



# **RESTful Network API for Device Capabilities**

Candidate Version 1.0 – 20 Dec 2011

---

**Open Mobile Alliance**

OMA-TS-REST\_NetAPI\_DeviceCapabilities-V1\_0-20111220-C

Use of this document is subject to all of the terms and conditions of the Use Agreement located at <http://www.openmobilealliance.org/UseAgreement.html>.

Unless this document is clearly designated as an approved specification, this document is a work in process, is not an approved Open Mobile Alliance™ specification, and is subject to revision or removal without notice.

You may use this document or any part of the document for internal or educational purposes only, provided you do not modify, edit or take out of context the information in this document in any manner. Information contained in this document may be used, at your sole risk, for any purposes. You may not use this document in any other manner without the prior written permission of the Open Mobile Alliance. The Open Mobile Alliance authorizes you to copy this document, provided that you retain all copyright and other proprietary notices contained in the original materials on any copies of the materials and that you comply strictly with these terms. This copyright permission does not constitute an endorsement of the products or services. The Open Mobile Alliance assumes no responsibility for errors or omissions in this document.

Each Open Mobile Alliance member has agreed to use reasonable endeavors to inform the Open Mobile Alliance in a timely manner of Essential IPR as it becomes aware that the Essential IPR is related to the prepared or published specification. However, the members do not have an obligation to conduct IPR searches. The declared Essential IPR is publicly available to members and non-members of the Open Mobile Alliance and may be found on the “OMA IPR Declarations” list at <http://www.openmobilealliance.org/ipr.html>. The Open Mobile Alliance has not conducted an independent IPR review of this document and the information contained herein, and makes no representations or warranties regarding third party IPR, including without limitation patents, copyrights or trade secret rights. This document may contain inventions for which you must obtain licenses from third parties before making, using or selling the inventions. Defined terms above are set forth in the schedule to the Open Mobile Alliance Application Form.

**NO REPRESENTATIONS OR WARRANTIES (WHETHER EXPRESS OR IMPLIED) ARE MADE BY THE OPEN MOBILE ALLIANCE OR ANY OPEN MOBILE ALLIANCE MEMBER OR ITS AFFILIATES REGARDING ANY OF THE IPR'S REPRESENTED ON THE “OMA IPR DECLARATIONS” LIST, INCLUDING, BUT NOT LIMITED TO THE ACCURACY, COMPLETENESS, VALIDITY OR RELEVANCE OF THE INFORMATION OR WHETHER OR NOT SUCH RIGHTS ARE ESSENTIAL OR NON-ESSENTIAL.**

**THE OPEN MOBILE ALLIANCE IS NOT LIABLE FOR AND HEREBY DISCLAIMS ANY DIRECT, INDIRECT, PUNITIVE, SPECIAL, INCIDENTAL, CONSEQUENTIAL, OR EXEMPLARY DAMAGES ARISING OUT OF OR IN CONNECTION WITH THE USE OF DOCUMENTS AND THE INFORMATION CONTAINED IN THE DOCUMENTS.**

© 2011 Open Mobile Alliance Ltd. All Rights Reserved.

Used with the permission of the Open Mobile Alliance Ltd. under the terms set forth above.

# Contents

<b>1. SCOPE</b> .....	<b>7</b>
<b>2. REFERENCES</b> .....	<b>8</b>
<b>2.1 NORMATIVE REFERENCES</b> .....	<b>8</b>
<b>2.2 INFORMATIVE REFERENCES</b> .....	<b>8</b>
<b>3. TERMINOLOGY AND CONVENTIONS</b> .....	<b>9</b>
<b>3.1 CONVENTIONS</b> .....	<b>9</b>
<b>3.2 DEFINITIONS</b> .....	<b>9</b>
<b>3.3 ABBREVIATIONS</b> .....	<b>9</b>
<b>4. INTRODUCTION</b> .....	<b>11</b>
<b>4.1 VERSION 1.0</b> .....	<b>11</b>
<b>5. DEVICE CAPABILITIES API DEFINITION</b> .....	<b>12</b>
<b>5.1 RESOURCES SUMMARY</b> .....	<b>12</b>
<b>5.2 DATA TYPES</b> .....	<b>17</b>
5.2.1 XML Namespaces.....	17
5.2.2 Structures .....	17
5.2.2.1 Type: DeviceCapabilities.....	17
5.2.2.2 Type: DeviceCapabilitiesNotification.....	17
5.2.2.3 Type: DeviceCapabilitiesCancellationNotification.....	18
5.2.2.4 Type: ConfigurationHistoryEntry .....	18
5.2.2.5 Type: DeviceConfigurationHistoryList.....	19
5.2.2.6 Type: DeviceConfiguration.....	19
5.2.2.7 Type: DeviceConfigurationList.....	19
5.2.2.8 Type: DeviceCapabilitiesChangeSubscriptionList.....	20
5.2.2.9 Type: DeviceCapabilitiesChangeSubscription.....	20
5.2.3 Enumerations .....	21
5.2.4 Values of the Link “rel” attribute.....	21
<b>5.3 SEQUENCE DIAGRAMS</b> .....	<b>22</b>
5.3.1 Retrieving device capabilities information .....	22
5.3.2 Subscribing for and notifying on device(s) capabilities.....	23
5.3.3 Pushing a configuration to device(s).....	23
5.3.4 Retrieving available configurations and history of configurations .....	24
<b>6. DETAILED SPECIFICATION OF THE RESOURCES</b> .....	<b>26</b>
<b>6.1 RESOURCE: DEVICE CAPABILITIES</b> .....	<b>26</b>
6.1.1 Request URL variables .....	26
6.1.2 Response Codes and Error Handling .....	26
6.1.3 GET.....	27
6.1.3.1 Example: Retrieve device capabilities (Informative).....	27
6.1.3.1.1 Request.....	27
6.1.3.1.2 Response.....	27
6.1.4 PUT.....	27
6.1.5 POST.....	27
6.1.6 DELETE .....	27
<b>6.2 RESOURCE: SUBSCRIPTIONS FOR DEVICE CAPABILITIES CHANGES NOTIFICATIONS</b> .....	<b>27</b>
6.2.1 Request URL variables .....	28
6.2.2 Response Codes and Error Handling .....	28
6.2.3 GET.....	28
6.2.3.1 Example: Retrieve list of subscriptions (Informative).....	28
6.2.3.1.1 Request.....	28
6.2.3.1.2 Response.....	28
6.2.4 PUT.....	29
6.2.5 POST.....	29
6.2.5.1 Example 1: Subscribe to device capabilities changes notifications for single device using ‘tel’ URI (Informative).....	29
6.2.5.1.1 Request.....	29
6.2.5.1.2 Response.....	29

6.2.5.2	<i>Example 2: Subscribe to device capabilities changes notifications for single device using 'acr' URI (Informative)</i> .....	30
6.2.5.2.1	Request.....	30
6.2.5.2.2	Response.....	30
6.2.5.3	<i>Example 3: Subscribe to device capabilities changes notifications for group of devices (Informative)</i> .....	31
6.2.5.3.1	Request.....	31
6.2.5.3.2	Response.....	31
6.2.5.4	<i>Example 4: Subscribe to device capabilities changes notifications for single device (with resource reference instead of full resource representation in the response body) (Informative)</i> .....	32
6.2.5.4.1	Request.....	32
6.2.5.4.2	Response.....	32
6.2.6	DELETE .....	32
<b>6.3</b>	<b>RESOURCE: INDIVIDUAL SUBSCRIPTION FOR DEVICE CAPABILITIES CHANGES NOTIFICATIONS .....</b>	<b>33</b>
6.3.1	Request URL variables .....	33
6.3.2	Response Codes and Error Handling .....	33
6.3.3	GET.....	33
6.3.3.1	<i>Example: Retrieve subscription (Informative)</i> .....	33
6.3.3.1.1	Request.....	33
6.3.3.1.2	Response.....	33
6.3.4	PUT.....	34
6.3.5	POST.....	34
6.3.6	DELETE .....	34
6.3.6.1	<i>Example: Delete subscription (Informative)</i> .....	34
6.3.6.1.1	Request.....	34
6.3.6.1.2	Response.....	34
<b>6.4</b>	<b>RESOURCE: NOTIFICATION ON DEVICE CAPABILITIES CHANGES.....</b>	<b>34</b>
6.4.1	Request URL variables .....	34
6.4.2	Response Codes and Error Handling .....	35
6.4.3	GET.....	35
6.4.4	PUT.....	35
6.4.5	POST.....	35
6.4.5.1	<i>Example 1: Device capabilities changes notification (Informative)</i> .....	35
6.4.5.1.1	Request.....	35
6.4.5.1.2	Response.....	35
6.4.5.2	<i>Example 2: End of changes notification (Informative)</i> .....	36
6.4.5.2.1	Request.....	36
6.4.5.2.2	Response.....	36
6.4.5.3	<i>Example 3: Error notification (notification cancellation) (Informative)</i> .....	36
6.4.5.3.1	Request.....	36
6.4.5.3.2	Response.....	37
6.4.6	DELETE .....	37
<b>6.5</b>	<b>RESOURCE: DEVICE CONFIGURATION .....</b>	<b>37</b>
6.5.1	Request URL variables .....	37
6.5.2	Response Codes and Error Handling .....	37
6.5.3	GET.....	37
6.5.4	PUT.....	38
6.5.5	POST.....	38
6.5.5.1	<i>Example 1: Push configuration to single device (Informative)</i> .....	38
6.5.5.1.1	Request.....	38
6.5.5.1.2	Response.....	38
6.5.5.2	<i>Example 2: Push configuration to group of devices (Informative)</i> .....	38
6.5.5.2.1	Request.....	38
6.5.5.2.2	Response.....	39
6.5.6	DELETE .....	39
<b>6.6</b>	<b>RESOURCE: AVAILABLE CONFIGURATIONS .....</b>	<b>39</b>
6.6.1	Request URL variables .....	39
6.6.2	Response Codes and Error Handling .....	39
6.6.3	GET.....	39
6.6.3.1	<i>Example: Retrieve available device configurations (Informative)</i> .....	39
6.6.3.1.1	Request.....	39
6.6.3.1.2	Response.....	40
6.6.4	PUT.....	40

6.6.5	POST.....	40
6.6.6	DELETE .....	40
<b>6.7</b>	<b>RESOURCE: CONFIGURATION HISTORY .....</b>	<b>40</b>
6.7.1	Request URL variables .....	41
6.7.2	Response Codes and Error Handling .....	41
6.7.3	GET.....	41
6.7.3.1	<i>Example: Retrieve configuration history (Informative) .....</i>	<i>41</i>
6.7.3.1.1	Request.....	41
6.7.3.1.2	Response.....	41
6.7.4	PUT.....	42
6.7.5	POST.....	42
6.7.6	DELETE .....	42
<b>APPENDIX A.</b>	<b>CHANGE HISTORY (INFORMATIVE).....</b>	<b>43</b>
<b>A.1</b>	<b>APPROVED VERSION HISTORY .....</b>	<b>43</b>
<b>A.2</b>	<b>DRAFT/CANDIDATE VERSION 1.0 HISTORY .....</b>	<b>43</b>
<b>APPENDIX B.</b>	<b>STATIC CONFORMANCE REQUIREMENTS (NORMATIVE).....</b>	<b>45</b>
<b>B.1</b>	<b>SCR FOR REST.DEV CAP SERVER .....</b>	<b>45</b>
B.1.1	SCR for REST.DevCap.AccCap Server .....	45
B.1.2	SCR for REST.DevCap.List.Subscr Server .....	45
B.1.3	SCR for REST.DevCap.Individual.Subscr Server.....	45
B.1.4	SCR for REST.DevCap.Notif Server.....	46
B.1.5	SCR for REST.DevCap.AccConfig Server.....	46
<b>APPENDIX C.</b>	<b>APPLICATION/X-WWW-FORM-URLENCODED REQUEST FORMAT FOR POST OPERATIONS (NORMATIVE) .....</b>	<b>47</b>
<b>C.1</b>	<b>START DEVICE CAPABILITIES CHANGES NOTIFICATIONS .....</b>	<b>47</b>
<b>C.1.1</b>	<b>Example 1, using 'tel' URI (Informative) .....</b>	<b>48</b>
C.1.1.1	<i>Request.....</i>	<i>48</i>
C.1.1.2	<i>Response .....</i>	<i>48</i>
<b>C.1.2</b>	<b>Example 2, using 'acr' URI (Informative) .....</b>	<b>48</b>
C.1.2.1	<i>Request.....</i>	<i>48</i>
C.1.2.2	<i>Response .....</i>	<i>49</i>
<b>C.2</b>	<b>PUSH A CONFIGURATION TO A DEVICE.....</b>	<b>49</b>
<b>C.2.1</b>	<b>Example (Informative) .....</b>	<b>49</b>
C.2.1.1	<i>Request .....</i>	<i>49</i>
C.2.1.2	<i>Response.....</i>	<i>50</i>
<b>APPENDIX D.</b>	<b>JSON EXAMPLES (INFORMATIVE) .....</b>	<b>51</b>
<b>D.1</b>	<b>RETRIEVE DEVICE CAPABILITIES (SECTION 6.1.3.1).....</b>	<b>51</b>
<b>D.2</b>	<b>RETRIEVE LIST OF SUBSCRIPTIONS (SECTION 6.2.3.1).....</b>	<b>51</b>
<b>D.3</b>	<b>SUBSCRIBE TO DEVICE CAPABILITIES CHANGES NOTIFICATIONS FOR SINGLE DEVICE USING 'TEL' URI (SECTION 6.2.5.1).....</b>	<b>52</b>
<b>D.4</b>	<b>SUBSCRIBE TO DEVICE CAPABILITIES CHANGES NOTIFICATIONS FOR SINGLE DEVICE USING 'ACR' URI (SECTION 6.2.5.2).....</b>	<b>53</b>
<b>D.5</b>	<b>RETRIEVE SUBSCRIPTION (SECTION 6.3.3.1).....</b>	<b>53</b>
<b>D.6</b>	<b>DELETE SUBSCRIPTION (SECTION 6.3.6.1).....</b>	<b>54</b>
<b>D.7</b>	<b>DEVICE CAPABILITIES CHANGES NOTIFICATION (SECTION 6.4.5.1) .....</b>	<b>54</b>
<b>D.8</b>	<b>END OF CHANGES NOTIFICATION (SECTION 6.4.5.2) .....</b>	<b>55</b>
<b>D.9</b>	<b>ERROR NOTIFICATION (NOTIFICATION CANCELLATION) (SECTION 6.4.5.3).....</b>	<b>56</b>
<b>D.10</b>	<b>PUSH CONFIGURATION TO SINGLE DEVICE (SECTION 6.5.5.1).....</b>	<b>56</b>
<b>D.11</b>	<b>RETRIEVE AVAILABLE DEVICE CONFIGURATIONS (SECTION 6.6.3.1).....</b>	<b>57</b>
<b>D.12</b>	<b>RETRIEVE CONFIGURATION HISTORY (SECTION 6.7.3.1).....</b>	<b>58</b>
<b>APPENDIX E.</b>	<b>PARLAY X OPERATIONS MAPPING (INFORMATIVE).....</b>	<b>60</b>
<b>APPENDIX F.</b>	<b>LIGHT-WEIGHT RESOURCES (INFORMATIVE).....</b>	<b>61</b>
<b>APPENDIX G.</b>	<b>AUTHORIZATION ASPECTS (NORMATIVE).....</b>	<b>62</b>
<b>G.1</b>	<b>USE WITH OMA AUTHORIZATION FRAMEWORK FOR NETWORK APIS.....</b>	<b>62</b>
G.1.1	Scope values .....	62

