Collection Baseline Interface

A.2.1 Introduction

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

A.2.2 Example URI

/exampleEasySetupBaselineInterfaceResURI

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"]
  - interface-batch :
    queryParameters:
```

```

390     if:
391         enum: ["oic.if.b"]
392
393 /exampleEasySetupBaselineInterfaceResURI:
394     description: |
395         Easy Setup resource stores useful information including current status of
396         unboxing device and last error code which are produced in a process of
397         easy setup.
398         Note that, Easy Setup resource is a type of collection resource, which
399         contains links to WiFiConf, DevConf resources and may additionally contain
400         links to other resources.
401
402     is : ['interface-baseline']
403     get:
404         description: |
405             Retrieve useful information during easy setup process :
406             1
407             A current status in easy setup process.
408             2
409             A last error code describing reason for failure occurred at the last
410             time.
411
412     responses :
413         200:
414             body:
415                 application/json:
416                     schema: /
417                         {
418                             "$schema": "http://json-schema.org/draft-04/schema#",
419                             "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
420 reserved.",
421                             "id": "http://www.openconnectivity.org/ocf-apis/core-
422 extensions/schemas/oic.r.easyssetup-schema.json#",
423                             "definitions": {
424                                 "oic.r.easyssetup": {
425                                     "type": "object",
426                                     "properties": {
427                                         "rt": {
428                                             "type": "array",
429                                             "minItems": 2,
430                                             "maxItems": 2,
431                                             "uniqueItems": true,
432                                             "items": {
433                                                 "enum": ["oic.r.easyssetup", "oic.wk.col"]
434                                             }
435                                         },
436                                         "ps": {
437                                             "type": "integer",
438                                             "enum": [0, 1, 2, 3],
439                                             "description": "Indicates the easy setup status of the device. (0: Need to
440 Setup, 1: Connecting to Enroller, 2: Connected to Enroller, 3: Failed to Connect to Enroller,
441 4~254: Reserved, 255: EOF)",
442                                             "readOnly": true
443                                         },
444                                         "lec": {
445                                             "type": "integer",
446                                             "enum": [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 255],
447                                             "description": "Indicates a failure reason (0: NO error, 1: A given SSID is
448 not found, 2: Wi-Fi's password is wrong, 3: IP address is not allocated, 4: No internet connection,
449 5: Timeout, 6: Wi-Fi Auth Type is not supported by the Enrollee, 7: Wi-Fi Encryption Type is not
450 supported by the Enrollee, 8: Wi-Fi Auth Type is wrong (failure while connecting to the Enroller),
451 9: Wi-Fi Encryption Type is wrong (failure while connecting to the Enroller), 10~254: Reserved,
452 255: Unknown error)",
453                                             "readOnly": true
454                                         },

```

```

455         "cn": {
456             "type": "array",
457             "description": "Indicates an array of connection types that trigger an
458 attempt to connect to the Enroller to start.",
459             "items": {
460                 "type": "integer",
461                 "description": "Connection type to attempt. (1 : Wi-Fi, 2 : other
462 entities / transports to be added in future (e.g. Connect to cloud / BLE))"
463             }
464         },
465     },
466     "required": ["ps", "lec", "cn"]
467 },
468 },
469 "type": "object",
470 "allOf": [
471     { "$ref": "../../core/schemas/oic.core-schema.json#/definitions/oic.core" },
472     { "$ref": "../../core/schemas/oic.collection-
473 schema.json#/definitions/oic.collection.properties" },
474     { "$ref": "../../core/schemas/oic.collection-
475 schema.json#/definitions/oic.collection.links.arrayoflinks" },
476     { "$ref": "#/definitions/oic.r.easyssetup" }
477 ]
478 }
479
480 example: /
481 {
482     "rt" : ["oic.r.easyssetup", "oic.wk.col"],
483     "if" : ["oic.if.ll", "oic.if.baseline", "oic.if.b"],
484     "ps" : 0,
485     "lec": 0,
486     "cn": [1],
487     "links": [
488         {
489             "href": "/EasySetupResURI",
490             "rt": ["oic.r.easyssetup", "oic.wk.col"],
491             "if": ["oic.if.b"],
492             "p":{"bm":3},
493             "eps": [
494                 {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2}
495             ],
496             "rel":["self", "item"]
497         },
498         {
499             "href": "/WiFiConfResURI",
500             "rt": ["oic.r.wificonf"],
501             "if": ["oic.if.baseline"],
502             "p":{"bm":3},
503             "eps": [
504                 {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2}
505             ]
506         },
507         {
508             "href": "/DevConfResURI",
509             "rt": ["oic.r.devconf"],
510             "if": ["oic.if.baseline"],
511             "p":{"bm":3},
512             "eps": [
513                 {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2}
514             ]
515         }
516     ]
517 }
518
519

```

A.2.5 Property Definition

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: schema	see	yes	Indicates an array of connection types that trigger an attempt to connect to the Enroller to start.

520

A.2.6 CRUDN behaviour

Resource	Create	Read	Update	Delete	Notify
/exampleEasySetupBaselineInterfaceResURI		get			

521 A.3 Wi-Fi Configuration Resource Baseline Interface

522 A.3.1 Introduction

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

525 A.3.2 Example URI

526 /example/WiFiConfBaselineInterfaceResURI

527 A.3.3 Resource Type

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

529 A.3.4 RAML Definition

```
530 #%RAML 0.8
531 title: Wi-Fi Configuration Resource
532 version: v0.0.3-20170611
533 traits:
534   - interface-rw :
535     queryParameters:
536       if:
537         enum: ["oic.if.rw"]
538   - interface-baseline :
539     queryParameters:
540       if:
541         enum: ["oic.if.baseline"]
542   - interface-all :
543     queryParameters:
544       if:
545         enum: ["oic.if.baseline", "oic.if.rw"]
546
547 /example/WiFiConfBaselineInterfaceResURI:
548   description: |
549     WiFiConf resource stores essential information to help an unboxing device
550     to connect to an existing Wi-Fi AP.
551
552   is : ['interface-baseline']
553   get:
554     description: |
555       Retrieve properties of WiFiConf resource.
556       The information includes :
557       1
558       Wi-Fi SSID and password
559       2
560       Wi-Fi Security type (i.e
561       auth type and encryption type)
562       3
563       Wi-Fi hardware capability (i.e
564       supported frequencies, modes,
565       auth types and encryption types)
566
567   responses :
568     200:
569       body:
570         application/json:
571           schema: /
572             {
573               "$schema": "http://json-schema.org/draft-04/schema#",
```

