



# RESTful Network API for Quality of Service

## Candidate Version 1.0 – 25 Nov 2014

---

**Open Mobile Alliance**  
OMA-TS-REST\_NetAPI\_QoS-V1\_0-20141125-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 endeavours 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.

© 2014 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>9</b>
<b>2. REFERENCES</b> .....	<b>10</b>
<b>2.1 NORMATIVE REFERENCES</b> .....	<b>10</b>
<b>2.2 INFORMATIVE REFERENCES</b> .....	<b>10</b>
<b>3. TERMINOLOGY AND CONVENTIONS</b> .....	<b>11</b>
<b>3.1 CONVENTIONS</b> .....	<b>11</b>
<b>3.2 DEFINITIONS</b> .....	<b>11</b>
<b>3.3 ABBREVIATIONS</b> .....	<b>11</b>
<b>4. INTRODUCTION</b> .....	<b>13</b>
<b>4.1 VERSION 1.0</b> .....	<b>13</b>
<b>5. QUALITY OF SERVICE API DEFINITION</b> .....	<b>14</b>
<b>5.1 RESOURCES SUMMARY</b> .....	<b>14</b>
<b>5.2 DATA TYPES</b> .....	<b>21</b>
5.2.1 XML Namespaces.....	21
5.2.2 Structures .....	21
5.2.2.1 Type: <i>PredefinedQoSFeatureList</i> .....	21
5.2.2.2 Type: <i>PredefinedQoSFeature</i> .....	21
5.2.2.3 Type: <i>PredefinedMediaInfo</i> .....	22
5.2.2.4 Type: <i>QoSFeatureData</i> .....	22
5.2.2.5 Type: <i>MediaComponentDescription</i> .....	26
5.2.2.6 Type: <i>BandwidthInformation</i> .....	28
5.2.2.7 Type: <i>IpFlowDescription</i> .....	28
5.2.2.8 Type: <i>FlowDescription</i> .....	29
5.2.2.9 Type: <i>IpAddress</i> .....	29
5.2.2.10 Type: <i>PortNumber</i> .....	30
5.2.2.11 Type: <i>PortRange</i> .....	30
5.2.2.12 Type: <i>AppliedQoSFeatureList</i> .....	30
5.2.2.13 Type: <i>QoSFeaturesSubscriptionList</i> .....	30
5.2.2.14 Type: <i>PredefinesQoSFeaturesSubscriptionList</i> .....	30
5.2.2.15 Type: <i>AppliedQoSFeaturesSubscriptionList</i> .....	31
5.2.2.16 Type: <i>PredefinedQoSFeaturesSubscription</i> .....	31
5.2.2.17 Type: <i>AppliedQoSFeaturesSubscription</i> .....	33
5.2.2.18 Type: <i>PredefinedQoSAvailabilityNotification</i> .....	34
5.2.2.19 Type: <i>AppliedQoSFeaturesNotification</i> .....	34
5.2.3 Enumerations .....	35
5.2.3.1 Enumeration: <i>DefaultAction</i> .....	35
5.2.3.2 Enumeration: <i>FlowDirection</i> .....	35
5.2.3.3 Enumeration: <i>FlowStatus</i> .....	35
5.2.3.4 Enumeration: <i>FlowUsage</i> .....	36
5.2.3.5 Enumeration: <i>MediaType</i> .....	36
5.2.3.6 Enumeration: <i>ProtocolType</i> .....	36
5.2.3.7 Enumeration: <i>QoSEvent</i> .....	37
5.2.3.8 Enumeration: <i>ReservationPriority</i> .....	37
5.2.4 Values of the Link “rel” attribute.....	37
<b>5.3 SEQUENCE DIAGRAMS</b> .....	<b>38</b>
5.3.1 Applying a predefined QoS feature on the end user connection .....	38
5.3.2 Applying a custom QoS feature on the end user connection.....	39
5.3.3 Updating a QoS feature applied on the end user connection .....	40
5.3.4 Subscription to notifications on availability of predefined QoS features.....	41
5.3.5 Subscription to notifications about events occurring for applied QoS features .....	44
5.3.6 A limit for applied QoS feature has been reached .....	46
<b>6. DETAILED SPECIFICATION OF THE RESOURCES</b> .....	<b>48</b>
<b>6.1 RESOURCE: PREDEFINED QoS FEATURES AVAILABLE TO A USER</b> .....	<b>48</b>
6.1.1 Request URL variables .....	48
6.1.2 Response Codes and Error Handling .....	48

6.1.3	GET.....	49
6.1.3.1	<i>Example 1: Retrieving a list of predefined QoS features generally available to the end user (Informative) .....</i>	49
6.1.3.1.1	Request.....	49
6.1.3.1.2	Response.....	49
6.1.3.2	<i>Example 2: Retrieving a list of predefined QoS features currently available to the end user (Informative).....</i>	50
6.1.3.2.1	Request.....	51
6.1.3.2.2	Response.....	51
6.1.4	PUT.....	51
6.1.5	POST.....	51
6.1.6	DELETE .....	51
<b>6.2</b>	<b>RESOURCE: QOS FEATURES APPLIED ON AN END USER CONNECTION .....</b>	<b>52</b>
6.2.1	Request URL variables .....	52
6.2.2	Response Codes and Error Handling .....	52
6.2.3	GET.....	52
6.2.3.1	<i>Example: Retrieving a list of QoS features applied on the end user connection (Informative).....</i>	52
6.2.3.1.1	Request.....	52
6.2.3.1.2	Response.....	52
6.2.4	PUT.....	53
6.2.5	POST.....	53
6.2.5.1	<i>Example 1: Request to apply a predefined QoS feature on the end user connection (Informative) .....</i>	53
6.2.5.1.1	Request.....	53
6.2.5.1.2	Response.....	54
6.2.5.2	<i>Example 2: Request to apply a predefined QoS feature on the end user connection, response with location of created resource (Informative) .....</i>	55
6.2.5.2.1	Request.....	55
6.2.5.2.2	Response.....	55
6.2.5.3	<i>Example 3: Request to apply a custom QoS feature on the end user connection (Informative).....</i>	56
6.2.5.3.1	Request.....	56
6.2.5.3.2	Response.....	57
6.2.5.4	<i>Example 4: Application service provider sponsoring user's data usage (Informative) .....</i>	59
6.2.5.4.1	Request.....	59
6.2.5.4.2	Response.....	60
6.2.5.5	<i>Example 4: Request to apply a custom QoS feature on the end user connection, which is not supported by the server (Informative).....</i>	61
6.2.5.5.1	Request.....	61
6.2.5.5.2	Response.....	62
6.2.6	DELETE .....	62
<b>6.3</b>	<b>RESOURCE: INDIVIDUAL QOS FEATURE APPLIED ON AN END USER CONNECTION .....</b>	<b>62</b>
6.3.1	Request URL variables .....	62
6.3.2	Response Codes and Error Handling .....	63
6.3.3	GET.....	63
6.3.3.1	<i>Example: Retrieving an individual QoS feature applied on the end user connection (Informative).....</i>	63
6.3.3.1.1	Request.....	63
6.3.3.1.2	Response.....	63
6.3.4	PUT.....	64
6.3.4.1	<i>Example: Updating attributes for QoS feature applied on the end user connection (Informative).....</i>	64
6.3.4.1.1	Request.....	64
6.3.4.1.2	Response.....	65
6.3.5	POST.....	65
6.3.6	DELETE .....	65
6.3.6.1	<i>Example: Removing QoS feature which is applied to the end user connection (Informative).....</i>	66
6.3.6.1.1	Request.....	66
6.3.6.1.2	Response.....	66
<b>6.4</b>	<b>RESOURCE: INDIVIDUAL ATTRIBUTE FOR QOS FEATURE APPLIED ON AN END USER CONNECTION .....</b>	<b>66</b>
6.4.1	Request URL variables .....	66
6.4.1.1	<i>Light-weight relative resource paths.....</i>	66
6.4.2	Response Codes and Error Handling .....	67
6.4.3	GET.....	67
6.4.3.1	<i>Example 1: Retrieving duration time for QoS feature applied on the end user connection (Informative).....</i>	67
6.4.3.1.1	Request.....	67
6.4.3.1.2	Response.....	67