G.1.1.1	Definitions.....	62
G.1.1.2	Downscoping .....	62
G.1.1.3	Mapping with resources and methods.....	63
G.1.2	Use of 'acr:Authorization' .....	66

## Figures

Figure 1	Resource structure defined by this specification.....	13
Figure 2	Retrieving device capabilities .....	22
Figure 3	Subscribing for change notifications and handling notifications .....	23
Figure 4	Push configuration to device(s).....	24
Figure 5	Retrieving available configurations or configuration history for targeted device .....	25

## Tables

Table 1:	Parlay X operations mapping .....	60
Table 2:	Scope values for RESTful Device Capabilities API .....	62
Table 3:	Required scope values for: Device capabilities.....	64
Table 4:	Required scope values for: Subscription Management for device capabilities changes.....	64
Table 5:	Required scope values for: Retrieving configuration (available or history) and pushing a configuration .....	65

# 1. Scope

This specification defines a RESTful Device Capabilities API using an HTTP protocol binding, based on the similar API defined in [3GPP 29.199-18].

## 2. References

### 2.1 Normative References

- [3GPP 29.199-18] 3GPP Technical Specification, “Open Service Access (OSA); Parlay X Web Services; Part 18: Device capabilities and configuration (Release 8)”, URL:<http://www.3gpp.org/>
- [Autho4API\_10] “Authorization Framework for Network APIs”, Open Mobile Alliance™, OMA-ER-Autho4API-V1\_0, URL: <http://www.openmobilealliance.org/>
- [IETF\_ACR\_draft] “The acr URI for anonymous users”, S.Jakobsson, K.Smith, July 2011, URL: <http://tools.ietf.org/html/draft-uri-acr-extension-03>
- [REST\_NetAPI\_Common] “Common definitions for RESTful Network APIs”, Open Mobile Alliance™, OMA-TS-REST\_NetAPI\_Common-V1\_0, URL: <http://www.openmobilealliance.org/>
- [REST\_NetAPI\_NotificationChannel] “RESTful Network API for Notification Channel”, Open Mobile Alliance™, OMA-TS-REST\_NetAPI\_NotificationChannel-V1\_0, URL: <http://www.openmobilealliance.org/>
- [REST\_SUP\_DeviceCapabilities] “XML schema for the RESTful Network API for Device Capabilities”, Open Mobile Alliance™, OMA-SUP-XSD\_rest\_netapi\_devicecapabilities-V1\_0, URL:<http://www.openmobilealliance.org/>
- [RFC2119] “Key words for use in RFCs to Indicate Requirement Levels”, S. Bradner, March 1997, URL:<http://www.ietf.org/rfc/rfc2119.txt>
- [RFC2616] “Hypertext Transfer Protocol -- HTTP/1.1”, R. Fielding et. al, January 1999, URL:<http://www.ietf.org/rfc/rfc2616.txt>
- [RFC3261] “SIP: Session Initiation Protocol”, J. Rosenberg et al., June 2002, URL: <http://www.rfc-editor.org/rfc/rfc3261.txt>
- [RFC3966] “The tel URI for Telephone Numbers”, H.Schulzrinne, December 2004, URL: <http://www.ietf.org/rfc/rfc3966.txt>
- [RFC3986] “Uniform Resource Identifier (URI): Generic Syntax”, R. Fielding et. al, January 2005, URL:<http://www.ietf.org/rfc/rfc3986.txt>
- [RFC4627] “The application/json Media Type for JavaScript Object Notation (JSON)”, D. Crockford, July 2006, URL: <http://www.ietf.org/rfc/rfc4627.txt>
- [SCRRULES] “SCR Rules and Procedures”, Open Mobile Alliance™, OMA-ORG-SCR\_Rules\_and\_Procedures, URL:<http://www.openmobilealliance.org/>
- [W3C\_URLENC] HTML 4.01 Specification, Section 17.13.4 Form content types, The World Wide Web Consortium, URL: <http://www.w3.org/TR/html401/interact/forms.html#h-17.13.4.1>
- [XMLSchema1] W3C Recommendation, XML Schema Part 1: Structures Second Edition, URL: <http://www.w3.org/TR/xmlschema-1/>
- [XMLSchema2] W3C Recommendation, XML Schema Part 2: Datatypes Second Edition, URL: <http://www.w3.org/TR/xmlschema-2/>

### 2.2 Informative References

- [OMADICT] “Dictionary for OMA Specifications”, Version 2.8, Open Mobile Alliance™, OMA-ORG-Dictionary-V2\_8, URL:<http://www.openmobilealliance.org/>
- [ParlayREST\_DeviceCapabilities] “RESTful bindings for Parlay X Web Services – DeviceCapabilities”, Version 1.0, Open Mobile Alliance™, OMA-TS-ParlayREST\_DeviceCapabilities-V1\_0
- [REST\_WP] “Guidelines for RESTful Network APIs”, Open Mobile Alliance™, OMA-WP-Guidelines\_for\_RESTful\_Network\_APIs, URL:<http://www.openmobilealliance.org/>



## 3. Terminology and Conventions

### 3.1 Conventions

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [RFC2119].

All sections and appendixes, except “Scope” and “Introduction”, are normative, unless they are explicitly indicated to be informative.

### 3.2 Definitions

For the purpose of this TS, all definitions from the OMA Dictionary apply [OMADICT].

Client-side Notification URL	An HTTP URL exposed by a client, on which it is capable of receiving notifications and that can be used by the client when subscribing to notifications.
Long Polling	A variation of the traditional polling technique, where the server does not reply to a request unless a particular event, status or timeout has occurred. Once the server has sent a response, it closes the connection, and typically the client immediately sends a new request. This allows the emulation of an information push from a server to a client.
Notification Channel	A channel created on the request of the client and used to deliver notifications from a server to a client. The channel is represented as a resource and provides means for the server to post notifications and for the client to receive them via specified delivery mechanisms.  For example in the case of Long Polling the channel resource is defined by a pair of URLs. One of the URLs is used by the client as a call-back URL when subscribing for notifications. The other URL is used by the client to retrieve notifications from the Notification Server.
Notification Server	A server that is capable of creating and maintaining Notification Channels.
Server-side Notification URL	An HTTP URL exposed by a Notification Server, that identifies a Notification Channel and that can be used by a client when subscribing to notifications.

### 3.3 Abbreviations

<b>ACR</b>	Anonymous Customer Reference
<b>API</b>	Application Programming Interface
<b>HTTP</b>	HyperText Transfer Protocol
<b>ID</b>	Identifier
<b>IMEI</b>	International Mobile Equipment Identity
<b>JSON</b>	JavaScript Object Notation
<b>MIME</b>	Multipurpose Internet Mail Extensions
<b>MSISDN</b>	Mobile Subscriber ISDN Number
<b>OMA</b>	Open Mobile Alliance
<b>REST</b>	REpresentational State Transfer
<b>SCR</b>	Static Conformance Requirements
<b>SIP</b>	Session Initiation Protocol
<b>TS</b>	Technical Specification
<b>UAProfile</b>	User Agent Profile
<b>URI</b>	Uniform Resource Identifier

---

<b>URL</b>	Uniform Resource Locator
<b>XML</b>	eXtensible Markup Language
<b>XSD</b>	XML Schema Definition

## 4. Introduction

The Technical Specification for the RESTful Network API for Device Capabilities contains the HTTP protocol binding based on the Parlay X Device Capabilities Web Services [3GPP 29.199-18] specification, using the REST architectural style. The specification provides resource definitions, the HTTP verbs applicable for each of these resources, and the element data structures, as well as support material including flow diagrams and examples using the various supported message body formats (i.e. XML, JSON, and application/x-www-form-urlencoded).

### 4.1 Version 1.0

The RESTful Network API for Device Capabilities V1.0 is a republication of the ParlayREST Device Capabilities API V1.0 [ParlayREST\_DeviceCapabilities] as part of the suite of OMA RESTful Network APIs.

Bug fixes and structural changes to fit that suite, but also functional changes have been applied.

Version 1.0 of the RESTful Network API for Device Capabilities keeps supporting the following operations:

- Retrieve device capabilities
- Create a subscription for device capabilities change notifications (for a single device or a group of devices)
- Send a notification on device capabilities changes to a subscribed application
- Set device configuration (for a single device or a group of devices)
- Retrieve available configurations for a given device model
- Retrieve configuration history for a given device

The following new functionality has been introduced:

- Support for scope values used with authorization framework defined in [Autho4API\_10]
- Support for Anonymous Customer Reference (ACR) as an end user identifier
- Support for “acr:Authorization” as a reserved keyword in a resource URL variable that identifies an end user

## 5. Device Capabilities API definition

This section is organized to support a comprehensive understanding of the RESTful Device Capabilities API design. It specifies the definition of all resources, definition of all data structures, and definitions of all operations permitted on the specified resources.

The Device Capabilities API will allow applications to retrieve information about device capabilities and push device configuration to a device. Applications can subscribe to notifications of device capabilities changes for one or more devices. In addition, the Device Capabilities API provides applications with the ability to obtain the available device configurations for a given device model, as well as the configuration history for a given device.

Common data types, naming conventions, fault definitions and namespaces are defined in [REST\_NetAPI\_Common].

The remainder of this document is structured as follows:

Section 5 starts with a diagram representing the resources hierarchy, followed by a table listing all the resources (and their URL) used by this API, along with the data structure and the supported HTTP verbs (section 5.1). What follows are the data structures (section 5.2). A sample of typical use cases is included in section 5.3, described as high level flow diagrams.

Section 6 contains the detailed specification for each of the resources. Each such subsection defines the resource, the request URL variables that are common for all HTTP commands, the possible HTTP response codes, and the supported HTTP verbs. For each supported HTTP verb, a description of the functionality is provided, along with an example of a request and an example of a response. For each unsupported HTTP verb, the returned HTTP error status is specified, as well as what should be returned in the Allow header.

All examples in section 6 use XML as the format for the message body. Application/x-www-form-urlencoded examples are provided in Appendix C, while JSON examples are provided in Appendix C. Appendix B provides the Static Conformance Requirements (SCR).

Appendix E lists the Parlay X equivalent operation for each supported REST resource and method combination, where applicable.

Appendix F provides a list of all light-weight resources, where applicable.

Appendix G defines authorization aspects to control access to the resources defined in this specification.

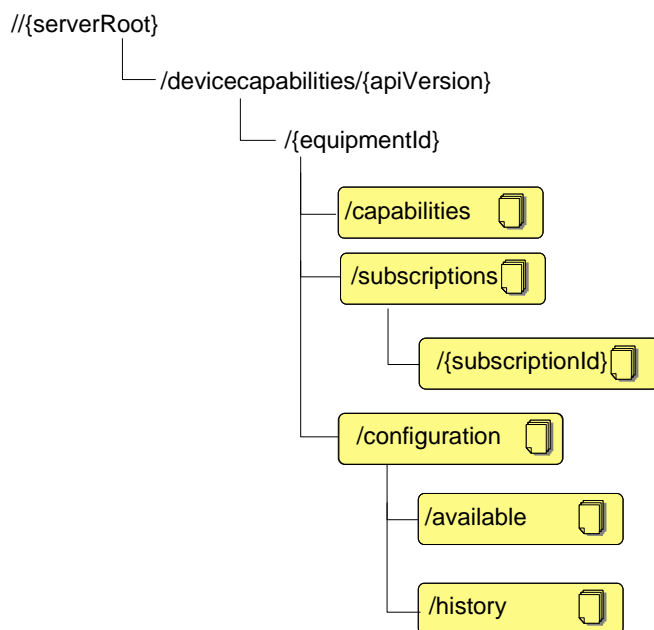
Note: Throughout this document client and application can be used interchangeably.

### 5.1 Resources Summary

This section summarizes all the resources used by the RESTful Device Capabilities API.