```

574         "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
575 reserved.",
576         "id": "http://www.openconnectivity.org/ocf-apis/core-
577 extensions/schemas/oic.r.wificonf-schema.json#",
578         "definitions": {
579             "oic.r.wificonf": {
580                 "type": "object",
581                 "properties": {
582                     "swmt": {
583                         "type": "array",
584                         "description": "Indicates supported Wi-Fi mode types. It can be multiple",
585                         "readOnly": true,
586                         "items":
587                         {
588                             "type": "string",
589                             "enum": ["A","B","G","N","AC"],
590                             "description": "Supported Wi-Fi Mode Type."
591                         }
592                     },
593                     "swf": {
594                         "type": "array",
595                         "description": "Indicates Supported Wi-Fi frequencies by the Enrollee. Can
596 be multiple. Valid values are ('2.4G', '5G')",
597                         "readOnly": true,
598                         "items":
599                         {
600                             "type": "string",
601                             "pattern": "^(2\\.4|5)G$"
602                         }
603                     },
604                     "tnn": {
605                         "type": "string",
606                         "description": "Indicates Target Network Name (SSID of Wi-Fi AP)",
607                         "pattern": "^.*$"
608                     },
609                     "cd": {
610                         "type": "string",
611                         "description": "Indicates credential information of Wi-Fi AP",
612                         "pattern": "^.*$"
613                     },
614                     "wat": {
615                         "type": "string",
616                         "enum": ["None", "WEP", "WPA_PSK", "WPA2_PSK"],
617                         "description": "Indicates Wi-Fi Auth Type"
618                     },
619                     "wet": {
620                         "type": "string",
621                         "enum": ["None", "WEP_64", "WEP_128", "TKIP", "AES", "TKIP_AES"],
622                         "description": "Indicates Wi-Fi Encryption Type"
623                     },
624                     "swat": {
625                         "type": "array",
626                         "description": "Indicates supported Wi-Fi Auth types. It can be multiple",
627                         "readOnly": true,
628                         "items":
629                         {
630                             "type": "string",
631                             "enum": ["None", "WEP", "WPA_PSK", "WPA2_PSK"],
632                             "description": "Indicates Wi-Fi Auth Type"
633                         }
634                     },
635                     "swet": {
636                         "type": "array",
637                         "description": "Indicates supported Wi-Fi Encryption types. It can be
638 multiple",
639                         "readOnly": true,
640                         "items":
641                         {
642                             "type": "string",
643                             "enum": ["None", "WEP_64", "WEP_128", "TKIP", "AES", "TKIP_AES"],
644                             "description": "Indicates Wi-Fi Encryption Type"

```

```

645         }
646     },
647     },
648     "required":["swmt", "swf", "swat", "swet", "tnn", "wat", "wet"]
649 },
650 },
651 "type": "object",
652 "allOf": [
653   { "$ref": "../core/schemas/oic.core-schema.json#/definitions/oic.core"},
654   { "$ref": "#/definitions/oic.r.wificonf" }
655 ]
656 }
657
658 example: /
659 {
660   "rt": ["oic.r.wificonf"],
661   "swmt" : ["A", "B", "G"],
662   "swf": ["2.4G", "5G"],
663   "tnn": "Home_AP_SSID",
664   "cd": "Home_AP_PWD",
665   "wat": "WPA2_PSK",
666   "wet": "TKIP",
667   "swat": ["WPA_PSK", "WPA2_PSK"],
668   "swet": ["TKIP", "AES", "TKIP_AES"]
669 }
670
671 post:
672   description: |
673     Deliver Wi-Fi AP's information for an unboxing device to connect to it.
674
675   body:
676     application/json:
677       schema: /
678         {
679           "$schema": "http://json-schema.org/draft-v4/schema#",
680           "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
681 reserved.",
682           "id": "http://www.openconnectivity.org/ocf-apis/core-extensions/schemas/oic.r.wificonf-
683 update-schema.json#",
684           "definitions": {
685             "oic.r.wificonf": {
686               "type": "object",
687               "properties": {
688                 "tnn": {
689                   "type": "string",
690                   "description": "Indicates Target Network Name (SSID of Wi-Fi AP)",
691                   "pattern": "^.*$"
692                 },
693                 "cd": {
694                   "type": "string",
695                   "description": "Indicates credential information of Wi-Fi AP",
696                   "pattern": "^.*$"
697                 },
698                 "wat": {
699                   "enum": ["None", "WEP", "WPA_PSK", "WPA2_PSK"],
700                   "description": "Indicates Wi-Fi Auth Type"
701                 },
702                 "wet": {
703                   "enum": ["None", "WEP_64", "WEP_128", "TKIP", "AES", "TKIP_AES"],
704                   "description": "Indicates Wi-Fi Encryption Type"
705                 }
706               },
707               "required":["tnn", "wat", "wet"]
708             }
709           },
710           "type": "object",
711           "allOf": [

```

```

712         { "$ref": "../../../core/schemas/oic.core-schema.json#/definitions/oic.core"},
713         { "$ref": "#/definitions/oic.r.wificonf" }
714     ]
715 }
716
717 example: /
718 {
719     "tnn": "Home_AP_SSID",
720     "cd": "Home_AP_PWD",
721     "wat": "WPA2_PSK",
722     "wet": "AES"
723 }
724
725 responses :
726 200:
727   body:
728     application/json:
729     schema: /
730     {
731         "$schema": "http://json-schema.org/draft-v4/schema#",
732         "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
733 reserved.",
734         "id": "http://www.openconnectivity.org/ocf-apis/core-
735 extensions/schemas/oic.r.wificonf-update-schema.json#",
736         "definitions": {
737             "oic.r.wificonf": {
738                 "type": "object",
739                 "properties": {
740                     "tnn": {
741                         "type": "string",
742                         "description": "Indicates Target Network Name (SSID of Wi-Fi AP)",
743                         "pattern": "^.*$"
744                     },
745                     "cd": {
746                         "type": "string",
747                         "description": "Indicates credential information of Wi-Fi AP",
748                         "pattern": "^.*$"
749                     },
750                     "wat": {
751                         "enum": ["None", "WEP", "WPA_PSK", "WPA2_PSK"],
752                         "description": "Indicates Wi-Fi Auth Type"
753                     },
754                     "wet": {
755                         "enum": ["None", "WEP_64", "WEP_128", "TKIP", "AES", "TKIP_AES"],
756                         "description": "Indicates Wi-Fi Encryption Type"
757                     }
758                 },
759                 "required":["tnn", "wat", "wet"]
760             }
761         },
762         "type": "object",
763         "allOf": [
764             { "$ref": "../../../core/schemas/oic.core-schema.json#/definitions/oic.core"},
765             { "$ref": "#/definitions/oic.r.wificonf" }
766         ]
767     }
768
769 example: /
770 {
771     "tnn": "Home_AP_SSID",
772     "cd": "Home_AP_PWD",
773     "wat": "WPA2_PSK",
774     "wet": "AES"
775 }
776

```

777 **A.3.5 Property Definition**

Property name	Value type	Mandatory	Access mode	Description
tnn	string	yes		Indicates Target Network Name (SSID of Wi-Fi AP)
swmt	array: schema see	yes	Read Only	Indicates supported Wi-Fi mode types. It can be multiple
swat	array: schema see	yes	Read Only	Indicates supported Wi-Fi Auth types. It can be multiple
cd	string			Indicates credential information of Wi-Fi AP
swf	array: schema see	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: schema see	yes	Read Only	Indicates supported Wi-Fi Encryption types. It can be multiple

778 **A.3.6 CRUDN behaviour**

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

779 **A.4 Device Configuration**

780 **A.4.1 Introduction**

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

783 **A.4.2 Example URI**

784 /example/DevConfResURI

785 **A.4.3 Resource Type**

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

787 **A.4.4 RAML Definition**

788 `##RAML 0.8`

789 `title: Device Configuration Resource`

790 `version: v0.0.2-20170604`

791 `traits:`