- 6.4.3.2 *Example 2: Retrieving IP flow status on media number level for QoS feature applied on the end user connection (Informative)* ..... 67
  - 6.4.3.2.1 Request..... 67
  - 6.4.3.2.2 Response..... 68
- 6.4.3.3 *Example 3: Retrieving IP flow status on IP flow number level for QoS feature applied on the end user connection (Informative)* ..... 68
  - 6.4.3.3.1 Request..... 68
  - 6.4.3.3.2 Response..... 68
- 6.4.4 PUT ..... 68
  - 6.4.4.1 *Example 1: Updating duration time for QoS feature applied on the end user connection (Informative)*..... 68
    - 6.4.4.1.1 Request..... 68
    - 6.4.4.1.2 Response..... 68
  - 6.4.4.2 *Example 2: Updating IP flow status on IP flow number level for QoS feature applied on the end user connection (Informative)*..... 69
    - 6.4.4.2.1 Request..... 69
    - 6.4.4.2.2 Response..... 69
- 6.4.5 POST..... 69
- 6.4.6 DELETE ..... 69
- 6.5 RESOURCE: ALL SUBSCRIPTIONS TO QoS NOTIFICATIONS ..... 69**
  - 6.5.1 Request URL variables ..... 69
  - 6.5.2 Response Codes and Error Handling ..... 70
  - 6.5.3 GET..... 70
    - 6.5.3.1 *Example: Retrieving all subscriptions to notifications related to QoS features (Informative)*..... 70
      - 6.5.3.1.1 Request..... 70
      - 6.5.3.1.2 Response..... 70
  - 6.5.4 PUT ..... 71
  - 6.5.5 POST..... 71
  - 6.5.6 DELETE ..... 71
- 6.6 RESOURCE: ALL SUBSCRIPTIONS TO NOTIFICATIONS FOR PREDEFINED QoS FEATURES ..... 71**
  - 6.6.1 Request URL variables ..... 71
  - 6.6.2 Response Codes and Error Handling ..... 71
  - 6.6.3 GET..... 71
    - 6.6.3.1 *Example: Retrieving all subscriptions to notifications on availability of predefined QoS features (Informative)* ..... 72
      - 6.6.3.1.1 Request..... 72
      - 6.6.3.1.2 Response..... 72
  - 6.6.4 PUT..... 72
  - 6.6.5 POST..... 72
    - 6.6.5.1 *Example: Creating a new subscription for notifications on availability of predefined QoS features (Informative)* ..... 72
      - 6.6.5.1.1 Request..... 72
      - 6.6.5.1.2 Response..... 73
  - 6.6.6 DELETE ..... 73
- 6.7 RESOURCE: INDIVIDUAL SUBSCRIPTION TO NOTIFICATIONS FOR PREDEFINED QoS FEATURES ..... 73**
  - 6.7.1 Request URL variables ..... 73
  - 6.7.2 Response Codes and Error Handling ..... 74
  - 6.7.3 GET..... 74
    - 6.7.3.1 *Example: Retrieving an individual subscriptions to notifications on availability of predefined QoS features (Informative)*..... 74
      - 6.7.3.1.1 Request..... 74
      - 6.7.3.1.2 Response..... 74
  - 6.7.4 PUT..... 74
  - 6.7.5 POST..... 75
  - 6.7.6 DELETE ..... 75
    - 6.7.6.1 *Example: Cancelling a subscription to notifications on availability of predefined QoS features (Informative)*..... 75
      - 6.7.6.1.1 Request..... 75
      - 6.7.6.1.2 Response..... 75
- 6.8 RESOURCE: ALL SUBSCRIPTIONS TO NOTIFICATIONS FOR QoS FEATURES APPLIED ON AN END USER CONNECTION ..... 75**
  - 6.8.1 Request URL variables ..... 75
  - 6.8.2 Response Codes and Error Handling ..... 75
  - 6.8.3 GET..... 75

- 6.8.3.1 *Example: Retrieving all subscriptions to notifications about events occurring for applied QoS features (Informative)* . 76
  - 6.8.3.1.1 Request..... 76
  - 6.8.3.1.2 Response..... 76
- 6.8.4 PUT..... 76
- 6.8.5 POST..... 76
  - 6.8.5.1 *Example: Creating a new subscription for notifications about events occurring for applied QoS features (Informative)*... 76
    - 6.8.5.1.1 Request..... 76
    - 6.8.5.1.2 Response..... 77
- 6.8.6 DELETE ..... 77
- 6.9 RESOURCE: INDIVIDUAL SUBSCRIPTION TO NOTIFICATIONS FOR QoS FEATURES APPLIED ON AN END USER CONNECTION ..... 77**
  - 6.9.1 Request URL variables ..... 77
  - 6.9.2 Response Codes and Error Handling ..... 78
  - 6.9.3 GET..... 78
    - 6.9.3.1 *Example: Retrieving an individual subscription to notifications about events occurring for applied QoS features (Informative)*..... 78
      - 6.9.3.1.1 Request..... 78
      - 6.9.3.1.2 Response..... 78
  - 6.9.4 PUT..... 78
  - 6.9.5 POST..... 78
  - 6.9.6 DELETE ..... 79
    - 6.9.6.1 *Example: Cancelling a subscription to notifications about events occurring for applied QoS features (Informative)*.... 79
      - 6.9.6.1.1 Request..... 79
      - 6.9.6.1.2 Response..... 79
- 6.10 RESOURCE: CLIENT NOTIFICATION ABOUT AVAILABILITY OF PREDEFINED QoS FEATURES ..... 79**
  - 6.10.1 Request URL variables ..... 79
  - 6.10.2 Response Codes and Error Handling ..... 79
  - 6.10.3 GET..... 79
  - 6.10.4 PUT..... 79
  - 6.10.5 POST..... 79
    - 6.10.5.1 *Example: Notifying a client that a predefined QoS feature is available to the end user (Informative)*..... 80
      - 6.10.5.1.1 Request..... 80
      - 6.10.5.1.2 Response..... 80
  - 6.10.6 DELETE ..... 80
- 6.11 RESOURCE: CLIENT NOTIFICATION ABOUT AN EVENT OCCURRED FOR QoS FEATURES APPLIED ON AN END USER CONNECTION ..... 80**
  - 6.11.1 Request URL variables ..... 80
  - 6.11.2 Response Codes and Error Handling ..... 80
  - 6.11.3 GET..... 80
  - 6.11.4 PUT..... 81
  - 6.11.5 POST..... 81
    - 6.11.5.1 *Example: Notifying a client about an event occurring for applied QoS features (Informative)*..... 81
      - 6.11.5.1.1 Request..... 81
      - 6.11.5.1.2 Response..... 81
  - 6.11.6 DELETE ..... 81
- 7. FAULT DEFINITIONS ..... 82**
  - 7.1 SERVICE EXCEPTIONS..... 82**
    - 7.1.1 SVC0340: Insufficient connection resources..... 82
    - 7.1.2 SVC0341: Unknown QoS feature identifier ..... 82
    - 7.1.3 SVC0342: End user is not online ..... 82
    - 7.1.4 SVC0343: Specified and temporary QoS feature conflict ..... 82
    - 7.1.5 SVC1010: Insufficient information to identify IP flow ..... 83
    - 7.1.6 SVC1011: Specified IP flow does not exist..... 83
    - 7.1.7 SVC1012: Duplicate media/flow number..... 83
  - 7.2 POLICY EXCEPTIONS ..... 83**
    - 7.2.1 POL1032: Custom QoS features not supported ..... 83
    - 7.2.2 POL1033: Specifying volume limits not supported..... 83
    - 7.2.3 POL1034: Subscriptions to predefined QoS features not supported..... 84