The "apiVersion" URL variable SHALL have the value "v1" to indicate that the API corresponds to this version of the specification. See [REST\_NetAPI\_Common] which specifies the semantics of this variable.

The figure below visualizes the resource structure defined by this specification. Note that those nodes in the resource tree which have associated HTTP methods defined in this specification are depicted by solid boxes.



**Figure 1 Resource structure defined by this specification**

The following tables give a detailed overview of the resources defined in this specification, the data type of their representation and the allowed HTTP methods.

**Purpose: to allow client to retrieve device capabilities**

Resource	URL Base URL: http://{serverRoot}/devicecapabilities/{apiVersion}	Data Structures	HTTP verbs			
			GET	PUT	POST	DELETE
Device capabilities	/{equipmentId}/capabilities	DeviceCapabilities	retrieve capabilities of the identified device	no	no	no

**Purpose: to allow client to manage subscriptions for device capabilities changes notifications**

Resource	URL Base URL: http://{serverRoot}/devicecapabilities/{apiVersion}	Data Structures	HTTP verbs			
			GET	PUT	POST	DELETE
Subscriptions for device capabilities changes notifications	/{equipmentId}/subscriptions	DeviceCapabilitiesChangeSubscriptionList  common:ResourceReference (optional alternative for POST response)  DeviceCapabilitiesChangeSubscription (used for POST)	retrieve the list of device capabilities changes subscriptions for the identified device(s)	no	create a new device capabilities changes subscription for the identified device(s)	no
Individual subscription for device capabilities changes notifications	/{equipmentId}/subscriptions/{subscriptionId}	DeviceCapabilitiesChangeSubscription	retrieve an individual device capabilities changes subscription for the identified device(s)	no	no	delete an individual device capabilities changes subscription for the identified device(s)

**Purpose: to allow client to receive notifications about changes in device capabilities**

Resource	URL <specified by the client>	Data Structures	HTTP verbs			
			GET	PUT	POST	DELETE
Notification on device capabilities changes	<specified by the client when subscription is created or during provisioning process>	DeviceCapabilitiesNotification DeviceCapabilitiesCancellationNotification	no	no	send notifications to the application (changes, end of notifications, cancellation)	no

**Purpose: to allow client to configure a device, retrieve available configurations for a device and retrieve history of configurations**

Resource	URL Base URL: http://{serverRoot}/devicecapabilities/{apiVersion}	Data Structures	HTTP verbs			
			GET	PUT	POST	DELETE
Device configuration	/{equipmentId}/configuration	DeviceConfiguration	no	no	push a configuration to the identified device(s)	no
Available configurations	/{equipmentId}/configuration/available	DeviceConfigurationList	retrieve available configurations for the identified device	no	no	no
Configuration history	/{equipmentId}/configuration/history	DeviceConfigurationHistoryList	retrieve configuration history for the identified device.	no	no	no



## 5.2 Data Types

### 5.2.1 XML Namespaces

The namespace for the Device Capabilities data types is:

urn:oma:xml:rest:netapi:devicecapabilities:1

The 'xsd' namespace is used in the present document to refer to the XML Schema data types defined in XML Schema [XMLSchema1, XMLSchema2]. The 'common' namespace is used in the present document to refer to the data types defined in [REST\_NetAPI\_Common]. The use of the names 'xsd' and 'common' is not semantically significant.

The XML schema for the data structures defined in the section below is given in [REST\_SUP\_DeviceCapabilities].

Applications following the RESTful Network API for Device Capabilities V 1.0 specification SHALL use the namespace urn:oma:xml:rest:netapi:devicecapabilities:1.

Note: Server implementations can choose to also support the legacy namespace urn:oma:xml:rest:devicecapabilities:1 for the Device Capabilities data types, in order to allow backwards-compatibility with [ParlayREST\_DeviceCapabilities] applications. Use of this legacy namespace is deprecated and support is foreseen to be withdrawn in future versions of this specification. In messages sent from the server to the application, the legacy namespace is suggested to be used by the server if it was used by a legacy application in the corresponding request or subscription message.

### 5.2.2 Structures

The subsections of this section define the data structures used in the RESTful Device Capabilities API.

Some of the structures can be instantiated as so-called root elements.

For structures that contain elements which describe an equipment identifier, the statements in section 6 regarding 'tel', 'sip' and 'acr' URI schemes apply.

#### 5.2.2.1 Type: DeviceCapabilities

Device Capabilities for queries

Element	Type	Optional	Description
deviceId	xsd:string	No	A unique equipment identifier for the device (e.g. IMEI)
name	xsd:string	No	The name of the device/model
resourceURL	xsd:anyURI	No	Self referring URL
link	common:Link[0..unbounded]	Yes	Provided by the server, points to other resources that are in relationship with the current resource (e.g. rel="UserAgentProfileReference" (section 5.2.4)).

A root element named deviceCapabilities of type DeviceCapabilities is allowed in response bodies.

#### 5.2.2.2 Type: DeviceCapabilitiesNotification

Device capabilities for capabilities changes notification

Element	Type	Optional	Description
callbackData	xsd:string	Yes	The 'callbackData' element if it was passed by the application in the 'callbackReference' element when creating the subscription for notifications about device capabilities changes. See [REST_NetAPI_Common] for details.
changeNotificationEnd	xsd:boolean	Yes	Default: false. Set to true if it is a final notification about status change.
deviceAddress	xsd:anyURI	Yes	Address of a device (e.g. 'sip' URI, 'tel' URI, 'acr' URI) if the notification refers to an individual device. Not present if the notification applies to the entire subscription
deviceId	xsd:string	No	A unique equipment identifier for the device (e.g. IMEI)
link	common:Link[0..unbounded]	Yes	Provided by the server, points to other resources that are in relationship with the current resource (e.g. rel="DeviceCapabilitiesChangeSubscription", rel="DeviceConfiguration" (section 5.2.4)).

A root element named deviceCapabilitiesNotification of type DeviceCapabilitiesNotification is allowed in request and/or response bodies.

### 5.2.2.3 Type: DeviceCapabilitiesCancellationNotification

Cancellation for device capabilities notification

Element	Type	Optional	Description
callbackData	xsd:string	Yes	The 'callbackData' element if it was passed by the application in the 'callbackReference' element when creating the subscription for notifications about device capabilities changes. See [REST_NetAPI_Common] for details.
deviceAddress	xsd:anyURI	Yes	Address of a device (e.g. 'sip' URI, 'tel' URI, 'acr' URI) if the notification refers to an individual device. Not present if the notification applies to the entire subscription
reason	common:ServiceError	No	Reason for the notification being discontinued
link	common:Link [0..unbounded]	Yes	Provided by the server, points to other resources that are in relationship with the current resource (e.g. rel="DeviceCapabilitiesChangeSubscription", rel="DeviceConfiguration" (section 5.2.4)).

A root element named deviceCapabilitiesCancellationNotification of type DeviceCapabilitiesCancellationNotification is allowed in request and/or response bodies.

### 5.2.2.4 Type: ConfigurationHistoryEntry

Device configuration history for query

Element	Type	Optional	Description
deviceConfiguration	DeviceConfiguration	No	A device configuration
timestamp	xsd:dateTime	No	The date/time when the configuration was sent to the device address.
link	common:Link[0..unbounded]	Yes	Provided by the server, points to other resources that are in relationship with the current resource.

### 5.2.2.5 Type: DeviceConfigurationHistoryList

Device configurations history list for query

Element	Type	Optional	Description
configurationHistoryEntry	ConfigurationHistoryEntry [0..unbounded]	Yes	A list of configuration history elements
resourceURL	xsd:anyURI	No	Self referring URL

A root element named deviceConfigurationHistoryList of type DeviceConfigurationHistoryList is allowed in response bodies.

### 5.2.2.6 Type: DeviceConfiguration

Device configuration for query or pushing

Element	Type	Optional	Description
configurationId	xsd:string	No	A unique identifier for the configuration, defined by the server
name	xsd:string	No	The name of the configuration
description	xsd:string	No	The description of the configuration
link	common:Link[0..unbounded]	Yes	Provided by the server, points to other resources that are in relationship with the current resource (e.g. rel="ConfigurationProfileReference" (section 5.2.4)). A link to a "ConfigurationProfileReference" MUST be provided by the server, in a response to a GET request for available configurations from the application. In this release, there is no support for an application to provide to the server a link to a "ConfigurationProfileReference", in a POST request to push a device configuration.

A root element named deviceConfiguration of type DeviceConfiguration is allowed in request and/or response bodies.

### 5.2.2.7 Type: DeviceConfigurationList

Available device configurations list for query

Element	Type	Optional	Description
deviceConfiguration	DeviceConfiguration[0..unbounded]	Yes	A list of available device configurations
resourceURL	xsd:anyURI	No	Self referring URL

A root element named deviceConfigurationList of type DeviceConfigurationList is allowed in response bodies.

### 5.2.2.8 Type: DeviceCapabilitiesChangeSubscriptionList

List of subscriptions to change notifications

Element	Type	Optional	Description
deviceCapabilitiesChangeSubscription	DeviceCapabilitiesChangeSubscription [0..unbounded]	Yes	A list of available device capabilities change subscriptions
resourceURL	xsd:anyURI	No	Self referring URL

A root element named deviceCapabilitiesChangeSubscriptionList of type DeviceCapabilitiesChangeSubscriptionList is allowed in response bodies.

### 5.2.2.9 Type: DeviceCapabilitiesChangeSubscription

Subscription to change notifications

Element	Type	Optional	Description
timeCreated	xsd:dateTime	No	The date/time when the subscription was created
callbackReference	common:CallbackReference	No	Client's notification URL and OPTIONAL callbackData
clientCorrelator	xsd:string	Yes	<p>A correlator that the client can use to tag this particular resource representation during a request to create a resource on the server.</p> <p>This field SHOULD be present. Note: this allows the client to recover from communication failures during resource creation and therefore avoids re-sending the message in such situations.</p> <p>In case the field is present, the server SHALL not alter its value, and SHALL provide it as part of the representation of this resource. In case the field is not present, the server SHALL NOT generate it.</p>
resourceURL	xsd:anyURI	Yes	Self referring URL. The resourceURL SHALL NOT be included in POST requests by the client, but MUST be included in POST requests representing notifications by the server to the client, when a complete representation of the resource is embedded in the notification. The resourceURL MUST also be included in responses to any HTTP method that returns an entity body, and in PUT requests.
link	common:Link[0..unbounded]	Yes	Provided by the server, points to other resources that are in relationship with the current resource (e.g. rel="DeviceConfiguration" (section 5.2.4)).

A root element named deviceCapabilitiesChangeSubscription of type DeviceCapabilitiesChangeSubscription is allowed in request and/or response bodies.

Note that the clientCorrelator is used for purposes of error recovery as specified in [REST\_NetAPI\_Common], and internal client purposes. The server is NOT REQUIRED to use the clientCorrelator value in any form in the creation of the URL of the resource. The document [REST\_NetAPI\_Common] provides a recommendation regarding the generation of the value of this field.

### 5.2.3 Enumerations

This section is empty in this version of the specification.

### 5.2.4 Values of the Link “rel” attribute

The “rel” attribute of the Link element is a free string set by the server implementation, to indicate a relationship between the current resource and an external resource. The following are possible strings (list is non-exhaustive, and can be extended):

- ConfigurationProfileReference
- DeviceCapabilitiesChangeSubscription
- DeviceCapabilities
- DeviceConfiguration
- UserAgentProfileReference

These values indicate the kind of resource that the link points to. The value “UserAgentProfileReference” indicates that the accompanying Link href attribute refers to a document containing a device’s User Agent Profile (e.g. <http://.../deviceprofiles/N6230ir200.xml>). The value “ConfigurationProfileReference” indicates that the accompanying Link href attribute refers to a document containing a device’s configuration profile (e.g. <http://.../deviceconfig/12345.xml>).

## 5.3 Sequence Diagrams

The following sub-sections describe the resources, methods and steps involved in typical scenarios.

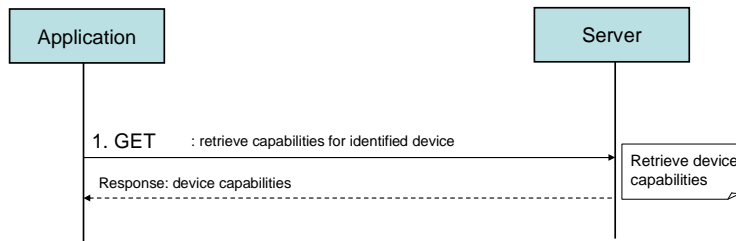
### 5.3.1 Retrieving device capabilities information

This figure below shows a scenario for retrieving device capabilities information..

The resource:

- To retrieve device capabilities for a single device, read the resource below with equipmentId identifying the targeted device

**`http://{serverRoot}/devicecapabilities/{apiVersion}/{equipmentId}/capabilities`**



**Figure 2 Retrieving device capabilities**

Outline of flow:

1. An application requests the device capabilities for a device identified by equipmentId (e.g. device address) using GET and receives the device’s capabilities information (including deviceId, model name, link to capabilities - e.g. UAProfile)

## 5.3.2 Subscribing for and notifying on device(s) capabilities

This figure below shows a scenario to create and use a subscription for device capabilities changes notifications.

The notification URL passed by the client during the subscription step can be a Client-side Notification URL, or a Server-side Notification URL. Refer to [REST\_NetAPI\_NotificationChannel] for sequence flows illustrating the creation of a Notification Channel and obtaining a Server-side Notification URL on the server-side, and its use by the client via Long Polling.