```

792 - interface :
793   queryParameters:
794     if:
795       enum: ["oic.if.baseline", "oic.if.r"]
796
797 /example/DevConfResURI:
798   description: |
799     Device configuration resource stores a preference of device settings like
800     device name
801     Vender-specific information can be added to the resource.
802
803   is : ['interface']
804   get:
805     description: |
806       Retrieve various settings regarding to device-specific settings
807       1
808       Device name (human-friendly name to be detected by mediator during
809       easy setup)
810
811   responses :
812     200:
813       body:
814         application/json:
815           schema: /
816             {
817               "$schema": "http://json-schema.org/draft-04/schema#",
818               "description" : "Copyright (c) 2017 Open Connectivity Foundation, Inc. All rights
819 reserved.",
820               "id": "http://www.openconnectivity.org/ocf-apis/core-
821 extensions/schemas/oic.r.devconf-schema.json#",
822               "definitions": {
823                 "oic.r.devconf": {
824                   "type": "object",
825                   "oneOf": [
826                     {
827                       "properties": {
828                         "dn": {
829                           "type": "string",
830                           "description": "Indicates a pre-configured device name in language
831 indicated by 'dl' in /oic/con; presented by enrollee device to mediator device during easy-setup
832 process",
833                           "pattern": "^.*$",
834                           "readOnly": true
835                         }
836                       },
837                       "required":["dn"]
838                     },
839                     {
840                       "properties": {
841                         "dn": {
842                           "type": "array",
843                           "items": {
844                             "type": "object",
845                             "properties": {
846                               "language": {
847                                 "$ref": "../..core/schemas/oic.types-
848 schema.json#/definitions/language-tag",
849                                 "readOnly": true,
850                                 "description": "An RFC 5646 language tag."
851                               },
852                               "value": {
853                                 "type": "string",
854                                 "description": "Pre-configured device name in the indicated
855 language.",

```

```

856         "pattern": "^.*$",
857         "readOnly": true
858     }
859 }
860 },
861 "minItems" : 1,
862 "readOnly": true,
863 "description": "Localized device name."
864 }
865 },
866 "required": ["dn"]
867 }
868 ]
869 }
870 },
871 "type": "object",
872 "allOf": [
873   { "$ref": "../../core/schemas/oic.core-schema.json#/definitions/oic.core" },
874   { "$ref": "#/definitions/oic.r.devconf" }
875 ]
876 }
877
878 example: /
879 {
880   "rt": ["oic.r.devconf"],
881   "dn" : "My Refrigerator"
882 }
883

```

884 A.4.5 Property Definition

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.

885 A.4.6 CRUDN behaviour

Resource	Create	Read	Update	Delete	Notify
/example/DevConfResURI		get			

886

887

Annex B (informative)

Swagger2.0 definitions

888
889
890
891

892 B.1 Device Configuration

893 B.1.1 Introduction

894 Device configuration resource stores a preference of device settings like
895 device name. Vender-specific information can be added to the resource.
896 Retrieve various settings regarding to device-specific settings
897 1. Device name (human-friendly name to be detected by mediator during
898 easy setup)
899

900 B.1.2 Example URI

901 /example/DevConfResURI

902 B.1.3 Resource Type

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

904 B.1.4 Swagger2.0 Definition

```
905 {  
906   "swagger": "2.0",  
907   "info": {  
908     "title": "Device Configuration",  
909     "version": "v0.0.2-20170604",  
910     "license": {  
911       "name": "copyright 2016-2017 Open Connectivity Foundation, Inc. All rights reserved.",  
912       "x-description": "Redistribution and use in source and binary forms, with or without  
913 modification, are permitted provided that the following conditions are met:\n      1.  
914 Redistributions of source code must retain the above copyright notice, this list of conditions and  
915 the following disclaimer.\n      2. Redistributions in binary form must reproduce the above  
916 copyright notice, this list of conditions and the following disclaimer in the documentation and/or  
917 other materials provided with the distribution.\n\n      THIS SOFTWARE IS PROVIDED BY THE Open  
918 Connectivity Foundation, INC. \\\nAS IS\\n AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT  
919 LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR  
920 WARRANTIES OF NON-INFRINGEMENT, ARE DISCLAIMED.\n\n      IN NO EVENT SHALL THE Open Connectivity  
921 Foundation, INC. OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,  
922 EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS  
923 OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)\n\n      HOWEVER CAUSED AND  
924 ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR  
925 OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY  
926 OF SUCH DAMAGE.\n\n    }  
927   },  
928   "schemes": ["http"],  
929   "consumes": ["application/json"],  
930   "produces": ["application/json"],  
931   "paths": {  
932     "/example/DevConfResURI" : {  
933       "get": {  
934         "description": "Device configuration resource stores a preference of device settings  
935 like\ndevice name. Vender-specific information can be added to the resource.\nRetrieve various  
936 settings regarding to device-specific settings\n1. Device name (human-friendly name to be detected  
937 by mediator during\n  easy setup)\n",  
938         "parameters": [  
939           ],  
940         "responses": {  
941           "200": {  
942             "description": "",  
943             "x-example":  
944               {  
945
```



```

946         "rt": ["oic.r.devconf"],
947         "dn" : "My Refrigerator"
948     }
949     ,
950     "schema": { "$ref": "#/definitions/DevConf" }
951 }
952 }
953 }
954 }
955 },
956 "parameters": {
957     "interface" : {
958         "in" : "query",
959         "name" : "if",
960         "type" : "string",
961         "enum" : ["oic.if.baseline", "oic.if.r"]
962     }
963 },
964 "definitions": {
965     "DevConf" : {
966         "properties": {
967             "rt" :
968                 {
969                     "description": "Resource Type of the Resource",
970                     "items": {
971                         "maxLength": 64,
972                         "type": "string"
973                     },
974                     "minItems": 1,
975                     "readOnly": true,
976                     "type": "array"
977                 },
978             "n" :
979                 {
980                     "description": "Friendly name of the resource",
981                     "maxLength": 64,
982                     "readOnly": true,
983                     "type": "string"
984                 },
985             "id" :
986                 {
987                     "description": "Instance ID of this specific resource",
988                     "maxLength": 64,
989                     "readOnly": true,
990                     "type": "string"
991                 },
992             "if" :
993                 {
994                     "description": "The interface set supported by this resource",
995                     "items": {
996                         "enum": [
1000                 "oic.if.baseline",
1001                 "oic.if.ll",
1002                 "oic.if.b",
1003                 "oic.if.lb",
1004                 "oic.if.rw",
1005                 "oic.if.r",
1006                 "oic.if.a",
1007                 "oic.if.s"
1008             ],
1009                     "type": "string"
1010                 },
1011             "minItems": 1,
1012             "readOnly": true,
1013             "type": "array"
1014         }
1015     }
1016 }

```

1017 }
 1018 }
 1019 }
 1020 }

1021 **B.1.5 Property Definition**

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

1022 **B.1.6 CRUDN behaviour**

Resource	Create	Read	Update	Delete	Notify
/example/DevConfResURI		get			

1023 **B.2 EasySetup Collection**

1024 **B.2.1 Introduction**

1025 Easy Setup resource stores useful information including current status of
 1026 unboxing device and last error code which are produced in a process of
 1027 easy setup.
 1028 Note that, Easy Setup resource is a type of collection resource, which
 1029 contains links to WiFiConf, DevConf resources and may additionally contain
 1030 links to other resources.
 1031

1032 **B.2.2 Example URI**

1033 /exampleEasySetupBaselineInterfaceResURI

1034 **B.2.3 Resource Type**

1035 The resource type (rt) is defined as: ['oic.r.easyssetup', 'oic.wk.col'].

1036 **B.2.4 Swagger2.0 Definition**