7.2.4	POL1035: Multiple notifications for predefined QoS features not supported .....	84
7.2.5	POL1036: Sponsored QoS features not supported.....	84
<b>APPENDIX A.</b>	<b>CHANGE HISTORY (INFORMATIVE).....</b>	<b>85</b>
<b>A.1</b>	<b>APPROVED VERSION HISTORY .....</b>	<b>85</b>
<b>A.2</b>	<b>DRAFT/CANDIDATE VERSION 1.0 HISTORY .....</b>	<b>85</b>
<b>APPENDIX B.</b>	<b>STATIC CONFORMANCE REQUIREMENTS (NORMATIVE).....</b>	<b>87</b>
<b>B.1</b>	<b>SCR FOR REST.QoS SERVER.....</b>	<b>87</b>
B.1.1	SCR for REST.QoS.PredefinedQoSFeatures Server .....	87
B.1.2	SCR for REST.QoS.AppliedQoSFeatures Server .....	87
B.1.3	SCR for REST.QoS.IndAppliedQoSFeature Server .....	87
B.1.4	SCR for REST.QoS.IndAppliedQoSFeature.Attribute Server .....	87
B.1.5	SCR for REST.QoS.Subscriptions Server .....	88
B.1.6	SCR for REST.QoS.Subscriptions.Predefined Server .....	88
B.1.7	SCR for REST.QoS.IndSubscription.Predefined Server.....	88
B.1.8	SCR for REST.QoS.Subscriptions.Applied Server.....	89
B.1.9	SCR for REST.QoS.IndSubscription.Applied Server.....	89
B.1.10	SCR for REST.QoS.Notifications.Predefined Server .....	89
B.1.11	SCR for REST.QoS.Notifications.Applied Server.....	89
<b>APPENDIX C.</b>	<b>APPLICATION/X-WWW-FORM-URLENCODED REQUEST FORMAT FOR POST OPERATIONS (NORMATIVE).....</b>	<b>90</b>
<b>APPENDIX D.</b>	<b>JSON EXAMPLES (INFORMATIVE) .....</b>	<b>91</b>
<b>D.1</b>	<b>RETRIEVING A LIST OF PREDEFINED QoS FEATURES GENERALLY AVAILABLE TO THE END USER (SECTION 6.1.3.1).....</b>	<b>91</b>
<b>D.2</b>	<b>RETRIEVING A LIST OF PREDEFINED QoS FEATURES CURRENTLY AVAILABLE TO THE END USER (SECTION 6.1.3.2).....</b>	<b>92</b>
<b>D.3</b>	<b>RETRIEVING A LIST OF QoS FEATURES APPLIED ON THE END USER CONNECTION (SECTION 6.2.3.1).....</b>	<b>93</b>
<b>D.4</b>	<b>REQUEST TO APPLY A PREDEFINED QoS FEATURE ON THE END USER CONNECTION (SECTION 6.2.5.1).....</b>	<b>94</b>
<b>D.5</b>	<b>REQUEST TO APPLY A PREDEFINED QoS FEATURE ON THE END USER CONNECTION, RESPONSE WITH LOCATION OF CREATED RESOURCE (SECTION 6.2.5.2).....</b>	<b>95</b>
<b>D.6</b>	<b>REQUEST TO APPLY A CUSTOM QoS FEATURE ON THE END USER CONNECTION (SECTION 6.2.5.3) .....</b>	<b>96</b>
<b>D.7</b>	<b>APPLICATION SERVICE PROVIDER SPONSORING USER'S DATA USAGE (SECTION 6.2.5.4).....</b>	<b>99</b>
<b>D.8</b>	<b>REQUEST TO APPLY A CUSTOM QoS FEATURE ON THE END USER CONNECTION, WHICH IS NOT SUPPORTED BY THE SERVER (SECTION 6.2.5.5) .....</b>	<b>100</b>
<b>D.9</b>	<b>RETRIEVING AN INDIVIDUAL QoS FEATURE APPLIED ON THE END USER CONNECTION (SECTION 6.3.3.1) .....</b>	<b>101</b>
<b>D.10</b>	<b>UPDATING ATTRIBUTES FOR QoS FEATURE APPLIED ON THE END USER CONNECTION (SECTION 6.3.4.1).....</b>	<b>102</b>
<b>D.11</b>	<b>REMOVING A QoS FEATURE WHICH IS APPLIED TO THE END USER CONNECTION (SECTION 6.3.6.1) .....</b>	<b>103</b>
<b>D.12</b>	<b>RETRIEVING DURATION TIME FOR QoS FEATURE APPLIED ON THE END USER CONNECTION (SECTION 6.4.3.1).....</b>	<b>103</b>
<b>D.13</b>	<b>RETRIEVING IP FLOW STATUS ON MEDIA NUMBER LEVEL FOR QoS FEATURE APPLIED ON THE END USER CONNECTION (SECTION 6.4.3.2) .....</b>	<b>103</b>
<b>D.14</b>	<b>RETRIEVING IP FLOW STATUS ON IP FLOW NUMBER LEVEL FOR QoS FEATURE APPLIED ON THE END USER CONNECTION (SECTION 6.4.3.3) .....</b>	<b>104</b>
<b>D.15</b>	<b>UPDATING DURATION TIME FOR QoS FEATURE APPLIED ON THE END USER CONNECTION (SECTION 6.4.4.1)....</b>	<b>104</b>
<b>D.16</b>	<b>UPDATING IP FLOW STATUS ON IP FLOW NUMBER LEVEL FOR QoS FEATURE APPLIED ON THE END USER CONNECTION (SECTION 6.4.4.2).....</b>	<b>104</b>
<b>D.17</b>	<b>RETRIEVING ALL SUBSCRIPTIONS TO NOTIFICATIONS RELATED TO QoS FEATURES (SECTION 6.5.3.1) .....</b>	<b>105</b>
<b>D.18</b>	<b>RETRIEVING ALL SUBSCRIPTIONS TO NOTIFICATIONS ON AVAILABILITY OF PREDEFINED QoS FEATURES (SECTION 6.6.3.1).....</b>	<b>106</b>
<b>D.19</b>	<b>CREATING A NEW SUBSCRIPTION FOR NOTIFICATIONS ON AVAILABILITY OF PREDEFINED QoS FEATURES (SECTION 6.6.5.1).....</b>	<b>106</b>
<b>D.20</b>	<b>RETRIEVING AN INDIVIDUAL SUBSCRIPTIONS TO NOTIFICATIONS ON AVAILABILITY OF PREDEFINED QoS FEATURES (SECTION 6.7.3.1).....</b>	<b>107</b>
<b>D.21</b>	<b>CANCELLING A SUBSCRIPTION TO NOTIFICATIONS ON AVAILABILITY OF PREDEFINED QoS FEATURES (SECTION 6.7.6.1).....</b>	<b>108</b>
<b>D.22</b>	<b>RETRIEVING ALL SUBSCRIPTIONS TO NOTIFICATIONS ABOUT EVENTS OCCURRING FOR APPLIED QoS FEATURES (SECTION 6.8.3.1).....</b>	<b>108</b>

D.23 CREATING A NEW SUBSCRIPTION FOR NOTIFICATIONS ABOUT EVENTS OCCURRING FOR APPLIED QoS FEATURES (SECTION 6.8.5.1)..... 108