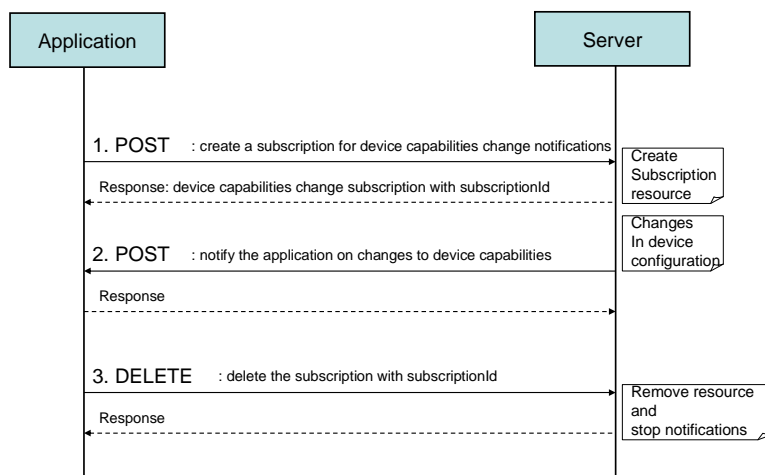
The resources:

- To start subscription to notifications about changes to device capabilities for identified device(s), create new resource under

**http://{serverRoot}/devicecapabilities/{apiVersion}/{equipmentId}/subscriptions**

- To delete an individual subscription, use the resource

**http://{serverRoot}/devicecapabilities/{apiVersion}/{equipmentId}/subscriptions/{subscriptionId}**



**Figure 3 Subscribing for change notifications and handling notifications**

1. An application creates a new notification subscription for changes in the device capabilities of the device(s) identified by {equipmentId} by using POST and receives the resulting resource URL containing the subscriptionId.
2. When the device capabilities for the identified device(s) change the server notifies the application using POST to the application supplied notifyURL.
3. An application deletes a subscription for changes in the device capabilities of the identified device(s) and stops notifications for the application by using DELETE to resource URL containing the subscriptionId.

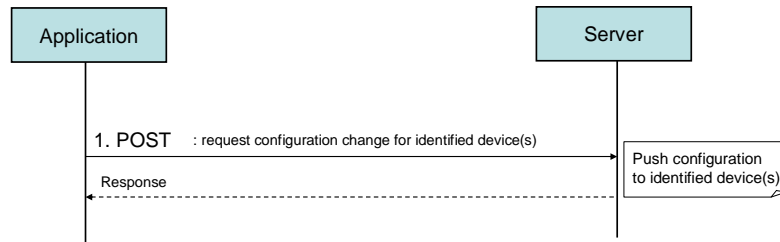
## 5.3.3 Pushing a configuration to device(s)

This figure below shows a scenario for pushing a configuration to an identified device or group of devices.

The resource:

- To push a configuration to identified device(s), POST the configuration to the resource below with equipmentId identifying the targeted device(s)

**http://{serverRoot}/devicecapabilities/{apiVersion}/{equipmentId}/configuration**



**Figure 4 Push configuration to device(s)**

Outline of flow:

1. An application requests that a new configuration is pushed to device(s) identified by equipmentId using POST.

### 5.3.4 Retrieving available configurations and history of configurations

This figure below shows a scenario to retrieve available configurations or history of configurations for device(s) identified by {equipmentId}.

The resources:

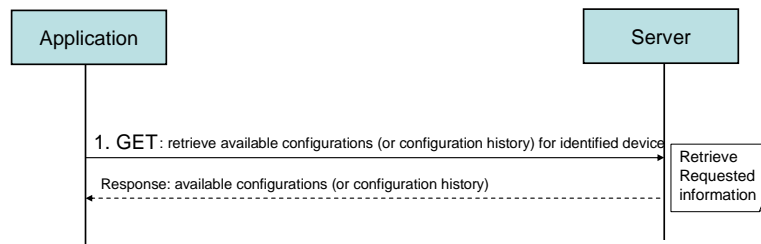
- To retrieve available configurations for an identified device, read the resource below with equipmentId identifying the targeted device

**http://{serverRoot}/devicecapabilities/{apiVersion}/{equipmentId}/configuration/available**

- To retrieve configuration history for an identified device, read the resource below with equipmentId identifying the targeted device

**http://{serverRoot}/devicecapabilities/{apiVersion}/{equipmentId}/configuration/history**





**Figure 5 Retrieving available configurations or configuration history for targeted device**

1. An application requests/receives available configurations or configurations history for a device identified by equipmentId using GET.

## 6. Detailed specification of the resources

The following applies to all resources defined in this specification regardless of the representation format (i.e. XML, JSON, application/x-www-form-urlencoded):

- Reserved characters in URL variables (parts of a URL denoted below by a name in curly brackets) **MUST** be percent-encoded according to [RFC3986]. Note that this always applies, no matter whether the URL is used as a Request URL or inside the representation of a resource (such as in “resourceURL” and “link” elements).
- If an equipment identifier of type anyURI is in the form of an MSISDN, it **MUST** be defined as a global number according to [RFC3966] (e.g. tel:+19585550100). The use of characters other than digits and the leading “+” sign **SHOULD** be avoided in order to ensure uniqueness of the resource URL. This applies regardless of whether the user identifier appears in a URL variable or in a parameter in the body of an HTTP message.
- If an equipment identifier of type anyURI is in the form of a SIP URI, it **MUST** be defined according to [RFC3261].
- If an equipment identifier of type anyURI is in the form of an Anonymous Customer Reference (ACR), it **MUST** be defined according to [IETF\_ACR\_draft], i.e. it **MUST** include the protocol prefix 'acr:' followed by the ACR.
  - The ACR ‘Authorization’ is a supported reserved keyword, and **MUST NOT** be assigned as an ACR to any particular end user. See G.1.2 for details regarding the use of this reserved keyword.
- For requests and responses that have a body, the following applies: in the requests received, the server **SHALL** support JSON and XML encoding of the parameters in the body, and **MAY** support application/x-www-form-urlencoded parameters in the body. The Server **SHALL** return either JSON or XML encoded parameters in the response body, according to the result of the content type negotiation as specified in [REST\_NetAPI\_Common]. In notifications to the Client, the server **SHALL** use either XML or JSON encoding, depending on which format the client has specified in the related subscription. The generation and handling of the JSON representations **SHALL** follow the rules for JSON encoding in HTTP Requests/Responses as specified in [REST\_NetAPI\_Common].

### 6.1 Resource: Device capabilities

The resource used is: **http://{serverRoot}/devicecapabilities/{apiVersion}/{equipmentId}/capabilities**

The resource is intended to retrieve the capabilities of an identified device. The device is identified by its equipmentId. In the current release, reading configuration for group of devices is not supported. If a group URI is provided, a service provider policy will determine whether the request can or cannot be supported. In case the request cannot be supported, a fault (POL0006) will be returned to the application.

The information returned is the device capabilities consisting of a unique ID for the device type, the name/model of the device and a link to the User Agent Profile XML file for the device.

#### 6.1.1 Request URL variables

The following request URL variables are common for all HTTP commands:

Name	Description
serverRoot	server base url: hostname+port+base path. Port and base path are OPTIONAL. Example: example.com/exampleAPI
apiVersion	version of the API client wants to use. The value of this variable is defined in section 5.1.
equipmentId	equipment identifier. Examples: tel:+19585550100, acr:pseudonym123

See section 6 for a statement on the escaping of reserved characters in URL variables.

#### 6.1.2 Response Codes and Error Handling

For HTTP response codes, see [REST\_NetAPI\_Common].

For Policy Exception and Service Exception fault codes applicable to Device Capabilities, see [3GPP 29.199-18].

## 6.1.3 GET

This operation is used to retrieve capabilities of the identified device.

### 6.1.3.1 Example: Retrieve device capabilities

(Informative)

#### 6.1.3.1.1 Request

```
GET /exampleAPI/devicecapabilities/v1/tel%3A%2B19585550100/capabilities HTTP/1.1
Accept: application/xml
Host: example.com
```

#### 6.1.3.1.2 Response

```
HTTP/1.1 200 OK
Date: Thu, 04 Jun 2010 02:52:00 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<dc:deviceCapabilities xmlns:dc="urn:oma:xml:rest:netapi:devicecapabilities:1">
  <deviceId>123456789012345</deviceId>
  <name>devname123</name>
  <resourceURL>http://example.com/exampleAPI/devicecapabilities/v1/tel%3A%2B19585550100/capabilities</resourceURL>
  <link rel="UserAgentProfileReference"
href="http://example.com/exampleconfigurations/exampledeviceprofiles/A1234xyz123.xml"/>
</dc:deviceCapabilities>
```

## 6.1.4 PUT

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET' field in the response as per section 14.7 of [RFC 2616].

## 6.1.5 POST

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET' field in the response as per section 14.7 of [RFC 2616].

## 6.1.6 DELETE

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET' field in the response as per section 14.7 of [RFC 2616].

## 6.2 Resource: Subscriptions for device capabilities changes notifications

The resource used is: <http://{serverRoot}/devicecapabilities/{apiVersion}/{equipmentId}/subscriptions>

The application sets a notification trigger for changes in the equipment identified (device or group of devices). In case a single device is targeted the application passes deviceAddress (e.g. the device URI) for the equipmentId. In case a group of devices is targeted the application passes a deviceGroupId (e.g. a group URI).

This resource can be used in conjunction with a Client-side Notification URL, or in conjunction with a Server-side Notification URL. In this latter case, the application MUST first create a Notification Channel (see [REST\_NetAPI\_NotificationChannel]) before creating a subscription.

Note: in this release, a single subscription per client is supported. The reason for the resource structure is to provide for consistency with other APIs supporting subscriptions, and allow extensions in future releases for multiple subscriptions per client.

## 6.2.1 Request URL variables

The following request URL variables are common for all HTTP commands:

Name	Description
serverRoot	server base url: hostname+port+base path. Port and base path are OPTIONAL. Example: example.com/exampleAPI
apiVersion	version of the API client wants to use. The value of this variable is defined in section 5.1.
equipmentId	equipment identifier. Examples: tel:+19585550100, acr:pseudonym123

See section 6 for a statement on the escaping of reserved characters in URL variables.

## 6.2.2 Response Codes and Error Handling

For HTTP response codes, see [REST\_NetAPI\_Common].

For Policy Exception and Service Exception fault codes applicable to Device Capabilities, see [3GPP 29.199-18].

## 6.2.3 GET

This operation is used to retrieve the list of device capabilities changes subscriptions to notifications for the identified device(s).

In this release, a single subscription per application is supported.

### 6.2.3.1 Example: Retrieve list of subscriptions (Informative)

This example shows also an alternative way to indicate desired content type in response from the server, by using URL query parameter “?resFormat” which is described in [REST\_NetAPI\_Common].

#### 6.2.3.1.1 Request

```
GET /exampleAPI/devicecapabilities/v1/tel%3A%2B19585550100/subscriptions?resFormat=XML HTTP/1.1
Accept: application/xml
Host: example.com
```

#### 6.2.3.1.2 Response

```
HTTP/1.1 200 OK
Date: Thu, 04 Jun 2010 02:52:00 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<dc:deviceCapabilitiesChangeSubscriptionList xmlns:dc="urn:oma:xml:rest:netapi:devicecapabilities:1">
  <deviceCapabilitiesChangeSubscription>
    <timeCreated>2010-03-21T13:23:21Z</timeCreated>
    <callbackReference>
      <notifyURL>http://application.example.com/notifications/CapabilitiesChangeNotification</notifyURL>
```

```

    <callbackData>12345</callbackData>
  </callbackReference>
  <clientCorrelator>54321</clientCorrelator>
  <resourceURL>http://example.com/exampleAPI/devicecapabilities/v1/tel%3A%2B19585550100/subscriptions/sub123</resourceURL>
</deviceCapabilitiesChangeSubscription>
  <resourceURL>http://example.com/exampleAPI/devicecapabilities/v1/tel%3A%2B19585550100/subscriptions</resourceURL>
</dc:deviceCapabilitiesChangeSubscriptionList>

```

## 6.2.4 PUT

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET, POST' field in the response as per section 14.7 of [RFC 2616].

## 6.2.5 POST

This operation is used to create a new device capabilities changes subscription to notifications for the identified device(s).

The notifyURL in the callbackReference either contains the Client-side Notification URL (as defined by the client) or the Server-side Notification URL (as obtained during the creation of the Notification Channel [REST\_NetAPI\_NotificationChannel]).

### 6.2.5.1 Example 1: Subscribe to device capabilities changes notifications for single device using 'tel' URI (Informative)

This example also illustrates how to indicate the desired notification format.

#### 6.2.5.1.1 Request

```

POST /exampleAPI/devicecapabilities/v1/tel%3A%2B19585550100/subscriptions HTTP/1.1
Accept: application/xml
Host: example.com
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<dc:deviceCapabilitiesChangeSubscription xmlns:dc="urn:oma:xml:rest:netapi:devicecapabilities:1">
  <timeCreated>2010-03-21T13:23:21Z</timeCreated>
  <callbackReference>
    <notifyURL>http://application.example.com/notifications/CapabilitiesChangeNotification</notifyURL>
    <callbackData>12345</callbackData>
    <notificationFormat>JSON</notificationFormat>
  </callbackReference>
  <clientCorrelator>54321</clientCorrelator>
</dc:deviceCapabilitiesChangeSubscription>

```

#### 6.2.5.1.2 Response

```

HTTP/1.1 201 Created
Date: Thu, 04 Jun 2010 02:52:00 GMT

```

```

Location: http://example.com/exampleAPI/devicecapabilities/v1/ tel%3A%2B19585550100/subscriptions/sub123
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<dc:deviceCapabilitiesChangeSubscription xmlns:dc="urn:oma:xml:rest:netapi:devicecapabilities:1">
  <timeCreated>2010-03-21T13:23:21Z</timeCreated>
  <callbackReference>
    <notifyURL>http://application.example.com/notifications/CapabilitiesChangeNotification</notifyURL>
    <callbackData>12345</callbackData>
    <notificationFormat>JSON</notificationFormat>
  </callbackReference>
  <clientCorrelator>54321</clientCorrelator>
  <resourceURL>
http://example.com/exampleAPI/devicecapabilities/v1/tel%3A%2B19585550100/subscriptions/sub123</resourceURL>
</dc:deviceCapabilitiesChangeSubscription>

```

### 6.2.5.2 Example 2: Subscribe to device capabilities changes notifications for single device using 'acr' URI (Informative)

This example also illustrates how to indicate the desired notification format.