```

1037 {
1038   "swagger": "2.0",
1039   "info": {
1040     "title": "Easy Setup Collection Baseline Interface",
1041     "version": "v0.0.3-20170611",
1042     "license": {
1043       "name": "copyright 2016-2017 Open Connectivity Foundation, Inc. All rights reserved.",
1044       "x-description": "Redistribution and use in source and binary forms, with or without
1045 modification, are permitted provided that the following conditions are met:\n      1.
1046 Redistributions of source code must retain the above copyright notice, this list of conditions and
1047 the following disclaimer.\n      2. Redistributions in binary form must reproduce the above
1048 copyright notice, this list of conditions and the following disclaimer in the documentation and/or
1049 other materials provided with the distribution.\n\n      THIS SOFTWARE IS PROVIDED BY THE Open
1050 Connectivity Foundation, INC. \ "AS IS\ " AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
1051 LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR
1052 WARRANTIES OF NON-INFRINGEMENT, ARE DISCLAIMED.\n\n      IN NO EVENT SHALL THE Open Connectivity
1053 Foundation, INC. OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
1054 EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
1055 OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)\n\n      HOWEVER CAUSED AND
1056 ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
1057 OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY
  
```

```

1058 OF SUCH DAMAGE.\n"
1059 }
1060 },
1061 "schemes": ["http"],
1062 "consumes": ["application/json"],
1063 "produces": ["application/json"],
1064 "paths": {
1065   "/exampleEasySetupBaselineInterfaceResURI" : {
1066     "get": {
1067       "description": "Easy Setup resource stores useful information including current status
1068 of\nunboxing device and last error code which are produced in a process of\neasy setup.\nNote that,
1069 Easy Setup resource is a type of collection resource, which\ncontains links to WiFiConf, DevConf
1070 resources and may additionally contain\nlinks to other resources.\nRetrieve useful information
1071 during easy setup process :\n 1. A current status in easy setup process.\n 2. A last error code
1072 describing reason for failure occurred at the last\n      time.\n",
1073       "parameters": [
1074         ],
1075       "responses": {
1076         "200": {
1077           "description": "",
1078           "x-example":
1079             {
1080               "rt" : ["oic.r.easyssetup", "oic.wk.col"],
1081               "if" : ["oic.if.ll", "oic.if.baseline", "oic.if.b"],
1082               "ps" : 0,
1083               "lec" : 0,
1084               "cn" : [1],
1085               "links": [
1086                 {
1087                   "href": "/EasySetupResURI",
1088                   "rt": ["oic.r.easyssetup", "oic.wk.col"],
1089                   "if": ["oic.if.b"],
1090                   "p":{"bm":3},
1091                   "eps": [
1092                     {"ep": "coaps://[fe80::bld6]:1111", "pri": 2}
1093                   ],
1094                   "rel":["self", "item"]
1095                 },
1096                 {
1097                   "href": "/WiFiConfResURI",
1098                   "rt": ["oic.r.wificonf"],
1099                   "if": ["oic.if.baseline"],
1100                   "p":{"bm":3},
1101                   "eps": [
1102                     {"ep": "coaps://[fe80::bld6]:1111", "pri": 2}
1103                   ]
1104                 },
1105                 {
1106                   "href": "/DevConfResURI",
1107                   "rt": ["oic.r.devconf"],
1108                   "if": ["oic.if.baseline"],
1109                   "p":{"bm":3},
1110                   "eps": [
1111                     {"ep": "coaps://[fe80::bld6]:1111", "pri": 2}
1112                   ]
1113                 }
1114               ]
1115             }
1116         },
1117         "schema": { "$ref": "#/definitions/EasySetup" }
1118       }
1119     }
1120   }
1121 },
1122 "/exampleEasySetupLLInterfaceResURI" : {
1123   "get": {
1124     "description": "Easy Setup resource stores useful information including current status
1125 of\nunboxing device and last error code which are produced in a process of\neasy setup.\nNote that,
1126 Easy Setup resource is a type of collection resource, which\ncontains links to WiFiConf, DevConf
1127 resources and may additionally contain\nlinks to other resources.\nRetrieve useful information
1128 during easy setup process :\n1. A current status in easy setup process.\n2. A last error code

```

```

1129 describing reason for failure occurred at the last\n  time.\n",
1130   "parameters": [
1131   ],
1132   "responses": {
1133     "200": {
1134       "description": "",
1135       "x-example":
1136         [
1137           {
1138             "href": "/EasySetupResURI",
1139             "rt": ["oic.r.easyssetup", "oic.wk.col"],
1140             "if": ["oic.if.b"],
1141             "p":{"bm":3},
1142             "eps": [
1143               {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2}
1144             ],
1145             "rel":["self", "item"]
1146           },
1147           {
1148             "href": "/WiFiConfResURI",
1149             "rt": ["oic.r.wificonf"],
1150             "if": ["oic.if.baseline"],
1151             "p":{"bm":3},
1152             "eps": [
1153               {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2}
1154             ]
1155           },
1156           {
1157             "href": "/DevConfResURI",
1158             "rt": ["oic.r.devconf"],
1159             "if": ["oic.if.baseline"],
1160             "p":{"bm":3},
1161             "eps": [
1162               {"ep": "coaps://[fe80::b1d6]:1111", "pri": 2}
1163             ]
1164           }
1165         ]
1166       ,
1167       "schema": { "$ref": "#/definitions/slinks" }
1168     }
1169   }
1170 },
1171 },
1172 "/exampleEasySetupBatchInterfaceResURI" : {
1173   "get": {
1174     "description": "Easy Setup resource stores useful information including current status
1175 of\n\nboxing device and last error code which are produced in a process of\neasy setup.\n\nNote that,
1176 Easy Setup resource is a type of collection resource, which\ncontains links to WiFiConf, DevConf
1177 resources and may additionally contain\nlinks to other resources.\n\nRetrieve useful information
1178 during easy setup process :\n\n1. A current status in easy setup process.\n\n2. A last error code
1179 describing reason for failure occurred at the last\n  time.\n",
1180     "parameters": [
1181     ],
1182     "responses": {
1183       "200": {
1184         "description": "",
1185         "x-example":
1186           [
1187             {
1188               "href": "/EasySetupResURI",
1189               "rep":{"
1190                 "ps" : 0,
1191                 "lec": 0,
1192                 "cn": [1]
1193               }
1194             },
1195             {
1196               "href": "/WiFiConfResURI",
1197               "rep":{"
1198                 "swmt" : ["A", "B", "G"],
1199                 "swf": ["2.4G", "5G"],

```

```

1200         "tnn": "Home_AP_SSID",
1201         "cd": "Home_AP_PWD",
1202         "wat": "WPA2_PSK",
1203         "wet": "AES"
1204     }
1205 },
1206 {
1207     "href": "/DevConfResURI",
1208     "rep":{
1209         "dn" : "My Refrigerator"
1210     }
1211 }
1212 ]
1213 ,
1214 "schema": { "$ref": "#/definitions/sbatch" }
1215 }
1216 },
1217 ],
1218 "post": {
1219     "description": "Able to deliver Wi-Fi, Device configuration and other
1220 configuration\ninformation in a batch by utilizing 'batch' interface.\nIf you want to deliver Wi-Fi
1221 and Device configuration information in a batch,\nyou can write all properties you want to send
1222 with a 'batch' interface.\nThe below example is the case to send Easy Setup and Wi-Fi
1223 configuration\n(i.e. connection type, target network, auth type information) in a batch.\n",
1224     "parameters": [
1225         {
1226             "name": "body",
1227             "in": "body",
1228             "required": true,
1229             "schema": { "$ref": "#/definitions/sbatch-update" },
1230             "x-example":
1231             [
1232                 {
1233                     "href": "/EasySetupResURI",
1234                     "rep":{
1235                         "cn": [1]
1236                     }
1237                 },
1238                 {
1239                     "href": "/WiFiConfResURI",
1240                     "rep":{
1241                         "tnn": "Home_AP_SSID",
1242                         "cd": "Home_AP_PWD",
1243                         "wat": "WPA2_PSK",
1244                         "wet": "AES"
1245                     }
1246                 }
1247             ]
1248         }
1249     ],
1250     "responses": {
1251         "200": {
1252             "description": "",
1253             "x-example":
1254             [
1255                 {
1256                     "href": "/EasySetupResURI",
1257                     "rep" : {
1258                         "ps" : 0,
1259                         "lec": 0,
1260                         "cn": [1]
1261                     }
1262                 },
1263                 {
1264                     "href": "/WiFiConfResURI",
1265                     "rep" : {
1266                         "swmt" : ["A", "B", "G"],
1267                         "swf" : ["2.4G", "5G"],
1268                         "tnn": "Home_AP_SSID",
1269                         "cd": "Home_AP_PWD",
1270                         "wat": "WPA2_PSK",

```