D.24 RETRIEVING AN INDIVIDUAL SUBSCRIPTION TO NOTIFICATIONS ABOUT EVENTS OCCURRING FOR APPLIED QoS FEATURES (SECTION 6.9.3.1)..... 109

D.25 CANCELLING A SUBSCRIPTION TO NOTIFICATIONS ABOUT EVENTS OCCURRING FOR APPLIED QoS FEATURES (SECTION 6.9.6.1)..... 110

D.26 NOTIFYING A CLIENT THAT A PREDEFINED QoS FEATURE IS AVAILABLE TO THE END USER (SECTION 6.10.5.1) ..... 110

D.27 NOTIFYING A CLIENT ABOUT AN EVENT OCCURRING FOR APPLIED QoS FEATURES (SECTION 6.11.5.1) ..... 110

APPENDIX E. PARLAY X OPERATIONS MAPPING (INFORMATIVE) ..... 112

APPENDIX F. LIGHT-WEIGHT RESOURCES (INFORMATIVE) ..... 113

APPENDIX G. AUTHORIZATION ASPECTS (NORMATIVE) ..... 114

G.1 USE WITH OMA AUTHORIZATION FRAMEWORK FOR NETWORK APIS..... 114

G.1.1 Scope values ..... 114

G.1.1.1 Definitions..... 114

G.1.1.2 Downscoping ..... 114

G.1.1.3 Mapping with resources and methods..... 115

G.1.2 Use of ‘acr:auth’ ..... 119

## Figures

Figure 1 Resource structure defined by this specification..... 15

Figure 2 Applying a predefined QoS feature on the end user connection ..... 39

Figure 3 Applying a custom QoS feature on the end user connection..... 40

Figure 4 Updating duration time for an applied QoS feature..... 41

Figure 5 Subscription to notifications on availability of predefined QoS features ..... 43

Figure 6 Subscription to notifications about events occurring for applied QoS features..... 45

Figure 7 A limit set for applied QoS feature has been reached ..... 46

## Tables

Table 1 Parlay X operations mapping ..... 112

Table 2 Scope values for RESTful Quality of Service API..... 114

Table 3 Required scope values for: Retrieval of available predefined QoS features ..... 116

Table 4 Required scope values for: Management of applied QoS features ..... 116

Table 5 Required scope values for: Retrieval of all subscriptions regarding QoS features ..... 116

Table 6 Required scope values for: Management of subscriptions to notifications for predefined QoS features..... 117

Table 7 Required scope values for: Management of subscriptions to notifications for applied QoS features..... 117

Table 8 Required scope values for: Notifications for predefined QoS features ..... 118

Table 9 Required scope values for: Notifications for applied QoS features ..... 118



# 1. Scope

This specification defines a RESTful API for Quality of Service using HTTP protocol bindings.

## 2. References

### 2.1 Normative References

- [3GPP\_TS\_29.214] “3rd Generation Partnership Project”; Technical Specification Group Core Network and Terminals; Policy and Charging Control over Rx reference point (Release 12), 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/>
- [REST\_NetAPI\_ACR] “RESTful Network API for Anonymous Customer Reference Management”, Open Mobile Alliance™, OMA-TS-REST\_NetAPI\_ACR-V1\_0, URL: <http://www.openmobilealliance.org/>
- [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\_QoS] “XML schema for the RESTful Network API for Quality of Service”, Open Mobile Alliance™, OMA-SUP-XSD\_rest\_netapi\_qos-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.ietf.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>
- [RFC4566] “SDP: Session Description Protocol”, M.Handley et. al, July 2006, 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>
- [RFC6466] “IANA Registration of “image” Media Type for the Session Description Protocol (SDP)”,G.Salgueiro, December 2011, URL:<http://www.ietf.org/rfc/rfc6466.txt>
- [SCRRULES] “SCR Rules and Procedures”, Open Mobile Alliance™, OMA-ORG-SCR\_Rules\_and\_Procedures, URL:<http://www.openmobilealliance.org/>
- [XMLSchema1] W3C XML Schema Definition Language (XSD) 1.1 Part 1: Structures Second Edition, W3C Recommendation 5 April 2012, URL: <http://www.w3.org/TR/xmlschema11-1/>
- [XMLSchema2] W3C XML Schema Definition Language (XSD) 1.1 Part 2: Datatypes, W3C Recommendation 5 April 2012, URL: <http://www.w3.org/TR/xmlschema11-2/>

### 2.2 Informative References

- [3GPP\_TS\_29.199-17] “3rd Generation Partnership Project”; Technical Specification Group Core Network and Terminals; “Open Service Access (OSA); Parlay X Web Services; Part 17: Application-driven Quality of Service (QoS) (Release 8)”, URL:<http://www.3gpp.org/>
- [OMADICT] “Dictionary for OMA Specifications”, Version 2.9, Open Mobile Alliance™, OMA-ORG-Dictionary-V2\_9, URL:<http://www.openmobilealliance.org/>
- [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].

<b>Client-side Notification URL</b>	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.
<b>Heavy-weight Resource</b>	A resource which is identified by a resource URL which is then used by HTTP methods to operate on the entire data structure representing the resource
<b>Light-weight Resource</b>	A subordinate resource of a Heavy-weight Resource which is identified by its own resource URL which is then used by HTTP methods to operate on a part of the data structure representing the Heavy-weight Resource. The Light-weight Resource URL can be seen as an extension of the Heavy-weight Resource URL. There could be several levels of Light-weight Resources below the ancestor Heavy-weight Resource, depending on the data structure
<b>Long Polling</b>	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.
<b>Notification Channel</b>	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.
<b>Notification Server</b>	A server that is capable of creating and maintaining Notification Channels.
<b>Server-side Notification URL</b>	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>JSON</b>	JavaScript Object Notation
<b>MIME</b>	Multipurpose Internet Mail Extensions
<b>OMA</b>	Open Mobile Alliance
<b>QoS</b>	Quality of Service
<b>REST</b>	REpresentational State Transfer
<b>SCR</b>	Static Conformance Requirements

---

<b>SIP</b>	Session Initiation Protocol
<b>TS</b>	Technical Specification
<b>URI</b>	Uniform Resource Identifier
<b>URL</b>	Uniform Resource Locator
<b>WP</b>	White Paper
<b>XML</b>	eXtensible Markup Language
<b>XSD</b>	XML Schema Definition

## 4. Introduction

The Technical Specification of the RESTful Network API for Quality of Service contains HTTP protocol bindings for QoS, 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).

### 4.1 Version 1.0

Version 1.0 of this specification supports the following operations:

- Retrieve a list of predefined QoS features available to an end user
- Request to apply a predefined QoS feature on an end user connection on a temporary basis
- Request to apply a specific QoS feature on an end user connection on a temporary basis
- Retrieve QoS features currently applied on an end user connection
- Update a QoS feature currently applied on an end user connection
- Update an individual attribute (e.g. duration) for QoS feature applied on an end user connection
- Manage subscriptions to notifications on availability of a predefined QoS feature(s) to an end user
- Notify a client when a predefined QoS feature(s) became available again to an end user
- Manage subscriptions to notifications about events occurring for QoS feature applied on an end user connection
- Notify a client about an event occurred for QoS feature applied on an end user connection

In addition, this specification provides:

- 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:auth” as a reserved keyword in an ACR

## 5. Quality of Service API definition

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

This Network API provides methods that enable an application to govern the quality of service (QoS) applied on end user network connections on a temporary basis. The application can request a certain QoS to be applied on the end user connection by using either a predefined QoS feature, or a custom QoS feature, depending on the method supported by the server's policy.

Predefined QoS features are defined by the service provider and as such have predetermined QoS properties (e.g. bandwidth, media type, priority, etc) associated with them for particular service usage. The API also provides support for an application to retrieve predefined QoS features available to an end user.