#### 6.2.5.2.1 Request

```

POST /exampleAPI/devicecapabilities/v1/acr%3A%2B19585550100/subscriptions HTTP/1.1
Accept: application/xml
Host: example.com
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<dc:deviceCapabilitiesChangeSubscription xmlns:dc="urn:oma:xml:rest:netapi:devicecapabilities:1">
  <timeCreated>2010-03-21T13:23:21Z</timeCreated>
  <callbackReference>
    <notifyURL>http://application.example.com/notifications/CapabilitiesChangeNotification</notifyURL>
    <callbackData>12345</callbackData>
    <notificationFormat>JSON</notificationFormat>
  </callbackReference>
  <clientCorrelator>54321</clientCorrelator>
</dc:deviceCapabilitiesChangeSubscription>

```

#### 6.2.5.2.2 Response

```

HTTP/1.1 201 Created
Date: Thu, 04 Jun 2010 02:52:00 GMT
Location: http://example.com/exampleAPI/devicecapabilities/v1/ acr%3A%2B19585550100/subscriptions/sub123
Content-Type: application/xml
Content-Length: nnnn

```

```
<?xml version="1.0" encoding="UTF-8"?>
<dc:deviceCapabilitiesChangeSubscription xmlns:dc="urn:oma:xml:rest:netapi:devicecapabilities:1">
  <timeCreated>2010-03-21T13:23:21Z</timeCreated>
  <callbackReference>
    <notifyURL>http://application.example.com/notifications/CapabilitiesChangeNotification</notifyURL>
    <callbackData>12345</callbackData>
    <notificationFormat>JSON</notificationFormat>
  </callbackReference>
  <clientCorrelator>54321</clientCorrelator>
  <resourceURL>
    http://example.com/exampleAPI/devicecapabilities/v1/acr%3Apsedonym123/subscriptions/sub123</resourceURL>
  </dc:deviceCapabilitiesChangeSubscription>
```

### 6.2.5.3 Example 3: Subscribe to device capabilities changes notifications for group of devices (Informative)

#### 6.2.5.3.1 Request

POST /exampleAPI/devicecapabilities/v1/GRP19585550100/subscriptions HTTP/1.1

Accept: application/xml

Host: example.com

Content-Type: application/xml

Content-Length: nnnn

```
<?xml version="1.0" encoding="UTF-8"?>
<dc:deviceCapabilitiesChangeSubscription xmlns:dc="urn:oma:xml:rest:netapi:devicecapabilities:1">
  <timeCreated>2010-03-21T13:23:21Z</timeCreated>
  <callbackReference>
    <notifyURL>http://application.example.com/notifications/CapabilitiesChangeNotification</notifyURL>
    <callbackData>12345</callbackData>
  </callbackReference>
  <clientCorrelator>54321</clientCorrelator>
</dc:deviceCapabilitiesChangeSubscription>
```

#### 6.2.5.3.2 Response

HTTP/1.1 201 Created

Date: Thu, 04 Jun 2010 02:52:00 GMT

Location: http://example.com/exampleAPI/devicecapabilities/v1/GRP19585550100/subscriptions/sub123

Content-Type: application/xml

Content-Length: nnnn

```
<?xml version="1.0" encoding="UTF-8"?>
<dc:deviceCapabilitiesChangeSubscription xmlns:dc="urn:oma:xml:rest:netapi:devicecapabilities:1">
```

```

<timeCreated>2010-03-21T13:23:21Z</timeCreated>
<callbackReference>
  <notifyURL>http://application.example.com/notifications/CapabilitiesChangeNotification</notifyURL>
  <callbackData>12345</callbackData>
</callbackReference>
<clientCorrelator>54321</clientCorrelator>
<resourceURL> http://example.com/exampleAPI/devicecapabilities/v1/GRP19585550100/subscriptions/sub123</resourceURL>
</dc:deviceCapabilitiesChangeSubscription>

```

### 6.2.5.4 Example 4: Subscribe to device capabilities changes notifications for single device (with resource reference instead of full resource representation in the response body) (Informative)

#### 6.2.5.4.1 Request

```

POST /exampleAPI/devicecapabilities/v1/tel%3A%2B19585550100/subscriptions HTTP/1.1
Accept: application/xml
Host: example.com
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<dc:deviceCapabilitiesChangeSubscription xmlns:dc="urn:oma:xml:rest:netapi:devicecapabilities:1">
  <timeCreated>2010-03-21T13:23:21Z</timeCreated>
  <callbackReference>
    <notifyURL>http://application.example.com/notifications/CapabilitiesChangeNotification</notifyURL>
    <callbackData>12345</callbackData>
  </callbackReference>
  <clientCorrelator>54321</clientCorrelator>
</dc:deviceCapabilitiesChangeSubscription>

```

#### 6.2.5.4.2 Response

```

HTTP/1.1 201 Created
Date: Thu, 04 Jun 2010 02:52:00 GMT
Location: http://example.com/exampleAPI/devicecapabilities/v1/tel%3A%2B19585550100/subscriptions/sub123
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<common:resourceReference xmlns:common="urn:oma:xml:rest:netapi:common:1">
  <resourceURL>http://example.com/exampleAPI/devicecapabilities/v1/tel%3A%2B19585550100/subscriptions/sub123</resourceURL>
</common:resourceReference>

```

## 6.2.6 DELETE

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET, POST' field in the response as per section 14.7 of [RFC 2616].



## 6.3 Resource: Individual subscription for device capabilities changes notifications

The resource used is: `http://{serverRoot}/devicecapabilities/{apiVersion}/{equipmentId}/subscriptions/{subscriptionId}`

The application can retrieve or delete a notification trigger for changes in the equipment identified (device or group of devices). In case a single device is targeted the application passes deviceAddress (e.g. the device URI) for the equipmentId. In case a group of devices is targeted the application passes a deviceGroupId (e.g. a group URI).

In this release, a single subscription per application is supported. The reason for the resource structure supporting a list of subscriptions is to provide consistency with other APIs supporting subscriptions, and allow extensions in future releases for multiple subscriptions per application.

### 6.3.1 Request URL variables

The following request URL variables are common for all HTTP commands:

Name	Description
serverRoot	server base url: hostname+port+base path. Port and base path are OPTIONAL. Example: example.com/exampleAPI
apiVersion	version of the API client wants to use. The value of this variable is defined in section 5.1.
equipmentId	equipment identifier. Examples: tel:+19585550100, acr:pseudonym123
subscriptionId	subscription identifier

See section 6 for a statement on the escaping of reserved characters in URL variables.

### 6.3.2 Response Codes and Error Handling

For HTTP response codes, see [REST\_NetAPI\_Common].

For Policy Exception and Service Exception fault codes applicable to Device Capabilities, see [3GPP 29.199-18].

### 6.3.3 GET

This operation is used to retrieve an individual device capabilities changes subscription to notifications for a given device or group of devices.

#### 6.3.3.1 Example: Retrieve subscription (Informative)

##### 6.3.3.1.1 Request

```
GET /exampleAPI/devicecapabilities/v1/tel%3A%2B19585550100/subscriptions/sub123 HTTP/1.1
Accept: application/xml
Host: example.com
```

##### 6.3.3.1.2 Response

```
HTTP/1.1 200 OK
Date: Thu, 04 Jun 2010 02:52:00 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
```

```
<dc:deviceCapabilitiesChangeSubscription xmlns:dc="urn:oma:xml:rest:netapi:devicecapabilities:1">
  <timeCreated>2010-03-21T13:23:21Z</timeCreated>
  <callbackReference>
    <notifyURL>http://application.example.com/notifications/CapabilitiesChangeNotification</notifyURL>
    <callbackData>12345</callbackData>
  </callbackReference>
  <clientCorrelator>54321</clientCorrelator>
<resourceURL>http://example.com/exampleAPI/devicecapabilities/v1/tel%3A%2B19585550100/subscriptions/sub123</resourceURL>
</dc:deviceCapabilitiesChangeSubscription>
```

### 6.3.4 PUT

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET, DELETE' field in the response as per section 14.7 of [RFC 2616].

### 6.3.5 POST

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET, DELETE' field in the response as per section 14.7 of [RFC 2616].

### 6.3.6 DELETE

This operation is used to delete an individual device capabilities changes subscription to notifications for a given device or group of devices.

#### 6.3.6.1 Example: Delete subscription

(Informative)

##### 6.3.6.1.1 Request

```
DELETE /exampleAPI/devicecapabilities/v1/tel%3A%2B19585550100/subscriptions/sub123 HTTP/1.1
Accept: application/xml
Host: example.com
```

##### 6.3.6.1.2 Response

```
HTTP/1.1 204 No Content
Date: Thu, 04 Jun 2010 02:52:00 GMT
```

## 6.4 Resource: Notification on device capabilities changes

This resource is a callback URL provided by the client for notification about device capabilities changes. The RESTful Device Capabilities API does not make any assumption about the structure of this URL. If this URL is a Client-side Notification URL, the server will POST notifications directly to it. If this URL is a Server-side Notification URL, the server uses it to determine the address of the Notification Server to which the notifications will subsequently be POSTed. The way the server determines the address of the Notification Server is out of scope of this specification.

Note: In the case when the client has set up a Notification Channel in order to use Long Polling to obtain the notifications, in order to retrieve the notifications, the client needs to use the Long Polling mechanism described in [REST\_NetAPI\_NotificationChannel], instead of the mechanism described below in section 6.4.5.

### 6.4.1 Request URL variables

Client provided.

## 6.4.2 Response Codes and Error Handling

For HTTP response codes, see [REST\_NetAPI\_Common].

### 6.4.3 GET

Method not allowed by the resource. The returned HTTP error status is 405. The client should also include the 'Allow: POST' field in the response as per section 14.7 of [RFC 2616].

### 6.4.4 PUT

Method not allowed by the resource. The returned HTTP error status is 405. The client should also include the 'Allow: POST' field in the response as per section 14.7 of [RFC 2616].

### 6.4.5 POST

This operation is used by the server to send notifications to the application. There are three types of notifications:

- 1) normal change – a new deviceId is delivered as part of the structure
- 2) end of notifications - the notifications have ended for this subscription. This message will be delivered when the duration for notifications has been completed. This message will not be delivered in the case of an error ending the notifications or deliberate ending of the notification by client.
- 3) error – indicates cancelling the notifications with a reason for cancellation

#### 6.4.5.1 Example 1: Device capabilities changes notification (Informative)

##### 6.4.5.1.1 Request

```
POST /notifications/CapabilitiesChangeNotification HTTP/1.1
Accept: application/xml
Host: application.example.com
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<dc:deviceCapabilitiesNotification xmlns:dc="urn:oma:xml:rest:netapi:devicecapabilities:1">
  <callbackData>12345</callbackData>
  <changeNotificationEnd>false</changeNotificationEnd>
  <deviceAddress>tel:+19585550100</deviceAddress>
  <deviceId>123456789012345</deviceId>
  <link rel="DeviceCapabilitiesChangeSubscription" href=
"http://example.com/exampleAPI/devicecapabilities/v1/tel%3A%2B19585550100/subscriptions/sub123"/>
  <link rel="DeviceCapabilities"
href="http://example.com/exampleAPI/devicecapabilities/v1/tel%3A%2B19585550100/capabilities"/>
  <link rel="DeviceConfiguration"
href="http://example.com/exampleAPI/devicecapabilities/v1/tel%3A%2B19585550100/configuration"/>
</dc:deviceCapabilitiesNotification>
```

##### 6.4.5.1.2 Response

```
HTTP/1.1 204 No Content
Date: Thu, 04 Jun 2010 02:52:00 GMT
```

## 6.4.5.2 Example 2: End of changes notification

(Informative)

### 6.4.5.2.1 Request

```
POST /notifications/CapabilitiesChangeNotification HTTP/1.1
```

```
Accept : application/xml
```

```
Host: application.example.com
```

```
Content-Type: application/xml
```

```
Content-Length: nnnn
```

```
<?xml version="1.0" encoding="UTF-8"?>
<dc:deviceCapabilitiesNotification xmlns:dc="urn:oma:xml:rest:netapi:devicecapabilities:1">
  <callbackData>12345</callbackData>
  <changeNotificationEnd>true</changeNotificationEnd>
  <deviceAddress>tel:+19585550100</deviceAddress>
  <deviceId>123456789012345</deviceId>
  <link rel="DeviceCapabilitiesChangeSubscription" href=
"http://example.com/exampleAPI/devicecapabilities/v1/tel%3A%2B19585550100/subscriptions/sub123"/>
  <link rel="DeviceCapabilities"
href="http://example.com/exampleAPI/devicecapabilities/v1/tel%3A%2B19585550100/capabilities"/>
  <link rel="DeviceConfiguration"
href="http://example.com/exampleAPI/devicecapabilities/v1/tel%3A%2B19585550100/configuration"/>
</dc:deviceCapabilitiesNotification>
```

### 6.4.5.2.2 Response

```
HTTP/1.1 204 No Content
```

```
Date: Thu, 04 Jun 2010 02:52:00 GMT
```

## 6.4.5.3 Example 3: Error notification (notification cancellation)

(Informative)

### 6.4.5.3.1 Request

```
POST /notifications/StatusNotification HTTP/1.1
```

```
Accept : application/xml
```

```
Host: application.example.com
```

```
Content-Type: application/xml
```

```
Content-Length: nnnn
```

```
<?xml version="1.0" encoding="UTF-8"?>
<dc:deviceCapabilitiesCancellationNotification xmlns:dc="urn:oma:xml:rest:netapi:devicecapabilities:1">
  <callbackData>12345</callbackData>
  <deviceAddress>tel:+19585550100</deviceAddress>
  <reason>
```

```

<messageId>SVC0001</messageId>
<text>A service error occurred. %1 %2</text>
<variables>Status information is not available for</variables>
<variables>tel:+19585550100</variables>
</reason>
<link rel="DeviceCapabilitiesChangeSubscription" href=
"http://example.com/exampleAPI/devicecapabilities/v1/tel%3A%2B19585550100/subscriptions/sub123"/>
<link rel="DeviceCapabilities" href="http://example.com/exampleAPI/devicecapabilities/v1/tel%3A%2B19585550100/capabilities"/>
<link rel="DeviceConfiguration"
href="http://example.com/exampleAPI/devicecapabilities/v1/tel%3A%2B19585550100/configuration"/>
</dc:deviceCapabilitiesCancellationNotification>

```

### 6.4.5.3.2 Response

HTTP/1.1 204 No Content  
Date: Thu, 04 Jun 2010 02:52:00 GMT

## 6.4.6 DELETE

Method not allowed by the resource. The returned HTTP error status is 405. The client should also include the ‘Allow: POST’ field in the response as per section 14.7 of [RFC 2616].