```

1271         "wet": "AES"
1272     }
1273     },
1274     {
1275         "href": "/DevConfResURI",
1276         "rep": {
1277             "dn": "My Refrigerator"
1278         }
1279     }
1280 ]
1281 ,
1282 "schema": { "$ref": "#/definitions/sbatch" }
1283 }
1284 }
1285 }
1286 }
1287 },
1288 "parameters": {
1289     "interface-ll" : {
1290         "in" : "query",
1291         "name" : "if",
1292         "type" : "string",
1293         "enum" : ["oic.if.ll"]
1294     },
1295     "interface-baseline" : {
1296         "in" : "query",
1297         "name" : "if",
1298         "type" : "string",
1299         "enum" : ["oic.if.baseline"]
1300     },
1301     "interface-all" : {
1302         "in" : "query",
1303         "name" : "if",
1304         "type" : "string",
1305         "enum" : ["oic.if.baseline", "oic.if.ll", "oic.if.b"]
1306     },
1307     "interface-batch" : {
1308         "in" : "query",
1309         "name" : "if",
1310         "type" : "string",
1311         "enum" : ["oic.if.b"]
1312     }
1313 },
1314 "definitions": {
1315     "EasySetup" : {
1316         "properties": {
1317             "rt" :
1318                 {
1319                     "items": {
1320                         "enum": [
1321                             "oic.r.easyssetup",
1322                             "oic.wk.col"
1323                         ]
1324                     },
1325                     "maxItems": 2,
1326                     "minItems": 2,
1327                     "type": "array",
1328                     "uniqueItems": true
1329                 },
1330             "ps" :
1331                 {
1332                     "description": "Indicates the easy setup status of the device. (0: Need to Setup, 1:
1333 Connecting to Enroller, 2: Connected to Enroller, 3: Failed to Connect to Enroller, 4-254:
1334 Reserved, 255: EOF)",
1335                     "enum": [
1336                         0,
1337                         1,
1338                         2,
1339                         3
1340                     ],
1341

```

```

1342         "readOnly": true,
1343         "type": "integer"
1344     },
1345     "lec" :
1346     {
1347         "description": "Indicates a failure reason (0: NO error, 1: A given SSID is not found, 2:
1348 Wi-Fi's password is wrong, 3: IP address is not allocated, 4: No internet connection, 5: Timeout,
1349 6: Wi-Fi Auth Type is not supported by the Enrollee, 7: Wi-Fi Encryption Type is not supported by
1350 the Enrollee, 8: Wi-Fi Auth Type is wrong (failure while connecting to the Enroller), 9: Wi-Fi
1351 Encryption Type is wrong (failure while connecting to the Enroller), 10~254: Reserved, 255: Unknown
1352 error)",
1353         "enum": [
1354             0,
1355             1,
1356             2,
1357             3,
1358             4,
1359             5,
1360             6,
1361             7,
1362             8,
1363             9,
1364             255
1365         ],
1366         "readOnly": true,
1367         "type": "integer"
1368     },
1369 },
1370
1371 "cn" :
1372 {
1373     "description": "Indicates an array of connection types that trigger an attempt to connect
1374 to the Enroller to start.",
1375     "items": {
1376         "description": "Connection type to attempt. (1 : Wi-Fi, 2 : other entities / transports
1377 to be added in future (e.g. Connect to cloud / BLE))",
1378         "type": "integer"
1379     },
1380     "type": "array"
1381 },
1382
1383 "links" :
1384 {
1385     "description": "A set of simple or individual OIC Links.",
1386     "items": {
1387         "properties": {
1388             "anchor": {
1389                 "description": "This is used to override the context URI e.g. override the URI of
1390 the containing collection.",
1391                 "format": "uri",
1392                 "maxLength": 256,
1393                 "type": "string"
1394             },
1395             "di": {
1396                 "description": "The Device ID formatted according to IETF RFC 4122.",
1397                 "pattern": "^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-
1398 9]{12}$",
1399                 "type": "string"
1400             },
1401             "eps": {
1402                 "description": "the Endpoint information of the target Resource",
1403                 "items": {
1404                     "properties": {
1405                         "ep": {
1406                             "description": "Transport Protocol Suite + Endpoint Locator",
1407                             "format": "uri",
1408                             "type": "string"
1409                         },
1410                         "pri": {
1411                             "description": "The priority among multiple Endpoints",
1412                             "minimum": 1,

```

```

1413         "type": "integer"
1414     },
1415     },
1416     "type": "object"
1417 },
1418 "type": "array"
1419 },
1420 "href": {
1421     "description": "This is the target URI, it can be specified as a Relative Reference
1422 or fully-qualified URI.",
1423     "format": "uri",
1424     "maxLength": 256,
1425     "type": "string"
1426 },
1427 "if": {
1428     "description": "The interface set supported by this resource",
1429     "items": {
1430         "enum": [
1431             "oic.if.baseline",
1432             "oic.if.ll",
1433             "oic.if.b",
1434             "oic.if.rw",
1435             "oic.if.r",
1436             "oic.if.a",
1437             "oic.if.s"
1438         ],
1439         "type": "string"
1440     },
1441     "minItems": 1,
1442     "type": "array"
1443 },
1444 "ins": {
1445     "description": "The instance identifier for this web link in an array of web links
1446 - used in collections",
1447     "type": "integer"
1448 },
1449 "p": {
1450     "description": "Specifies the framework policies on the Resource referenced by the
1451 target URI",
1452     "properties": {
1453         "bm": {
1454             "description": "Specifies the framework policies on the Resource referenced by
1455 the target URI for e.g. observable and discoverable",
1456             "type": "integer"
1457         }
1458     },
1459     "required": [
1460         "bm"
1461     ],
1462     "type": "object"
1463 },
1464 "rel": {
1465     "description": "The relation of the target URI referenced by the link to the
1466 context URI",
1467     "oneOf": [
1468         {
1469             "default": [
1470                 "hosts"
1471             ],
1472             "items": {
1473                 "maxLength": 64,
1474                 "type": "string"
1475             },
1476             "minItems": 1,
1477             "type": "array"
1478         },
1479         {
1480             "default": "hosts",
1481             "maxLength": 64,
1482             "type": "string"
1483         }
1484     ]
1485 }

```