For some implementations, in order to identify the end user connection to which a certain QoS is to be applied, the applications might be required to provide valid IP addresses and port numbers associated with that connection. How the applications get to know such information is out of scope of this specification.

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 detailed specification for each of the resources. Each such subsection defines the resource, the request URL variables that are common for all HTTP methods, 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, while JSON examples are provided in Appendix D.

Section 7 contains fault definition details such as Service Exceptions and Policy Exceptions.

Appendix B provides the Static Conformance Requirements (SCR).

Appendix C provides application/x-www-form-urlencoded examples, where applicable.

Appendix E provides the operations mapping to a pre-existing baseline specification, where applicable.

Appendix F provides a list of all Light-weight Resources, where applicable.

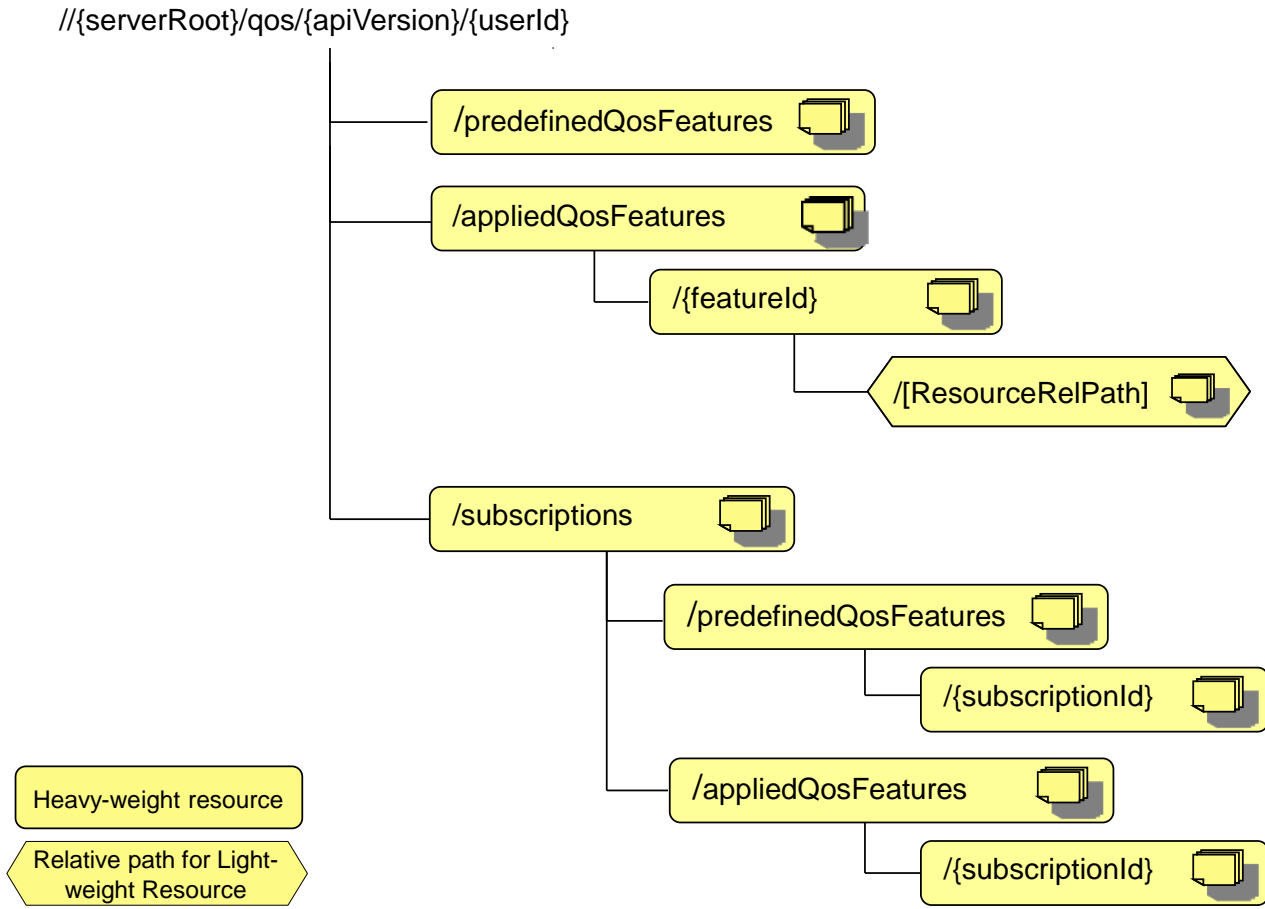
Appendix G defines authorization aspects to control access to the resources defined in this specification.

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 Network API for Quality of Service

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.



**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 a list of predefined QoS features that can be applied on the end user connection**

Resource	URL Base URL: http://{serverRoot}/qos/{apiVersion}/{userId}	Data Structures	HTTP verbs			
			GET	PUT	POST	DELETE
Predefined QoS features available to a user	/predefinedQosFeatures	PredefinedQosFeatureList	Retrieve a list of available predefined QoS features that can be applied on the end user connection on a temporary basis	no	no	no

**Purpose: To allow client to manage QoS features applied on an end user connection**

Resource	URL Base URL: http://{serverRoot}/qos/{apiVersion}/{userId}	Data Structures	HTTP verbs			
			GET	PUT	POST	DELETE
QoS features applied on an end user connection	/appliedQosFeatures	AppliedQosFeatureList (used for GET) QosFeatureData (used for POST) common:ResourceReference (OPTIONAL alternative for POST response)	Retrieve a list of all QoS features applied on an end user connection	no	Apply a new QoS feature on the end user connection on a temporary basis	no
Individual QoS feature applied on an end user connection	/appliedQosFeatures/{featureId}	QosFeatureData	Retrieve an individual QoS feature applied on an end user connection	Update an individual QoS feature applied on an end user connection	no	Delete (remove) an individual QoS feature applied on an end user connection



Resource	URL Base URL: http://{serverRoot}/qos/{apiVersion}/{userId}	Data Structures	HTTP verbs			
			GET	PUT	POST	DELETE
Individual attribute for QoS feature applied on an end user connection	/appliedQosFeatures/{featureId}/[ResourceRelPath]	The data structure corresponds to an element within QosFeatureData pointed out by request-URL.	Retrieve an individual attribute for QoS feature applied on an end user connection	Update an individual attribute for QoS feature applied on an end user connection	no	no

**Purpose: To allow client to manage all subscriptions for QoS features**

Resource	URL Base URL: http://{serverRoot}/qos/{apiVersion}/{userId}	Data Structures	HTTP verbs			
			GET	PUT	POST	DELETE
All subscriptions to QoS notifications	/subscriptions	QosFeaturesSubscriptionList	Retrieve a list of all active subscriptions to QoS feature notifications for the end user	no	no	no

**Purpose: To allow client to manage subscriptions to notifications about availability of predefined QoS features**

Resource	URL Base URL: http://{serverRoot}/qos/{apiVersion}/{userId}	Data Structures	HTTP verbs			
			GET	PUT	POST	DELETE
All subscriptions to notifications for predefined QoS features	/subscriptions/predefinedQosFeatures	PredefinedQosFeaturesSubscriptionList (used for GET)  PredefinedQosFeaturesSubscription (used for POST)  common:ResourceReference (OPTIONAL alternative for	Retrieve a list of active subscriptions to notifications for predefined QoS features	no	Create a new subscription to notifications for predefined QoS features availability	no

Resource	URL Base URL: http://{serverRoot}/qos/{apiVersion}/{userId}	Data Structures	HTTP verbs			
			GET	PUT	POST	DELETE
		POST response)				
Individual subscription to notifications for predefined QoS features	/subscriptions/predefinedQosFeatures/{subscriptionId}	PredefinedQosFeaturesSubscription	Retrieve an individual subscription to notifications for predefined QoS features	no	no	Cancel an individual subscription to notifications for predefined QoS features and stop corresponding notifications