## 6.5 Resource: Device configuration

The resource used is: **http://{serverRoot}/devicecapabilities/{apiVersion}/{equipmentId}/configuration**

The resource supports pushing a configuration to a device or a group of devices.

### 6.5.1 Request URL variables

The following request URL variables are common for all HTTP commands:

Name	Description
serverRoot	server base url: hostname+port+base path. Port and base path are OPTIONAL. Example: example.com/exampleAPI
apiVersion	version of the API client wants to use. The value of this variable is defined in section 5.1.
equipmentId	equipment identifier. Examples: tel:+19585550100, acr:pseudonym123

See section 6 for a statement on the escaping of reserved characters in URL variables.

### 6.5.2 Response Codes and Error Handling

For HTTP response codes, see [REST\_NetAPI\_Common].

For Policy Exception and Service Exception fault codes applicable to Device Capabilities, see [3GPP 29.199-18].

### 6.5.3 GET

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the ‘Allow: POST’ field in the response as per section 14.7 of [RFC 2616].

## 6.5.4 PUT

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: POST' field in the response as per section 14.7 of [RFC 2616].

## 6.5.5 POST

This operation is used to push a configuration to a device or a group of devices. This operation results in a one-time push of a device configuration, and the resource is no longer accessible by the application for other operations.

### 6.5.5.1 Example 1: Push configuration to single device (Informative)

#### 6.5.5.1.1 Request

```
POST /exampleAPI/devicecapabilities/v1/tel%3A%2B19585550100/configuration HTTP/1.1
Accept: application/xml
Host: example.com
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<dc:deviceConfiguration xmlns:dc="urn:oma:xml:rest:netapi:devicecapabilities:1">
  <configurationId>config12345</configurationId>
  <name>configname12345</name>
  <description>configdescription12345</description>
</dc:deviceConfiguration>
```

#### 6.5.5.1.2 Response

```
HTTP/1.1 204 No Content
Date: Thu, 04 Jun 2010 02:52:00 GMT
```

### 6.5.5.2 Example 2: Push configuration to group of devices (Informative)

#### 6.5.5.2.1 Request

```
POST /exampleAPI/devicecapabilities/v1/GRP19585550100/configuration HTTP/1.1
Accept: application/xml
Host: example.com
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<dc:deviceConfiguration xmlns:dc="urn:oma:xml:rest:netapi:devicecapabilities:1">
  <configurationId>config12345</configurationId>
  <name>configname12345</name>
  <description>configdescription12345</description>
</dc:deviceConfiguration>
```

### 6.5.5.2.2 Response

```
HTTP/1.1 204 No Content
Date: Thu, 04 Jun 2010 02:52:00 GMT
```

## 6.5.6 DELETE

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: POST' field in the response as per section 14.7 of [RFC 2616].

## 6.6 Resource: Available configurations

The resource used is: **http://{serverRoot}/devicecapabilities/{apiVersion}/{equipmentId}/configuration/available**

The resource is used to retrieve configurations available for an identified device. In the current release, reading configurations available for a group of devices is not supported. If a group URI is provided, a service provider policy will determine whether the request can or cannot be supported. In case the request cannot be supported, a fault (POL0006) will be returned to the application.

The configurations have to be made available in advance by the gateway operator.

### 6.6.1 Request URL variables

The following request URL variables are common for all HTTP commands:

Name	Description
serverRoot	server base url: hostname+port+base path. Port and base path are OPTIONAL. Example: example.com/exampleAPI
apiVersion	version of the API client wants to use. The value of this variable is defined in section 5.1.
equipmentId	equipment identifier. Examples: tel:+19585550100, acr:pseudonym123

See section 6 for a statement on the escaping of reserved characters in URL variables.

### 6.6.2 Response Codes and Error Handling

For HTTP response codes, see [REST\_NetAPI\_Common].

For Policy Exception and Service Exception fault codes applicable to Device Capabilities, see [3GPP 29.199-18].

### 6.6.3 GET

This operation is used to retrieve available configurations for the identified device.

#### 6.6.3.1 Example: Retrieve available device configurations (Informative)

##### 6.6.3.1.1 Request

```
GET /exampleAPI/devicecapabilities/v1/tel%3A%2B19585550100/configuration/available HTTP/1.1
Accept: application/xml
Host: example.com
```

### 6.6.3.1.2 Response

```
HTTP/1.1 200 OK
Date: Thu, 04 Jun 2010 02:52:00 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
  <dc:deviceConfigurationList xmlns:dc="urn:oma:xml:rest:netapi:devicecapabilities:1">
    <deviceConfiguration>
      <configurationId>config12345</configurationId>
      <name>configname12345</name>
      <description>configdescription12345</description>
      <link rel="ConfigurationProfileReference" href="http://example.com/exampleconfigurations/exampledeviceconfig/12345.xml"/>
    </deviceConfiguration>
    <deviceConfiguration>
      <configurationId>config12346</configurationId>
      <name>configname12346</name>
      <description>configdescription12346</description>
      <link rel="ConfigurationProfileReference" href="http://example.com/exampleconfigurations/exampledeviceconfig/12346.xml"/>
    </deviceConfiguration>
    <deviceConfiguration>
      <configurationId>config12347</configurationId>
      <name>configname12347</name>
      <description>configdescription12347</description>
      <link rel="ConfigurationProfileReference" href="http://example.com/exampleconfigurations/exampledeviceconfig/12347.xml"/>
    </deviceConfiguration>
    <resourceURL>http://example.com/exampleAPI/devicecapabilities/v1/tel%3A%2B19585550100/configuration/available</resourceURL>
  </dc:deviceConfigurationList>
```

### 6.6.4 PUT

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET' field in the response as per section 14.7 of [RFC 2616].

### 6.6.5 POST

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET' field in the response as per section 14.7 of [RFC 2616].

### 6.6.6 DELETE

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET' field in the response as per section 14.7 of [RFC 2616].

## 6.7 Resource: Configuration history

The resource used is: **http://{serverRoot}/devicecapabilities/{apiVersion}/{equipmentId}/configuration/history**

The resource is used to retrieve configurations previously pushed to an identified device. In the current release, reading configurations previously pushed to a group of devices is not supported. If a group URI is provided, a service provider policy



will determine whether the request can or cannot be supported. In case the request cannot be supported, a fault (POL0006) will be returned to the application.

## 6.7.1 Request URL variables

The following request URL variables are common for all HTTP commands:

Name	Description
serverRoot	server base url: hostname+port+base path. Port and base path are OPTIONAL. Example: example.com/exampleAPI
apiVersion	version of the API client wants to use. The value of this variable is defined in section 5.1.
equipmentId	equipment identifier. Examples: tel:+19585550100, acr:pseudonym123

See section 6 for a statement on the escaping of reserved characters in URL variables.

## 6.7.2 Response Codes and Error Handling

For HTTP response codes, see [REST\_NetAPI\_Common].

For Policy Exception and Service Exception fault codes applicable to Device Capabilities, see [3GPP 29.199-18].

## 6.7.3 GET

This operation is used to retrieve configuration history for the identified device.

### 6.7.3.1 Example: Retrieve configuration history

(Informative)

#### 6.7.3.1.1 Request

```
GET /exampleAPI/devicecapabilities/v1/tel%3A%2B19585550100/configuration/history HTTP/1.1
Accept: application/xml
Host: example.com
```

#### 6.7.3.1.2 Response

```
HTTP/1.1 200 OK
Date: Thu, 04 Jun 2010 02:52:00 GMT
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
  <dc:deviceConfigurationHistoryList xmlns:dc="urn:oma:xml:rest:netapi:devicecapabilities:1">
    <configurationHistoryEntry>
      <deviceConfiguration>
        <configurationId>config12345</configurationId>
        <name>configname12345</name>
        <description>configdescription12345</description>
        <link rel="ConfigurationProfileReference" href="http://example.com/exampleconfigurations/exampledeviceconfig/12345.xml"/>
      </deviceConfiguration>
      <timestamp>2009-11-19T12:00:00</timestamp>
    </configurationHistoryEntry>
```

```
<configurationHistoryEntry>
  <deviceConfiguration>
    <configurationId>config12347</configurationId>
    <name>configname12347</name>
    <description>configdescription12347</description>
    <link rel="ConfigurationProfileReference" href="http://example.com/exampleconfigurations/exampledeviceconfig/12347.xml"/>
  </deviceConfiguration>
  <timestamp>2009-10-19T12:00:00</timestamp>
</configurationHistoryEntry>

<resourceURL>http://example.com/exampleAPI/devicecapabilities/v1/tel%3A%2B19585550100/configuration/history</resourceURL>
</dc:deviceConfigurationHistoryList>
```

#### 6.7.4 PUT

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET' field in the response as per section 14.7 of [RFC 2616].

#### 6.7.5 POST

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET' field in the response as per section 14.7 of [RFC 2616].

#### 6.7.6 DELETE

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET' field in the response as per section 14.7 of [RFC 2616].

## Appendix A. Change History (Informative)

### A.1 Approved Version History

Reference	Date	Description
n/a	n/a	No prior version –or- No previous version within OMA

### A.2 Draft/Candidate Version 1.0 History

Document Identifier	Date	Sections	Description
Draft Version: OMA-TS- REST_NetAPI_DeviceCapabilities- V1_0	27 Apr 2011	Many	Structural changes to fit the OMA RESTful Network API release. This version inherits the technical content of OMA-TS-ParlayREST_DeviceCapabilities-V1_0-20110111-C and applies changes according to ARC INP 30R01, 98R02, 155R01, 156R01 and 153R04.
	03 May 2011	Many	Apply ARC INP 172R01, 175R01, 186, 187R02 and OMA-ARC-REST-NetAPI-2011-0003R01.
	27 May 2011	Many	Implemented CR: OMA-ARC-REST-NetAPI-2011-0030R01-CR_DeviceCapabilities_TS_Evolution_to_agreed_new_template
	06 June 2011	Many	Implemented AI: REST-NetAPI-2011-A025
	27 July	Many	Implemented CR OMA-ARC-REST-NetAPI-2011-0158-CR_DeviceCapabilities_telURI_fixes, OMA-ARC-REST-NetAPI-2011-0161-CR_DeviceCapabilities_Notif_Channel_changes
	03 August 2011	Many	Implemented CR OMA-ARC-REST-NetAPI-2011-0175-CR_DeviceCapabilities_Appendix_C_link_to_section_6 and REST-NetAPI-2011-A090
	24 August 2011	Many	Implemented: OMA-ARC-REST-NetAPI-2011-01189-CR_DevCap_resourceURL
	01 September 2011	Many	Implemented CR: OMA-ARC-REST-NetAPI-2011-0202R02-CR_DevCap_ACR
	20 September 2011	Many	Implemented CR: OMA-ARC-REST-NetAPI-2011-0242R03-CR_DevCap_prep_for_CONR
	21 September 2011	A.2	Re-inserted a history row that got dropped when TS draft was updated on 01 September 2011
	24 October 2011	Many	Implemented CR: OMA-ARC-REST-NetAPI-2011-0313R01-CR_DeviceCapabilities_TS_CONR_resolutions
	08 November 2011	Many	Implemented: OMA-ARC-REST-NetAPI-2011-0352-CR_DeviceCapabilities_CR325_resourceURL_blueprint, OMA-ARC-REST-NetAPI-2011-0358R01-CR_DeviceCapabilities_TS_CONR_resolutions_stage2
	16 November 2011	Many	Implemented: OMA-ARC-REST-NetAPI-2011-0383R01-CR_DeviceCapabilities_global_comments
24 November 2011	Many	Implemented: OMA-ARC-REST-NetAPI-2011-0384-CR_DeviceCapabilities_Appendix_G. Also editorials; replaced internal hyperlinks with cross-references.	
07 December 2011	Many	Implemented: OMA-ARC-REST-NetAPI-2011-0439R01-CR_DevCap_acr_Authorization_and_misc_CONR	

Document Identifier	Date	Sections	Description
	08 December 2011	Cover page	Editorial: removed header. Set the Page Layout on landscape pages to "Different First Page" in the Header setting.
Candidate Version: OMA-TS- REST_NetAPI_DeviceCapabilities- V1_0	20 Dec 2011	n/a	Status changed to Candidate by TP TP ref # OMA-TP-2011-0445- INP_REST_NetAPI_DeviceCapabilities_1_0_ERP_and_ETR_for_Can didate_approval

## Appendix B. Static Conformance Requirements (Normative)

The notation used in this appendix is specified in [SCRRULES].