```

1484     ]
1485   },
1486   "rt": {
1487     "description": "Resource Type of the Resource",
1488     "items": {
1489       "maxLength": 64,
1490       "type": "string"
1491     },
1492     "minItems": 1,
1493     "type": "array"
1494   },
1495   "title": {
1496     "description": "A title for the link relation. Can be used by the UI to provide a
1497 context.",
1498     "maxLength": 64,
1499     "type": "string"
1500   },
1501   "type": {
1502     "default": "application/cbor",
1503     "description": "A hint at the representation of the resource referenced by the
1504 target URI. This represents the media types that are used for both accepting and emitting.",
1505     "items": {
1506       "maxLength": 64,
1507       "type": "string"
1508     },
1509     "minItems": 1,
1510     "type": "array"
1511   }
1512 },
1513 "required": [
1514   "href",
1515   "rt",
1516   "if"
1517 ],
1518 "type": "object"
1519 },
1520 "type": "array"
1521 },
1522 "n" :
1523 {
1524   "description": "Friendly name of the resource",
1525   "maxLength": 64,
1526   "readOnly": true,
1527   "type": "string"
1528 },
1529 },
1530 "rts" :
1531 {
1532   "description": "Resource Type of the Resource",
1533   "items": {
1534     "maxLength": 64,
1535     "type": "string"
1536   },
1537   "minItems": 1,
1538   "readOnly": true,
1539   "type": "array"
1540 },
1541 },
1542 "id" :
1543 {
1544   "description": "Instance ID of this specific resource",
1545   "maxLength": 64,
1546   "readOnly": true,
1547   "type": "string"
1548 },
1549 },
1550 "if" :
1551 {
1552   "description": "The interface set supported by this resource",
1553   "items": {
1554

```

```

1555         "enum": [
1556             "oic.if.baseline",
1557             "oic.if.ll",
1558             "oic.if.b",
1559             "oic.if.lb",
1560             "oic.if.rw",
1561             "oic.if.r",
1562             "oic.if.a",
1563             "oic.if.s"
1564         ],
1565         "type": "string"
1566     },
1567     "minItems": 1,
1568     "readOnly": true,
1569     "type": "array"
1570 }
1571 }
1572 }
1573 }
1574 ,
1575 "slinks" : {
1576     "items" :
1577     {
1578         "properties": {
1579             "anchor": {
1580                 "description": "This is used to override the context URI e.g. override the URI of the
1581 containing collection.",
1582                 "format": "uri",
1583                 "maxLength": 256,
1584                 "type": "string"
1585             },
1586             "di": {
1587                 "description": "The Device ID formatted according to IETF RFC 4122.",
1588                 "pattern": "^[a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-
1589 9]{12}$",
1590                 "type": "string"
1591             },
1592             "eps": {
1593                 "description": "the Endpoint information of the target Resource",
1594                 "items": {
1595                     "properties": {
1596                         "ep": {
1597                             "description": "Transport Protocol Suite + Endpoint Locator",
1598                             "format": "uri",
1599                             "type": "string"
1600                         },
1601                         "pri": {
1602                             "description": "The priority among multiple Endpoints",
1603                             "minimum": 1,
1604                             "type": "integer"
1605                         }
1606                     },
1607                     "type": "object"
1608                 },
1609                 "type": "array"
1610             },
1611             "href": {
1612                 "description": "This is the target URI, it can be specified as a Relative Reference or
1613 fully-qualified URI.",
1614                 "format": "uri",
1615                 "maxLength": 256,
1616                 "type": "string"
1617             },
1618             "if": {
1619                 "description": "The interface set supported by this resource",
1620                 "items": {
1621                     "enum": [
1622                         "oic.if.baseline",
1623                         "oic.if.ll",
1624                         "oic.if.b",
1625                         "oic.if.rw",

```

```

1626         "oic.if.r",
1627         "oic.if.a",
1628         "oic.if.s"
1629     ],
1630     "type": "string"
1631 },
1632     "minItems": 1,
1633     "type": "array"
1634 },
1635     "ins": {
1636         "description": "The instance identifier for this web link in an array of web links -
1637 used in collections",
1638         "type": "integer"
1639     },
1640     "p": {
1641         "description": "Specifies the framework policies on the Resource referenced by the
1642 target URI",
1643         "properties": {
1644             "bm": {
1645                 "description": "Specifies the framework policies on the Resource referenced by the
1646 target URI for e.g. observable and discoverable",
1647                 "type": "integer"
1648             }
1649         },
1650         "required": [
1651             "bm"
1652         ],
1653         "type": "object"
1654     },
1655     "rel": {
1656         "description": "The relation of the target URI referenced by the link to the context
1657 URI",
1658         "oneOf": [
1659             {
1660                 "default": [
1661                     "hosts"
1662                 ],
1663                 "items": {
1664                     "maxLength": 64,
1665                     "type": "string"
1666                 },
1667                 "minItems": 1,
1668                 "type": "array"
1669             },
1670             {
1671                 "default": "hosts",
1672                 "maxLength": 64,
1673                 "type": "string"
1674             }
1675         ]
1676     },
1677     "rt": {
1678         "description": "Resource Type of the Resource",
1679         "items": {
1680             "maxLength": 64,
1681             "type": "string"
1682         },
1683         "minItems": 1,
1684         "type": "array"
1685     },
1686     "title": {
1687         "description": "A title for the link relation. Can be used by the UI to provide a
1688 context.",
1689         "maxLength": 64,
1690         "type": "string"
1691     },
1692     "type": {
1693         "default": "application/cbor",
1694         "description": "A hint at the representation of the resource referenced by the target
1695 URI. This represents the media types that are used for both accepting and emitting.",
1696         "items": {

```

```

1697         "maxLength": 64,
1698         "type": "string"
1699     },
1700     "minItems": 1,
1701     "type": "array"
1702 }
1703 },
1704 "required": [
1705     "href",
1706     "rt",
1707     "if"
1708 ],
1709 "type": "object"
1710 }
1711
1712 , "type" :
1713     "array"
1714
1715 , "title" :
1716     "EasySetup Object Links List Schema (auto merged)"
1717
1718 }
1719
1720 "sbatch" : {
1721     "title" :
1722         "EasySetup Batch Retrieve Format (auto merged)"
1723
1724     , "minItems" :
1725         1
1726
1727     , "items" :
1728         {
1729             "items": {
1730                 "additionalProperties": true,
1731                 "properties": {
1732                     "href": {
1733                         "description": "URI of the target resource relative assuming the collection URI as
1734 anchor",
1735                         "format": "uri",
1736                         "maxLength": 256,
1737                         "type": "string"
1738                     },
1739                     "rep": {
1740                         "oneOf": [
1741                             {
1742                                 "description": "The response payload from a single resource",
1743                                 "type": "object"
1744                             },
1745                             {
1746                                 "description": " The response payload from a collection (batch) resource",
1747                                 "items": {
1748                                     "type": "object"
1749                                 },
1750                                 "type": "array"
1751                             }
1752                         ]
1753                     }
1754                 },
1755                 "required": [
1756                     "href",
1757                     "rep"
1758                 ],
1759                 "type": "object"
1760             },
1761             "minItems": 1,
1762             "type": "array"
1763         }
1764
1765     , "type" :
1766         "array"
1767

```