**Purpose:** To allow client to manage subscriptions to notifications about events occurring for QoS features applied on an end user connection

Resource	URL Base URL: http://{serverRoot}/qos/{apiVersion}/{userId}	Data Structures	HTTP verbs			
			GET	PUT	POST	DELETE
All subscriptions to notifications for QoS features applied on an end user connection	/subscriptions/appliedQoSFeatures	AppliedQosFeaturesSubscriptionList (used for GET) AppliedQosFeaturesSubscription (used for POST) common:ResourceReference (OPTIONAL alternative for POST response)	Retrieve a list of active subscriptions to notifications for QoS features applied on an end user connection	no	Create a new subscription to notifications for QoS features applied on an end user connection	no

Resource	URL Base URL: http://{serverRoot}/qos/{apiVersion}/{userId}	Data Structures	HTTP verbs			
			GET	PUT	POST	DELETE
Individual subscription to notifications for QoS features applied on an end user connection	/subscriptions/appliedQoSFeatures/{subscriptionId}	AppliedQoSFeaturesSubscription	Retrieve an individual subscription to notifications for QoS features applied on an end user connection	no	no	Cancel an individual subscription to notifications for QoS features applied on an end user connection and stop corresponding notifications

**Purpose: To allow server to inform client about availability of predefined QoS features**

Resource	URL Base URL: <Specified by the client>	Data Structures	HTTP verbs			
			GET	PUT	POST	DELETE
Client notification about availability of predefined QoS features	Specified by client when the subscription is created or provisioned	PredefinedQoSFeaturesAvailabilityNotification	no	no	Notify client that predefined QoS feature(s) is available again	no

**Purpose: To allow server to inform client about an event occurred for QoS features applied on an end user connection**

Resource	URL Base URL: <Specified by the client>	Data Structures	HTTP verbs			
			GET	PUT	POST	DELETE
Client notification about an event occurred for QoS features applied on an end user connection	Specified by client when the subscription is created or provisioned	AppliedQosFeaturesNotification	no	no	Notify client about an event occurred for QoS features applied on an end user connection	no

## 5.2 Data Types

### 5.2.1 XML Namespaces

The XML namespace for the Quality of Service data types is:

urn:oma:xml:rest:netapi:qos:1

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

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

### 5.2.2 Structures

The subsections of this section define the data structures used in the Quality of Service API.

Some of the structures can be instantiated as so-called root elements, i.e. they define the type of a representation of a so-called Heavy-weight Resource.

The column [ResourceRelPath] in the tables below, if used, includes relative resource paths for Light-weight Resource URLs that are used to access individual elements in the data structure (so-called Light-weight Resources). A string from this column needs to be appended to the corresponding Heavy-weight Resource URL in order to create Light-weight Resource URL for that particular element in the data structure. "Not applicable" means that individual access to that element is not supported. The root element and data type of the resource associated with the [ResourceRelPath] are defined by the Element and Type columns in the row that defines the [ResourceRelPath].

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

#### 5.2.2.1 Type: PredefinedQoSFeatureList

This type represents a list of predefined QoS features.

Element	Type	Optional	Description
predefinedQoSFeature	PredefinedQoSFeature [0...unbounded]	Yes	Array of predefined QoS features available to a user.
resourceURL	xsd:anyURI	No	Self referring URL

A root element named predefinedQoSFeatureList of type PredefinedQoSFeatureList is allowed in response bodies.

#### 5.2.2.2 Type: PredefinedQoSFeature

This type represents a predefined QoS feature that can be applied on the end user connection on a temporary basis.

Element	Type	Optional	Description
predefinedQoSFeatureId	xsd:string	No	Contains an identifier of a predefined QoS feature.
predefinedQoSFeatureName	xsd:string	Yes	Contains a displayname of a predefined QoS feature (e.g. Gold, Silver, or VideoGold, VideoSilver, etc).

mediaInfo	PredefinedMediaInfo [0...unbounded]	Yes	If present, gives information about the media(s) associated with a predefined QoS feature.
reservationPriority	ReservationPriority	Yes	If present, gives information about the priority level the predefined QoS feature should have during a resource reservation.
link	common:Link [0..unbounded]	Yes	Link to other information relating to that particular predefined QoS feature.

### 5.2.2.3 Type: PredefinedMediaInfo

This type describes a media component for a predefined QoS feature.

Element	Type	Optional	Description
mediaType	MediaType	Yes	Specifies media type for a media component.
bandwidth	BandwidthInformation	Yes	Specifies bandwidth for a media component.
reservationPriority	ReservationPriority	Yes	Specifies the relative reservation priority for a media component within a QoS request.

### 5.2.2.4 Type: QosFeatureData

This type represents a QoS feature to be applied on the end user connection on a temporary basis. The data type can be used for implementation where only predefined QoS features are supported as well for the implementation where custom QoS features are supported.

Element	Type	Optional	[ResourceRelPath]	Description
clientCorrelator	xsd:string	Yes	Not applicable	<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 duplicate transaction creation 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>

predefinedQosFeatureId	xsd:string	Yes	Not applicable	<p>Contains an identifier of the predefined QoS feature that shall be applied for an end user connection.</p> <p>The parameter SHALL be specified only if predefined QoS feature is to be applied for an end user connection.</p> <p>Note that some servers may support only predefined QoS features and in such case this element MUST be included in the request to create resource.</p> <p>A predefined QoS feature is defined by operator and it has fixed QoS properties (e.g. bandwidth, media type, etc) that cannot be changed by an application.</p>
media	MediaComponentDescription [0...unbounded]	Yes	Not applicable	<p>Describes media components associated with a QoS feature that is to be applied for an end user connection.</p>
reservationPriority	ReservationPriority	Yes	Not applicable	<p>Specifies desired priority for handling of the request to apply selected QoS feature on an end user connection.</p> <p>If not specified, a default value "Low" is assumed.</p> <p>If element predefinedQosFeatureId is specified and that implicitly includes reservation priority information then this element SHOULD NOT be specified.</p>

duration	xsd:unsignedInt	Yes	duration	<p>Period of time (in seconds) this particular QoS feature is requested for (e.g. "3600", which is 60 minutes").</p> <p>If set to "0" (zero), a default duration time, which is specified by the service policy, will be used. If the parameter is omitted, the requested QoS feature will apply until the maximum duration time, which is specified by the service policy, unless the requested QoS feature is removed by deletion of the resource.</p> <p>This element MAY be given by the client during resource creation in order to signal the desired lifetime of the QoS feature. The server SHOULD return in this element the period of time for which the QoS feature will still be applied.</p> <p>If both "duration" and "volume" are supported by the server, then the first threshold that has been reached will apply.</p>
----------	-----------------	-----	----------	--



volume	xsd:unsignedInt	Yes	volume	<p>Specifies a data volume limit in kilobytes (1 kilobyte (kB) =1000 bytes) that shall be allocated for the requested QoS feature.</p> <p>If set to “0” (zero), a default data volume limit, which is specified by the service policy, will be used. If the parameter is omitted, the requested QoS feature will apply until the maximum data volume limit, which is specified by the service policy, is reached, unless the requested QoS feature is removed by deletion of the resource.</p> <p>This element MAY be given by the client during resource creation in order to signal the data volume limit that shall be allocated for the requested QoS feature. The server SHOULD return in this element the remaining data volume limit allocated for the QoS feature.</p> <p>If both “duration” and “volume” are supported by the server, then the first threshold that has been reached will apply.</p> <p>Note that if the server cannot track volume usage a policy exception POL1033 will be issued.</p>
defaultAction	DefaultAction	Yes	Not applicable	<p>Specifies a default action the server shall perform in case the allocated volume limit is reached, or QoS duration timer expires.</p> <p>If not specified, a default action “AutoCancellation” is assumed.</p>
sponsorId	xsd:string	Yes	Not applicable	<p>Identity of the party that shall be charged for the requested QoS feature (e.g. identity of the sponsor who pays user’s data usage in order to allow the user to access application service provider’s service).</p> <p>If the element is not present, then it is up to the operator policies to decide how the requested QoS feature will be charged.</p>

resourceURL	xsd:anyURI	Yes	Not applicable	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 be also included in responses to any HTTP method that returns an entity body, and in PUT requests.
-------------	------------	-----	----------------	---

A root element named qosFeatureData of type QoSFeatureData 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. [REST\_NetAPI\_Common] provides a recommendation regarding the generation of the value of this field.