### B.1 SCR for REST.DevCap Server

Item	Function	Reference	Requirement
REST-DEVCAP-SUPPORT-S-001-M	Support for the RESTful Device Capabilities API	5, 6	
REST-DEVCAP-SUPPORT-S-002-M	Support for the XML request & response format	6	
REST-DEVCAP-SUPPORT-S-003-M	Support for the JSON request & response format	6	
REST-DEVCAP-SUPPORT-S-004-O	Support for the application/x-form-urlencoded format	Appendix C	

#### B.1.1 SCR for REST.DevCap.AccCap Server

Item	Function	Reference	Requirement
REST-DEVCAP-ACCCAP-S-001-M	Support access to device(s) capabilities	6.1	
REST-DEVCAP-ACCCAP-S-002-M	Read device(s) capabilities - GET	6.1.3	

#### B.1.2 SCR for REST.DevCap.List.Subscr Server

Item	Function	Reference	Requirement
REST-DEVCAP-SUBSCR-S-001-M	Support subscriptions for capabilities changes notifications	6.2	
REST-DEVCAP-SUBSCR-S-002-O	Read list of subscriptions - GET	6.2.3	
REST-DEVCAP-SUBSCR-S-003-M	Create subscription for capabilities changes notifications - POST	6.2.5 C.1	

#### B.1.3 SCR for REST.DevCap.Individual.Subscr Server

Item	Function	Reference	Requirement
REST-DEVCAP-IND-SUBSCR-S-001-M	Support for management of individual subscription for capabilities changes notifications	6.3	REST-DEVCAP-IND-SUBSCR-S-002-O
REST-DEVCAP-IND-SUBSCR-S-002-O	Retrieve individual subscription for capabilities changes notifications - GET	6.3.3	

Item	Function	Reference	Requirement
REST-DEVCAP-IND-SUBSCR-S-003-M	Delete individual subscription for capabilities changes notifications - DELETE	6.3.6	

#### B.1.4 SCR for REST.DevCap.Notif Server

Item	Function	Reference	Requirement
REST-DEVCAP-NOTIF-S-001-M	Support for notifying application about device capabilities changes	6.4	
REST-DEVCAP-NOTIF-S-002-M	Notify application about device capabilities changes - POST	6.4.5	

#### B.1.5 SCR for REST.DevCap.AccConfig Server

Item	Function	Reference	Requirement
REST-DEVCAP-ACCCONFIG-S-001-O	Support for access to device(s) configurations	6.5 6.6 6.7	REST-DEVCAP-ACCCONFIG-S-002-O AND/OR REST-DEVCAP-ACCCONFIG-S-003-O AND/OR REST-DEVCAP-ACCCONFIG-S-004-O
REST-DEVCAP-ACCCONFIG-S-002-O	Push configuration to a device - POST	6.5.5 C.2	
REST-DEVCAP-ACCCONFIG-S-003-O	Retrieve available configurations for device(s) - GET	6.6.3	
REST-DEVCAP-ACCCONFIG-S-004-O	Retrieve configuration history for device(s) - GET	6.7.3	

## Appendix C. Application/x-www-form-urlencoded Request Format for POST Operations (Normative)

This section defines a format for RESTful Device Capabilities API requests where the body of the request is encoded using the application/x-www-form-urlencoded MIME type.

Note: only the request body is encoded as application/x-www-form-urlencoded, the response is still encoded as XML or JSON depending on the preference of the client and the capabilities of the server. Names and values MUST follow the application/x-www-form-urlencoded character escaping rules from [W3C\_URLENC].

The encoding is defined below for the following Device Capabilities REST operations which are based on POST requests:

- Create a subscription for device capabilities changes notification
- Push a configuration to a device

### C.1 Start device capabilities changes notifications

This operation is used to create a new device capabilities change subscription to notifications for the identified device(s), see section 6.2.5.

This REST operation is used by the application to start the device change notifications. It MUST use the HTTP POST operation. If the operation was successful, it returns an HTTP Status of “201 Created”.

The notifyURL either contains the Client-side Notification URL (as defined by the client) or the Server-side Notification URL (as obtained during the creation of the Notification Channel [REST\_NetAPI\_NotificationChannel]).

The following parameters are defined:

Name	Type/Values	Optional	Description
timeCreated	xsd:dateTime	No	The date/time when the subscription was created
notifyURL	xsd:anyURI	No	Notification endpoint definition. For the use of Client-side Notification URLs and Server-side Notification URLs in this parameter, see sections 6.2 and 6.2.5.
callbackData	xsd:string	Yes	Data the application can register with the server when subscribing to notifications, and that are passed back unchanged in each of the related notifications
notificationFormat	xsd:string	Yes	Default: XML. Application can specify format of the resource representation in notifications that are related to this subscription. The choice is between {XML, JSON}.
clientCorrelator	xsd:string	Yes	A correlator that the client can use to tag this particular resource representation during a request to create a resource on the server.  This field SHOULD be present. Note: this allows the client to recover from communication failures during resource creation and therefore avoids re-sending the message in such situations.  In case the field is present, the server SHALL not alter its value, and SHALL provide it as part of the representation of this resource. In case the field is not present, the server SHALL NOT generate it.

## C.1.1 Example 1, using 'tel' URI (Informative)

### C.1.1.1 Request

```
POST /exampleAPI/devicecapabilities/v1/tel%3A%2B19585550100/subscriptions HTTP/1.1
Accept: application/xml
Host: example.com
Content-Type: application/x-www-form-urlencoded
Content-Length: nnnn

timeCreated=2010-03-21T13:23:21Z&
clientCorrelator=54321&
notifyURL=http://application.example.com/notifications/CapabilitiesChangeNotification&
callbackData=12345&
notificationFormat=JSON
```

### C.1.1.2 Response

```
HTTP/1.1 201 Created
Date: Thu, 04 Jun 2010 02:52:00 GMT
Location: http://example.com/exampleAPI/devicecapabilities/v1/tel%3A%2B19585550100/subscriptions/sub123
Content-Type: application/xml
Content-Length: nnnn

<?xml version="1.0" encoding="UTF-8"?>
<dc:deviceCapabilitiesChangeSubscription xmlns:dc="urn:oma:xml:rest:netapi:devicecapabilities:1">
  <timeCreated>2010-03-21T13:23:21Z</timeCreated>
  <callbackReference>
    <notifyURL>http://application.example.com/notifications/CapabilitiesChangeNotification</notifyURL>
    <callbackData>12345</callbackData>
    <notificationFormat>JSON</notificationFormat>
  </callbackReference>
  <clientCorrelator>54321</clientCorrelator>
  <resourceURL>http://example.com/exampleAPI/devicecapabilities/v1/tel%3A%2B19585550100/subscriptions/sub123</resourceURL>
</dc:deviceCapabilitiesChangeSubscription>
```

## C.1.2 Example 2, using 'acr' URI (Informative)

### C.1.2.1 Request

```
POST /exampleAPI/devicecapabilities/v1/acr%3A%2B19585550100/subscriptions HTTP/1.1
Accept: application/xml
Host: example.com
Content-Type: application/x-www-form-urlencoded
Content-Length: nnnn

timeCreated=2010-03-21T13:23:21Z&
clientCorrelator=54321&
notifyURL=http://application.example.com/notifications/CapabilitiesChangeNotification&
callbackData=12345&
notificationFormat=JSON
```



## C.1.2.2 Response

```
HTTP/1.1 201 Created
Date: Thu, 04 Jun 2010 02:52:00 GMT
Location: http://example.com/exampleAPI/devicecapabilities/v1/acr%3A pseudonym123/subscriptions/sub123
Content-Type: application/xml
Content-Length: nnnn
```

```
<?xml version="1.0" encoding="UTF-8"?>
<dc:deviceCapabilitiesChangeSubscription xmlns:dc="urn:oma:xml:rest:netapi:devicecapabilities:1">
  <timeCreated>2010-03-21T13:23:21Z</timeCreated>
  <callbackReference>
    <notifyURL>http://application.example.com/notifications/CapabilitiesChangeNotification</notifyURL>
    <callbackData>12345</callbackData>
    <notificationFormat>JSON</notificationFormat>
  </callbackReference>
  <clientCorrelator>54321</clientCorrelator>
  <resourceURL>http://example.com/exampleAPI/devicecapabilities/v1/acr%3A pseudonym123/subscriptions/sub123</resourceURL>
</dc:deviceCapabilitiesChangeSubscription>
```

## C.2 Push a Configuration to a Device

This operation is used to push a configuration to a device/group of devices, see section 6.5.5.

It MUST use the HTTP POST operation. If the operation was successful, it returns an HTTP Status of “201 Created”.

The following parameters are defined:

Name	Type/Values	Optional	Description
configurationId	xsd:string	No	A unique identifier for the configuration
name	xsd:string	No	The name of the configuration
description	xsd:string	No	The description of the configuration

### C.2.1 Example

(Informative)

#### C.2.1.1 Request

```
POST /exampleAPI/devicecapabilities/v1/tel%3A%2B19585550100/configuration HTTP/1.1
Accept: application/xml
Host: example.com
Content-Type: application/x-www-form-urlencoded
Content-Length : nnnn
```

```
configurationId=config12345&
name=configname12345&
description=configdescription12345
```

### C.2.1.2 Response

HTTP/1.1 204 No Content  
Date: Thu, 04 Jun 2010 02:52:00 GMT

## Appendix D. JSON examples (Informative)

JSON (JavaScript Object Notation) is a lightweight, text-based, language-independent data interchange format. It provides a simple means to represent basic name-value pairs, arrays and objects. JSON is relatively trivial to parse and evaluate using standard JavaScript libraries, and hence is suited for invocations from browsers or other processors with JavaScript engines. Further information on JSON can be found at [RFC 4627].

The following examples show the request and response for various operations using a JSON binding. The examples follow the XML to JSON serialization rules in [REST\_NetAPI\_Common]. A JSON response can be obtained by using the content type negotiation mechanism specified in [REST\_NetAPI\_Common].

For full details on the operations themselves please refer to the section number indicated.

### D.1 Retrieve device capabilities (section 6.1.3.1)

Request:

```
GET /exampleAPI/devicecapabilities/v1/tel%3A%2B19585550100/capabilities HTTP/1.1
Accept: application/json
Host: example.com
```

Response:

```
HTTP/1.1 200 OK
Date: Thu, 04 Jun 2010 02:52:00 GMT
Content-Type: application/json
Content-Length: nnnn

{"deviceCapabilities": {
  "deviceId": "123456789012345",
  "link": {
    "href": "http://example.com/exampleconfigurations/exampledeviceprofiles/A1234xyz123.xml",
    "rel": "UserAgentProfileReference"
  },
  "name": "devname123",
  "resourceURL": "http://example.com/exampleAPI/devicecapabilities/v1/tel%3A%2B19585550100/capabilities"
}}
```

### D.2 Retrieve list of subscriptions (section 6.2.3.1)

Request:

```
GET /exampleAPI/devicecapabilities/v1/tel%3A%2B19585550100/subscriptions?resFormat=JSON HTTP/1.1
Accept: application/json
Host: example.com
```

Response:

```
HTTP/1.1 200 OK
Date: Thu, 04 Jun 2010 02:52:00 GMT
Content-Type: application/json
Content-Length: nnnn

{"deviceCapabilitiesChangeSubscriptionList": {
```

```

"deviceCapabilitiesChangeSubscription": {
  "callbackReference": {
    "callbackData": "12345",
    "notifyURL": "http://application.example.com/notifications/CapabilitiesChangeNotification"
  },
  "clientCorrelator": "54321",
  "resourceURL": "http://example.com/exampleAPI/devicecapabilities/v1/tel%3A%2B19585550100/subscriptions/sub123",
  "timeCreated": "2010-03-21T13:23:21Z"
},
"resourceURL": "http://example.com/exampleAPI/devicecapabilities/v1/tel%3A%2B19585550100/subscriptions"
}}

```

### D.3 Subscribe to device capabilities changes notifications for single device using 'tel' URI (section 6.2.5.1)

Request:

```

POST /exampleAPI/devicecapabilities/v1/tel%3A%2B19585550100/subscriptions HTTP/1.1
Accept: application/json
Host: example.com
Content-Type: application/json
Content-Length: nnnn

```

```

{"deviceCapabilitiesChangeSubscription": {
  "callbackReference": {
    "callbackData": "12345",
    "notificationFormat": "JSON",
    "notifyURL": "http://application.example.com/notifications/CapabilitiesChangeNotification"
  },
  "clientCorrelator": "54321",
  "timeCreated": "2010-03-21T13:23:21Z"
}}

```

Response:

```

HTTP/1.1 201 Created
Date: Thu, 04 Jun 2010 02:52:00 GMT
Location: http://example.com/exampleAPI/devicecapabilities/v1/tel%3A%2B19585550100/subscriptions/sub123
Content-Type: application/json
Content-Length: nnnn

```

```

{"deviceCapabilitiesChangeSubscription": {
  "callbackReference": {
    "callbackData": "12345",
    "notificationFormat": "JSON",
    "notifyURL": "http://application.example.com/notifications/CapabilitiesChangeNotification"
  },
  "clientCorrelator": "54321",
  "resourceURL": "http://example.com/exampleAPI/devicecapabilities/v1/tel%3A%2B19585550100/subscriptions/sub123",
  "timeCreated": "2010-03-21T13:23:21Z"
}}