```

1768     }
1769     ,
1770     "sbatch-update" : {
1771         "title" :
1772             "EasySetup Batch Update Format (auto merged)"
1773
1774         , "minItems" :
1775             1
1776
1777         , "items" :
1778             {
1779                 "items": {
1780                     "description": "array of resource representations to apply to the batch collection, using
1781 href to indicate which resource(s) in the batch to update. If the href property is empty,
1782 effectively making the URI reference to the collection itself, the representation is to be applied
1783 to all resources in the batch",
1784                     "items": {
1785                         "additionalProperties": true,
1786                         "properties": {
1787                             "href": {
1788                                 "description": "URI of the target resource relative assuming the collection URI as
1789 anchor",
1790                                 "format": "uri",
1791                                 "maxLength": 256,
1792                                 "type": "string"
1793                             },
1794                             "rep": {
1795                                 "oneOf": [
1796                                     {
1797                                         "description": "The response payload from a single resource",
1798                                         "type": "object"
1799                                     },
1800                                     {
1801                                         "description": " The response payload from a collection (batch) resource",
1802                                         "items": {
1803                                             "type": "object"
1804                                         },
1805                                         "type": "array"
1806                                     }
1807                                 ]
1808                             }
1809                         },
1810                         "required": [
1811                             "href",
1812                             "rep"
1813                         ],
1814                         "type": "object"
1815                     },
1816                     "minItems": 1,
1817                     "type": "array"
1818                 }
1819             }
1820
1821         , "type" :
1822             "array"
1823
1824     }
1825 }
1826 }
1827

```

1828 **B.2.5 Property Definition**

Property name	Value type	Mandatory	Access mode	Description
href	string	yes		URI of the target resource relative assuming the collection URI as anchor

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

				the target Resource
di	string			The Device ID formatted according to IETF RFC 4122.
rt	array: see schema	yes		Resource Type of the Resource
title	string			A title for the link relation. Can be used by the UI to provide a context.
id	string		Read Only	Instance ID of this specific resource
cn	array: see schema			Indicates an array of connection types that trigger an attempt to connect to the Enroller to start.
ps	integer		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: see schema	yes	Read Only	The interface set supported by this resource
lec	integer		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)
rts	array: see schema		Read Only	Resource Type of the Resource
rt	array: see schema	yes		
n	string		Read Only	Friendly name of the resource
links	array: see schema			A set of simple or individual OIC Links.

1829 **B.2.6 CRUDN behaviour**

Resource	Create	Read	Update	Delete	Notify
/exampleEasySetupBaselineInterfaceResURI		get			

1830 **B.3 Wi-Fi Configuration Resource**

1831 **B.3.1 Introduction**

1832 WiFiConf resource stores essential information to help an unboxing device
 1833 to connect to an existing Wi-Fi AP.
 1834

1835 **B.3.2 Example URI**

1836 /example/WiFiConfBaselineInterfaceResURI

1837 **B.3.3 Resource Type**

1838 The resource type (rt) is defined as: ['oic.r.wificnf'].

1839 **B.3.4 Swagger2.0 Definition**

```

1840 {
1841   "swagger": "2.0",
1842   "info": {
1843     "title": "Wi-Fi Configuration Resource Baseline Interface",
1844     "version": "v0.0.3-20170611",
1845     "license": {
1846       "name": "copyright 2016-2017 Open Connectivity Foundation, Inc. All rights reserved.",
1847       "x-description": "Redistribution and use in source and binary forms, with or without
1848 modification, are permitted provided that the following conditions are met:\n      1.
1849 Redistributions of source code must retain the above copyright notice, this list of conditions and/or
1850 the following disclaimer.\n      2. Redistributions in binary form must reproduce the above
1851 copyright notice, this list of conditions and the following disclaimer in the documentation and/or
1852 other materials provided with the distribution.\n\n      THIS SOFTWARE IS PROVIDED BY THE Open
1853 Connectivity Foundation, INC. \\"AS IS\\" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
1854 LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE OR

```



```

1855 WARRANTIES OF NON-INFRINGEMENT, ARE DISCLAIMED.\n          IN NO EVENT SHALL THE Open Connectivity
1856 Foundation, INC. OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
1857 EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
1858 OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)\n          HOWEVER CAUSED AND
1859 ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
1860 OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY
1861 OF SUCH DAMAGE.\n"
1862 }
1863 },
1864 "schemes": ["http"],
1865 "consumes": ["application/json"],
1866 "produces": ["application/json"],
1867 "paths": {
1868   "/example/WiFiConfBaselineInterfaceResURI" : {
1869     "get": {
1870       "description": "WiFiConf resource stores essential information to help an unboxing
1871 device\nto connect to an existing Wi-Fi AP.\nRetrieve properties of WiFiConf resource.\nThe
1872 information includes :\n1. Wi-Fi SSID and password\n2. Wi-Fi Security type (i.e. auth type and
1873 encryption type)\n3. Wi-Fi hardware capability (i.e. supported frequencies, modes,\n auth types
1874 and encryption types)\n",
1875       "parameters": [
1876         ],
1877       "responses": {
1878         "200": {
1879           "description": "",
1880           "x-example":
1881             {
1882               "rt": ["oic.r.wificonf"],
1883               "swmt": ["A", "B", "G"],
1884               "swf": ["2.4G", "5G"],
1885               "tnn": "Home_AP_SSID",
1886               "cd": "Home_AP_PWD",
1887               "wat": "WPA2_PSK",
1888               "wet": "TKIP",
1889               "swat": ["WPA_PSK", "WPA2_PSK"],
1890               "swet": ["TKIP", "AES", "TKIP_AES"]
1891             }
1892         },
1893         "schema": { "$ref": "#/definitions/WiFiConf" }
1894       }
1895     }
1896   },
1897   "post": {
1898     "description": "Deliver Wi-Fi AP's information for an unboxing device to connect to it.\n",
1899     "parameters": [
1900       {
1901         "name": "body",
1902         "in": "body",
1903         "required": true,
1904         "schema": { "$ref": "#/definitions/WiFiConfUpdate" },
1905         "x-example":
1906           {
1907             "tnn": "Home_AP_SSID",
1908             "cd": "Home_AP_PWD",
1909             "wat": "WPA2_PSK",
1910             "wet": "AES"
1911           }
1912       }
1913     ],
1914     "responses": {
1915       "200": {
1916         "description": "",
1917         "x-example":
1918           {
1919             "tnn": "Home_AP_SSID",
1920             "cd": "Home_AP_PWD",
1921             "wat": "WPA2_PSK",
1922             "wet": "AES"
1923           }
1924       },
1925       "schema": { "$ref": "#/definitions/WiFiConfUpdate" }
1926     }
1927   }
1928 }

```

```

1926     }
1927   }
1928 }
1929 },
1930 "/example/WiFiConfRWInterfaceResURI" : {
1931   "get": {
1932     "description": "WiFiConf resource stores essential information to help an unboxing
1933 device\nto connect to an existing Wi-Fi AP.\nRetrieve properties of WiFiConf resource that can be
1934 updated by a client.\n",
1935     "parameters": [
1936     ],
1937     "responses": {
1938       "200": {
1939         "description": "",
1940         "x-example":
1941           {
1942             "tnn": "Home_AP_SSID",
1943             "cd": "Home_AP_PWD",
1944             "wat": "WPA2_PSK",
1945             "wet": "AES"
1946           }
1947       },
1948       "schema": { "$ref": "#/definitions/WiFiConfUpdate" }
1949     }
1950   }
1951 },
1952 "post": {
1953   "description": "Deliver Wi-Fi AP's information for an unboxing device to connect to it.\n",
1954   "parameters": [
1955     {
1956       "name": "body",
1957       "in": "body",
1958       "required": true,
1959       "schema": { "$ref": "#/definitions/WiFiConfUpdate" },
1960       "x-example":
1961         {
1962           "tnn": "Home_AP_SSID",
1963           "cd": "Home_AP_PWD",
1964           "wat": "WPA2_PSK",
1965           "wet": "AES"
1966         }
1967     }
1968   ],
1969   "responses": {
1970     "200": {
1971       "description": "",
1972       "x-example":
1973         {
1974           "tnn": "Home_AP_SSID",
1975           "cd": "Home_AP_PWD",
1976           "wat": "WPA2_PSK",
1977           "wet": "AES"
1978         }
1979     },
1980     "schema": { "$ref": "#/definitions/WiFiConfUpdate" }
1981   }
1982 }
1983 }
1984 },
1985 },
1986 "parameters": {
1987   "interface-rw" : {
1988     "in" : "query",
1989     "name" : "if",
1990     "type" : "string",
1991     "enum" : ["oic.if.rw"]
1992   },
1993   "interface-baseline" : {
1994     "in" : "query",
1995     "name" : "if",
1996     "type" : "string",

```