### 5.2.2.5 Type: MediaComponentDescription

This type describes one media component. Note that if predefined QoS feature is used then some of the elements might have been already predefined there and therefore such elements SHOULD NOT be specified here again unless they are marked as mandatory.

Element	Type	Optionl	[ResourceRelPath]	Description
mediaNumber	xsd:unsignedInt	No	Not applicable	Specifies the ordinal number of the media component which is used to identify a particular media component among the other media components associated with the requested QoS.  It is a key property of a media component and SHALL NOT be altered when included in the Light-weight Resource URL.
mediaType	MediaType	Yes	Not applicable	Specifies the media type for a media component.
bandwidth	BandwidthInformation	Yes	media/{mediaNumber}/bandwidth	Specifies bandwidth requirements for a media component.
ipFlow	IpFlowDescription [0...unbounded]	Yes	Not applicable	Describes IP flows associated with a media component.
flowStatus	FlowStatus	Yes	media/{mediaNumber}/flowStatus	Specifies the status of IP flows associated with a media component.  If not specified, a default value "Enabled" is assumed.
reservationPriority	ReservationPriority	Yes	Not applicable	Specifies the relative reservation priority for a media component within a QoS request.

				If not specified, a default value "Low" is assumed.
--	--	--	--	---

### 5.2.2.6 Type: BandwidthInformation

This type describes the bandwidth requirements for a QoS feature.

Element	Type	Optional	Description
minUplinkBitRate	xsd:unsignedInt	Yes	Defines a minimum bit rate in bits per second in uplink (UL) direction.  The parameter is also referred as Guaranteed Bit Rate (GBR) in UL direction.
maxUplinkBitRate	xsd:unsignedInt	Yes	Defines a maximum bit rate in bits per second in uplink direction.
minDownlinkBitRate	xsd:unsignedInt	Yes	Defines a minimum bit rate in bits per second in downlink (DL) direction.  The parameter is also referred as Guaranteed Bit Rate (GBR) in DL direction.
maxDownlinkBitRate	xsd:unsignedInt	Yes	Defines a maximum bit rate in bits per second requested in downlink direction.

### 5.2.2.7 Type: IpFlowDescription

This type describes a set of IP flows (for uplink and/or downlink).

Element	Type	Optional	[ResourceRelPath]	Description
flowNumber	xsd:unsignedInt	No	Not applicable	Specifies the ordinal number of the IP flow(s) which is used to identify a particular IP flow(s) among the other IP flows for a media component.  It is a key property of IP flow and SHALL NOT be altered when included in the Light-weight Resource URL.
flowUsage	FlowUsage	Yes	Not applicable	Provides information about the usage of the IP flow (e.g. RTCP)
flowDescription	FlowDescription [0...2]	Yes	Not applicable	Describes the end points and protocols for IP flow(s) associated with a media component.  For bidirectional (two-ways) traffic, both uplink and downlink IP flows need to be described. For unidirectional (one-way) traffic, depending on the direction of the information flow, either downlink or uplink IP flow needs to be specified.

flowStatus	FlowStatus	Yes	media/{mediaNumber}/flow/{flowNumber}/flowStatus	<p>Specifies the status of IP flow(s) for a particular flow number.</p> <p>If specified, the value takes precedence over the value specified on the media component level for that particular flow number.</p> <p>If not specified, neither on media component level nor on flow number level, default value "Enabled" is assumed.</p>
------------	------------	-----	--	--

### 5.2.2.8 Type: FlowDescription

This data type describes end points and protocol for a single IP flow.

Element	Type	Optional	Description
direction	FlowDirection	No	Describes the direction of an IP flow from the user (identified by {userId} in the resource URL) perspective (e.g. Uplink).
protocol	ProtocolType	Yes	<p>Specifies the transport protocol to be used on the flow (e.g. TCP).</p> <p>If not specified, then it is assumed that any protocol over IP can be used for that particular flow.</p>
otherPartyIpAddress	IpAddress	Yes	IP address of the other party involved in communication with the user identified by {userId} in the resource URL. (An example of "other party" could be an application service/content provider)
otherPartyPortNumber	PortNumber	Yes	<p>Port number of the other party (e.g. 100).</p> <p>If not specified, depending on the protocol type used it could mean that either it is not required, or any port can be used.</p>
userIpAddress	IpAddress	Yes	IP address of the user identified by {userId} in the resource URL
userPortNumber	PortNumber	Yes	<p>Port number of the user (e.g. 10).</p> <p>If not specified, depending on the protocol type used it could mean that either it is not required, or any port can be used.</p>

### 5.2.2.9 Type: IpAddress

This data type represents one IP address.

Element	Type	Optional	Description
ipV4Address	xsd:string	Choice	Contains a valid routable IPv4 address.
ipV6Address	xsd:string	Choice	Contains a valid routable IPv6 address.

XSD modeling uses a “choice” to select either “ipV4Address” or “ipV6Address”, but not both of them.

### 5.2.2.10 Type: PortNumber

This data type represents port number(s).

Element	Type	Optional	Description
port	xsd:unsignedInt	Choice	Specifies a single port number
portRange	PortRange	Choice	Specifies a range of port numbers

XSD modeling uses a “choice” to select either “port” or “portRange”, but not both of them

### 5.2.2.11 Type: PortRange

This data type represents a range of port numbers.

Element	Type	Optional	Description
first	xsd:unsignedInt	No	Specifies the first port number in a range of port numbers.
last	xsd:unsignedInt	No	Specifies the last port number in a range of port numbers.

### 5.2.2.12 Type: AppliedQoSFeatureList

This type represents a list of QoS features applied on an end user connection on a temporary basis.

Element	Type	Optional	Description
qoSFeature	QoSFeatureData [0...unbounded]	Yes	Array of applied QoS features
resourceURL	xsd:anyURI	No	Self referring URL

A root element named appliedQoSFeatureList of type AppliedQoSFeatureList is allowed in response bodies.

### 5.2.2.13 Type: QoSFeaturesSubscriptionList

This type represents a list of all subscriptions to notifications about events concerning both predefined and applied QoS features.

Element	Type	Optional	Description
predefinedQoSFeaturesSubscriptionList	PredefinedQoSFeaturesSubscriptionList	Yes	Array of subscriptions for predefined QoS features
appliedQoSFeaturesSubscriptionList	AppliedQoSFeaturesSubscriptionList	Yes	Array of subscriptions for applied QoS features
resourceURL	xsd:anyURI	No	Self referring URL

A root element named qoSFeaturesSubscriptionList of type QoSFeaturesSubscriptionList is allowed in response bodies.

### 5.2.2.14 Type: PredefinesQoSFeaturesSubscriptionList

This type represents a list of subscriptions to notifications about events concerning predefined QoS features.