```



Host: example.com

Response:

```
HTTP/1.1 200 OK
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"deviceCapabilitiesChangeSubscription": {
  "callbackReference": {
    "callbackData": "12345",
    "notifyURL": "http://application.example.com/notifications/CapabilitiesChangeNotification"
  },
  "clientCorrelator": "54321",
  "resourceURL": "http://example.com/exampleAPI/devicecapabilities/v1/tel%3A%2B19585550100/subscriptions/sub123",
  "timeCreated": "2010-03-21T13:23:21Z"
}}
```

## D.6 Delete subscription (section 6.3.6.1)

Request:

```
DELETE /exampleAPI/devicecapabilities/v1/tel%3A%2B19585550100/subscriptions/sub123 HTTP/1.1
Accept: application/json
Host: example.com
```

Response:

```
HTTP/1.1 204 No Content
Date: Thu, 04 Jun 2010 02:52:00 GMT
```

## D.7 Device capabilities changes notification (section 6.4.5.1)

Request:

```
POST /notifications/CapabilityChangeNotification HTTP/1.1
Accept: application/json
Host: application.example.com
Content-Type: application/json
Content-Length: nnnn

{"deviceCapabilitiesNotification": {
  "callbackData": "12345",
  "changeNotificationEnd": "false",
  "deviceAddress": "tel:+19585550100",
  "deviceId": "123456789012345",
  "link": [
    {
      "href": "http://example.com/exampleAPI/devicecapabilities/v1/tel%3A%2B19585550100/subscriptions/sub123",
      "rel": "DeviceCapabilitiesChangeSubscription"
    }
  ]
}
```

```

    },
    {
      "href": "http://example.com/exampleAPI/devicecapabilities/v1/tel%3A%2B19585550100/capabilities",
      "rel": "DeviceCapabilities"
    },
    {
      "href": "http://example.com/exampleAPI/devicecapabilities/v1/tel%3A%2B19585550100/configuration",
      "rel": "DeviceConfiguration"
    }
  ]
}
}

```

Response:

```

HTTP/1.1 204 No content
Date: Thu, 04 Jun 2010 02:52:00 GMT

```

## D.8 End of changes notification (section 6.4.5.2)

Request:

```

POST /notifications/CapabilitiesChangeNotification HTTP/1.1

```

```

Accept: application/json

```

```

Host: application.example.com

```

```

Content-Type: application/json

```

```

Content-Length: nnnn

```

```

{"deviceCapabilitiesNotification": {
  "callbackData": "12345",
  "changeNotificationEnd": "true",
  "deviceAddress": "tel:+19585550100",
  "deviceId": "123456789012345",
  "link": [
    {
      "href": "http://example.com/exampleAPI/devicecapabilities/v1/tel%3A%2B19585550100/subscriptions/sub123",
      "rel": "DeviceCapabilitiesChangeSubscription"
    },
    {
      "href": "http://example.com/exampleAPI/devicecapabilities/v1/tel%3A%2B19585550100/capabilities",
      "rel": "DeviceCapabilities"
    },
    {
      "href": "http://example.com/exampleAPI/devicecapabilities/v1/tel%3A%2B19585550100/configuration",
      "rel": "DeviceConfiguration"
    }
  ]
}
}
}

```

Response:

```

HTTP/1.1 204 No Content

```

Date: Thu, 04 Jun 2010 02:52:00 GMT

## D.9 Error notification (notification cancellation) (section 6.4.5.3)

Request:

POST /notifications/StatusNotification HTTP/1.1

Accept : application/json

Host: application.example.com

Content-Type: application/json

Content-Length: nnnn

```
{
  "deviceCapabilitiesCancellationNotification": {
    "callbackData": "12345",
    "deviceAddress": "tel:+19585550100",
    "link": [
      {
        "href": "http://example.com/exampleAPI/devicecapabilities/v1/tel%3A%2B19585550100/subscriptions/sub123",
        "rel": "DeviceCapabilitiesChangeSubscription"
      },
      {
        "href": "http://example.com/exampleAPI/devicecapabilities/v1/tel%3A%2B19585550100/capabilities",
        "rel": "DeviceCapabilities"
      },
      {
        "href": "http://example.com/exampleAPI/devicecapabilities/v1/tel%3A%2B19585550100/configuration",
        "rel": "DeviceConfiguration"
      }
    ],
    "reason": {
      "messageId": "SVC0001",
      "text": "A service error occurred. %1 %2",
      "variables": [
        "Status information is not available for",
        " tel:+19585550100"
      ]
    }
  }
}
```

Response:

HTTP/1.1 204 No Content

Date: Thu, 04 Jun 2010 02:52:00 GMT

## D.10 Push configuration to single device (section 6.5.5.1)

Request:



```
POST /exampleAPI/devicecapabilities/v1/tel%3A%2B19585550100/configuration HTTP/1.1
Accept: application/json
Host: example.com
Content-Type: application/json
Content-Length: nnnn
```

```
{"deviceConfiguration": {
  "configurationId": "config12345",
  "description": "configdescription12345",
  "name": "configname12345"
}}
```

Response:

```
HTTP/1.1 204 No Content
Date: Thu, 04 Jun 2010 02:52:00 GMT
```

## D.11 Retrieve available device configurations (section 6.6.3.1)

Request:

```
GET /exampleAPI/devicecapabilities/v1/tel%3A%2B19585550100/configuration/available HTTP/1.1
Accept: application/json
Host: example.com
```

Response:

```
HTTP/1.1 200 OK
Date: Thu, 04 Jun 2009 02:51:59 GMT
Content-Type: application/json
Content-Length: nnnn

{"deviceConfigurationList": {
  "deviceConfiguration": [
    {
      "configurationId": "config12345",
      "description": "configdescription12345",
      "link": {
        "href": "http://example.com/exampleconfigurations/exampledeviceconfig/12345.xml",
        "rel": "ConfigurationProfileReference"
      },
      "name": "configname12345"
    },
    {
      "configurationId": "config12346",
      "description": "configdescription12346",
      "link": {
        "href": "http://example.com/exampleconfigurations/exampledeviceconfig/12346.xml",
        "rel": "ConfigurationProfileReference"
      },
      "name": "configname12346"
    }
  ]
}
```

```

{
  "configurationId": "config12347",
  "description": "configdescription12347",
  "link": {
    "href": "http://example.com/exampleconfigurations/exampledeviceconfig/12347.xml",
    "rel": "ConfigurationProfileReference"
  },
  "name": "configname12347"
}
],
"resourceURL": "http://example.com/exampleAPI/devicecapabilities/v1/tel%3A%2B19585550100/configuration/available"
}}

```

## D.12 Retrieve configuration history (section 6.7.3.1)

Request:

```

GET /exampleAPI/devicecapabilities/v1/tel%3A%2B19585550100/configuration/history HTTP/1.1
Accept: application/json
Host: example.com

```

Response:

```

HTTP/1.1 200 OK
Date: Thu, 04 Jun 2010 02:52:00 GMT
Content-Type: application/json
Content-Length: nnnn

{"deviceConfigurationHistoryList": {
  "configurationHistoryEntry": [
    {
      "deviceConfiguration": {
        "configurationId": "config12345",
        "description": "configdescription12345",
        "link": {
          "href": "http://example.com/exampleconfigurations/exampledeviceconfig/12345.xml",
          "rel": "ConfigurationProfileReference"
        },
        "name": "configname12345"
      },
      "timestamp": "2009-11-19T12:00:00"
    },
    {
      "deviceConfiguration": {
        "configurationId": "config12347",
        "description": "configdescription12347",
        "link": {
          "href": "http://example.com/exampleconfigurations/exampledeviceconfig/12347.xml",
          "rel": "ConfigurationProfileReference"
        },
        "name": "configname12347"
      },
      "timestamp": "2009-10-19T12:00:00"
    }
  ]
}

```

```
}  
],  
"resourceURL": "http://example.com/exampleAPI/devicecapabilities/v1/tel%3A%2B19585550100/configuration/history"  
}}
```

## Appendix E. Parlay X operations mapping (Informative)

The table below illustrates the mapping between REST resources/methods and Parlay X [3GPP 29.199-18] equivalent operations.

REST Resource	REST Method	REST section reference	Parlay X equivalent operation
Device capabilities	GET	6.1.3	getCapabilities
Subscriptions for device capabilities changes notifications	POST	6.2.5	startNotification
Individual subscription for device capabilities changes notifications	DELETE	6.3.6	endNotification
Notification on device capabilities changes	POST	6.4.5	deviceNotification, deviceEnd, and deviceError
Device configuration	POST	6.5.5	pushConfiguration
Available configurations	GET	6.6.3	getConfigurationList
Configuration history	GET	6.7.3	getConfigurationHistory

**Table 1: Parlay X operations mapping**

## Appendix F. Light-weight resources (Informative)

As this version of the specification does not define any light-weight resources, this Appendix is empty.

## Appendix G. Authorization aspects (Normative)

This appendix specifies how to use the RESTful DeviceCapabilities API in combination with some authorization frameworks.

### G.1 Use with OMA Authorization Framework for Network APIs

The RESTful Device Capabilities API MAY support the authorization framework defined in [Autho4API\_10].

A RESTful Device Capabilities API supporting [Autho4API\_10]:

- SHALL conform to section D.1 of [REST\_NetAPI\_Common];
- SHALL conform to this section G.1.

#### G.1.1 Scope values

##### G.1.1.1 Definitions

In compliance with [Autho4API\_10], an authorization server serving clients requests for getting authorized access to the resources exposed by the RESTful Device Capabilities API:

- SHALL support the scope values defined in the table below;
- MAY support scope values not defined in this specification.

Scope value	Description	For one-time access token
oma_rest_devicecapabilities.all_{apiVersion}	Provide access to all defined operations on the resources in this version of the API. The {apiVersion} part of this identifier SHALL have the same value as the “apiVersion” URL variable which is defined in section 5.1. This scope value is the union of the other scope values listed in next rows of this table.	No
oma_rest_devicecapabilities.cap	Provide access to all defined operations on device capabilities	No
oma_rest_devicecapabilities.subscr	Provide access to all defined operations on subscriptions	No
oma_rest_devicecapabilities.config_all	Provide access to all defined operations on configuration	No
oma_rest_devicecapabilities.config_read	Provide access only to read operations on configuration	No

**Table 2: Scope values for RESTful Device Capabilities API**

##### G.1.1.2 Downscoping

In the case where the client requests authorization for “oma\_rest\_devicecapabilities.all\_{apiVersion}” scope, the authorization server and/or resource owner MAY restrict the granted scope to some of the following scope values:

- “oma\_rest\_devicecapabilities.cap”
- “oma\_rest\_devicecapabilities.subscr”
- “oma\_rest\_devicecapabilities.config\_all”
- “oma\_rest\_devicecapabilities.config\_read”

In the case where the client requests authorization for “oma\_rest\_devicecapabilities.config\_all” scope, the authorization server and/or resource owner MAY restrict the granted scope to some of the following scope values:

- “oma\_rest\_devicecapabilities.config\_read”

### **G.1.1.3 Mapping with resources and methods**

Tables in this section specify how the scope values defined in section G.1.1.1 for the RESTful Device Capabilities API map to the REST resources and methods of this API. In these tables, the root “oma\_rest\_devicecapabilities.” of scope values is omitted for readability reasons.

Resource	URL Base URL: http://{serverRoot}/devicecapabilities/{apiVersion}	Section reference	HTTP verbs			
			GET	PUT	POST	DELETE
Device capabilities	{equipmentId}/capabilities	6.1	<b>all_{apiVersion} or cap</b>	n/a	n/a	

**Table 3: Required scope values for: Device capabilities**

Resource	URL Base URL: http://{serverRoot}/devicecapabilities/{apiVersion}	Section reference	HTTP verbs			
			GET	PUT	POST	DELETE
Subscriptions for device capabilities changes notifications	{equipmentId}/subscriptions	6.2	<b>all_{apiVersion} or subscr</b>	n/a	<b>all_{apiVersion} or subscr</b>	
Individual subscription for device capabilities changes notifications	{equipmentId}/subscriptions/{subscriptionId}	6.3	<b>all_{apiVersion} or subscr</b>	n/a	n/a	<b>all_{apiVersion} or subscr</b>

**Table 4: Required scope values for: Subscription Management for device capabilities changes**

Resource	URL Base URL: http://{serverRoot}/devicecapabilities/{apiVersion}	Section reference	HTTP verbs			
			GET	PUT	POST	DELETE
Device configuration	{equipmentId}/configuration	6.5	<b>all_{apiVersion} or config_all or config_read</b>	n/a	<b>all_{apiVersion} or config_all</b>	
Available configurations	{equipmentId}/configuration/available	6.6	<b>all_{apiVersion} or config_all or config_read</b>	n/a	n/a	



Configuration history	/{{equipmentId}}/configuration/history	6.7	all_{apiVersion} or config_all or config_read	n/a	n/a	
-----------------------	--	-----	--	-----	-----	--

**Table 5: Required scope values for: Retrieving configuration (available or history) and pushing a configuration**

## G.1.2 Use of 'acr:Authorization'

This section specifies the use of 'acr:Authorization' in place of an end user identifier in a resource URL path.

An 'acr' URI of the form 'acr:Authorization', where 'Authorization' is a reserved keyword MAY be used to avoid exposing a real end user identifier in the resource URL path.

A client MAY use 'acr:Authorization' in a resource URL in place of the {equipmentId} resource URL variable in the resource URL path, when the RESTful Device Capabilities API is used in combination with [Autho4API\_10].

In the case the RESTful Device Capabilities supports [Autho4API\_10], the server:

- SHALL accept 'acr:Authorization' as a valid value for the resource URL variable {equipmentId}.
- SHALL conform to [REST\_Common\_TS] section 5.8.1.1 regarding the processing of 'acr:Authorization'.