```

1997     "enum" : ["oic.if.baseline"]
1998   },
1999   "interface-all" : {
2000     "in" : "query",
2001     "name" : "if",
2002     "type" : "string",
2003     "enum" : ["oic.if.baseline", "oic.if.rw"]
2004   }
2005 },
2006 "definitions": {
2007   "WiFiConf" : {
2008     "properties": {
2009       "rt" :
2010         {
2011           "description": "Resource Type of the Resource",
2012           "items": {
2013             "maxLength": 64,
2014             "type": "string"
2015           },
2016           "minItems": 1,
2017           "readOnly": true,
2018           "type": "array"
2019         },
2020
2021       "tnn" :
2022         {
2023           "description": "Indicates Target Network Name (SSID of Wi-Fi AP)",
2024           "pattern": "^.*$",
2025           "type": "string"
2026         },
2027
2028       "swmt" :
2029         {
2030           "description": "Indicates supported Wi-Fi mode types. It can be multiple",
2031           "items": {
2032             "description": "Supported Wi-Fi Mode Type.",
2033             "enum": [
2034               "A",
2035               "B",
2036               "G",
2037               "N",
2038               "AC"
2039             ],
2040             "type": "string"
2041           },
2042           "readOnly": true,
2043           "type": "array"
2044         },
2045
2046       "wat" :
2047         {
2048           "description": "Indicates Wi-Fi Auth Type",
2049           "enum": [
2050             "None",
2051             "WEP",
2052             "WPA_PSK",
2053             "WPA2_PSK"
2054           ],
2055           "type": "string"
2056         },
2057
2058       "n" :
2059         {
2060           "description": "Friendly name of the resource",
2061           "maxLength": 64,
2062           "readOnly": true,
2063           "type": "string"
2064         },
2065
2066       "swat" :
2067         {

```

```

2068         "description": "Indicates supported Wi-Fi Auth types. It can be multiple",
2069         "items": {
2070             "description": "Indicates Wi-Fi Auth Type",
2071             "enum": [
2072                 "None",
2073                 "WEP",
2074                 "WPA_PSK",
2075                 "WPA2_PSK"
2076             ],
2077             "type": "string"
2078         },
2079         "readOnly": true,
2080         "type": "array"
2081     },
2082
2083     "swf" :
2084     {
2085         "description": "Indicates Supported Wi-Fi frequencies by the Enrollee. Can be multiple.
2086 Valid values are ('2.4G', '5G')",
2087         "items": {
2088             "pattern": "^(2\\.4|5)G$",
2089             "type": "string"
2090         },
2091         "readOnly": true,
2092         "type": "array"
2093     },
2094
2095     "swet" :
2096     {
2097         "description": "Indicates supported Wi-Fi Encryption types. It can be multiple",
2098         "items": {
2099             "description": "Indicates Wi-Fi Encryption Type",
2100             "enum": [
2101                 "None",
2102                 "WEP_64",
2103                 "WEP_128",
2104                 "TKIP",
2105                 "AES",
2106                 "TKIP_AES"
2107             ],
2108             "type": "string"
2109         },
2110         "readOnly": true,
2111         "type": "array"
2112     },
2113
2114     "wet" :
2115     {
2116         "description": "Indicates Wi-Fi Encryption Type",
2117         "enum": [
2118             "None",
2119             "WEP_64",
2120             "WEP_128",
2121             "TKIP",
2122             "AES",
2123             "TKIP_AES"
2124         ],
2125         "type": "string"
2126     },
2127
2128     "cd" :
2129     {
2130         "description": "Indicates credential information of Wi-Fi AP",
2131         "pattern": "^.*$",
2132         "type": "string"
2133     },
2134
2135     "id" :
2136     {
2137         "description": "Instance ID of this specific resource",
2138         "maxLength": 64,

```

```

2139         "readOnly": true,
2140         "type": "string"
2141     },
2142
2143     "if" :
2144     {
2145         "description": "The interface set supported by this resource",
2146         "items": {
2147             "enum": [
2148                 "oic.if.baseline",
2149                 "oic.if.ll",
2150                 "oic.if.b",
2151                 "oic.if.lb",
2152                 "oic.if.rw",
2153                 "oic.if.r",
2154                 "oic.if.a",
2155                 "oic.if.s"
2156             ],
2157             "type": "string"
2158         },
2159         "minItems": 1,
2160         "readOnly": true,
2161         "type": "array"
2162     }
2163 }
2164 }
2165 }
2166 /
2167 "WiFiConfUpdate" : {
2168     "properties": {
2169         "rt" :
2170         {
2171             "description": "Resource Type of the Resource",
2172             "items": {
2173                 "maxLength": 64,
2174                 "type": "string"
2175             },
2176             "minItems": 1,
2177             "readOnly": true,
2178             "type": "array"
2179         },
2180
2181         "wat" :
2182         {
2183             "description": "Indicates Wi-Fi Auth Type",
2184             "enum": [
2185                 "None",
2186                 "WEP",
2187                 "WPA_PSK",
2188                 "WPA2_PSK"
2189             ]
2190         },
2191
2192         "n" :
2193         {
2194             "description": "Friendly name of the resource",
2195             "maxLength": 64,
2196             "readOnly": true,
2197             "type": "string"
2198         },
2199
2200         "cd" :
2201         {
2202             "description": "Indicates credential information of Wi-Fi AP",
2203             "pattern": "^.*$",
2204             "type": "string"
2205         },
2206
2207         "wet" :
2208         {
2209             "description": "Indicates Wi-Fi Encryption Type",

```

```

2210     "enum": [
2211         "None",
2212         "WEP_64",
2213         "WEP_128",
2214         "TKIP",
2215         "AES",
2216         "TKIP_AES"
2217     ]
2218 },
2219
2220 "tnn" :
2221 {
2222     "description": "Indicates Target Network Name (SSID of Wi-Fi AP)",
2223     "pattern": "^.*$",
2224     "type": "string"
2225 },
2226
2227 "id" :
2228 {
2229     "description": "Instance ID of this specific resource",
2230     "maxLength": 64,
2231     "readOnly": true,
2232     "type": "string"
2233 },
2234
2235 "if" :
2236 {
2237     "description": "The interface set supported by this resource",
2238     "items": {
2239         "enum": [
2240             "oic.if.baseline",
2241             "oic.if.ll",
2242             "oic.if.b",
2243             "oic.if.lb",
2244             "oic.if.rw",
2245             "oic.if.r",
2246             "oic.if.a",
2247             "oic.if.s"
2248         ],
2249         "type": "string"
2250     },
2251     "minItems": 1,
2252     "readOnly": true,
2253     "type": "array"
2254 }
2255 }
2256 }
2257 }
2258 }
2259 }
2260 }

```

2261 **B.3.5 Property Definition**

Property name	Value type	Mandatory	Access mode	Description
wet	string			Indicates Wi-Fi Encryption Type
rt	array: see schema		Read Only	Resource Type of the Resource
n	string		Read Only	Friendly name of the resource
swet	array: see schema		Read Only	Indicates supported Wi-Fi Encryption types. It can be multiple
swmt	array: see schema		Read Only	Indicates supported Wi-Fi

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

2262

B.3.6 CRUDN behaviour

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