Element	Type	Optional	Description
predefinedQosFeaturesSubscription	PredefinedQosFeaturesSubscription [0..unbounded]	Yes	Array of subscriptions for predefined QoS features
resourceURL	xsd:anyURI	No	Self referring URL

A root element named predefinedQosFeaturesSubscriptionList of type PredefinedQosFeaturesSubscriptionList is allowed in response bodies.

### 5.2.2.15 Type: AppliedQosFeaturesSubscriptionList

This type represents a list of subscriptions to notifications about events concerning applied QoS features.

Element	Type	Optional	Description
appliedQosFeaturesSubscription	AppliedQosFeaturesSubscription [0..unbounded]	Yes	Array of subscriptions for applied QoS features
resourceURL	xsd:anyURI	No	Self referring URL

A root element named appliedQosFeaturesSubscriptionList of type AppliedQosFeaturesSubscriptionList is allowed in response bodies

### 5.2.2.16 Type: PredefinedQosFeaturesSubscription

This type represents a subscription to notifications about availability of predefined QoS features.

Element	Type	Optional	Description
callbackReference	Common:CallbackReference	No	Client's Notification URL and OPTIONAL callbackData
duration	xsd:unsignedInt	Yes	<p>Period of time (in seconds) notifications are provided for.</p> <p>If set to "0" (zero), a default duration time, which is specified by the service policy, will be used. If the parameter is omitted, the notifications will continue until the maximum duration time, which is specified by the service policy, unless the notifications are stopped by deletion of subscription for notifications.</p> <p>This element MAY be given by the client during resource creation in order to signal the desired lifetime of the subscription. The server SHOULD return in this element the period of time for which the subscription will still be valid.</p>

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 element MAY be present.</p> <p>Note: this allows the client to recover from communication failures during resource creation and therefore avoids duplicate subscriptions in such situations.</p> <p>In case the element 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>
predefinedQoSFeatures	xsd:string [1...unbounded]	No	<p>Contains a list of predefined QoS feature identities, temporary unavailable to a user, for which the application wants to be notified when they became available to the user again.</p>
multipleNotificationsRequested	xsd:boolean	Yes	<p>This OPTIONAL element signals whether a notification shall be sent after each occurrence where the specified predefined QoS feature became available again (after period of unavailability), or just after the first occurrence.</p> <p>Default value is “false” which means that a notification SHALL be sent only after the first occurrence where the QoS feature became available again. After the notifications have been sent for all features included in the subscription, the server SHALL remove the subscription.</p> <p>Note that a server according to its policy MAY NOT support multiple notifications (element value set to “true”) and in such case POL1035 will be returned.</p>
resourceURL	xsd:anyURI	Yes	<p>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.</p>

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



### 5.2.2.17 Type: AppliedQoSFeaturesSubscription

This type represents a subscription to notifications about events occurring for applied QoS features on the end user connection.

Element	Type	Optional	Description
callbackReference	Common:CallbackReference	No	Client's Notification URL and OPTIONAL callbackData
duration	xsd:unsignedInt	Yes	<p>Period of time (in seconds) notifications are provided for. If set to "0" (zero), a default duration time, which is specified by the service policy, will be used. If the parameter is omitted, the notifications will continue until the maximum duration time, which is specified by the service policy, unless the notifications are stopped by deletion of subscription for notifications.</p> <p>This element MAY be given by the client during resource creation in order to signal the desired lifetime of the subscription. The server SHOULD return in this element the period of time for which the subscription will still be valid.</p>
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 element MAY be present.</p> <p>Note: this allows the client to recover from communication failures during resource creation and therefore avoids duplicate subscriptions in such situations.</p> <p>In case the element 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>
eventType	QoSEvent [0...unbounded]	Yes	<p>Specifies QoS events for which notifications shall be sent.</p> <p>If not specified, then notifications shall be sent for all QoS events concerning applied QoS features.</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.
-------------	------------	-----	---

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

### 5.2.2.18 Type: PredefinedQosAvailabilityNotification

This type represents a notification that predefined QoS feature(s) that were temporary unavailable become available again.

Element	Type	Optional	Description
callbackData	xsd:string	Yes	The 'callbackData' element if it was passed by the application in the 'callbackReference' element when creating a subscription to notifications about QoS events.  See [REST_NetAPI_Common]
predefinedQosFeatureId	xsd:string [1...unbounded]	No	Contains a list of predefined QoS features that became available to an end user again.
link	Common:Link [0..unbounded]	Yes	Links to other resources that are in relationship to the notification.  The server SHOULD also include a link to the related subscription.

A root element named predefinedQosAvailabilityNotification of type PredefinedQosAvailabilityNotification is allowed in notification request bodies.

### 5.2.2.19 Type: AppliedQosFeaturesNotification

This type represents a notification about an event occurred for QoS feature(s) applied on an user connection.

Element	Type	Optional	Description
callbackData	xsd:string	Yes	The 'callbackData' element if it was passed by the application in the 'callbackReference' element when creating a subscription to notifications about QoS events.  See [REST_NetAPI_Common]
eventType	QosEvent	No	Type of event
eventDescription	xsd:string	Yes	Textual description of the event

link	Common:Link [0..unbounded]	Yes	Links to other resources that are in relationship to the notification.  The server SHOULD also include a link to the related subscription.
------	-------------------------------	-----	--

A root element named appliedQoSFeaturesNotification of type AppliedQoSFeaturesNotification is allowed in notification request bodies.

## 5.2.3 Enumerations

The subsections of this section define the enumerations used in the Quality of Service API.

### 5.2.3.1 Enumeration: DefaultAction

This enumeration defines possible values that can be used to specify a default action that is to be performed automatically by a server in case that volume data limit is reached, or QoS duration timer expires.

Enumeration	Description
AutoRenewal	The application data traffic flow continues with the same QoS feature.
AutoCancellation	The application data traffic flow is stopped and the requested QoS is removed.

### 5.2.3.2 Enumeration: FlowDirection

This enumeration defines possible values to be used to describe an IP flow direction.

Enumeration	Description
Uplink	A unidirectional link for transmission of data; in general the direction from user equipment to network.
Downlink	A unidirectional link for transmission of data; in general the direction from network to user equipment.

### 5.2.3.3 Enumeration: FlowStatus

This enumeration defines possible values to be used to specify the status for IP flow(s).

Enumeration	Description
EnabledUplink	This value shall be used to enable uplink IP flow and to disable associated downlink IP flow.
EnabledDownlink	This value shall be used to enable downlink IP flow and to disable associated uplink IP flow.
Enabled	The value shall be used to enable IP flows in both directions.
Disabled	The value shall be used to disable IP flows in both directions.
Removed	This value shall be used to remove IP flows in both directions.  Note that if the removed IP flows were the last one for that particular QoS feature, the QoS feature will be terminated.

### 5.2.3.4 Enumeration: FlowUsage

This enumeration defines possible values to be used to describe the usage of an IP flow.

Enumeration	Description
RTCP	This value is used to indicate that an IP flow is used to transport RTCP.
Signalling	This value is used to indicate that the IP flow is used to transport signalling protocols.

### 5.2.3.5 Enumeration: MediaType

This enumeration defines possible media types that can be specified for the end user connection (IP flow) for which a QoS is requested. The enumeration values listed below corresponds to media type definitions in [3GPP\_TS\_29.214], [RFC4566], and [RFC6466].

Enumeration	Description
Audio	Indicates that the requested QoS feature will be used for voice conversation.
Video	Indicates that the requested QoS feature will be used for video applications (e.g. live streaming).
Data	Indicates that the requested QoS feature will be used for a bulk-data transfer such as multicasting of program executables which will not typically be displayed to the user.  Note that this media type is included in [3GPP_TS_29.214] although it has been removed from [RFC4566] to which [3GPP_TS_29.214] refers.
Application	Indicates that the requested QoS feature will be used for a media flow such as whiteboard information etc.
Control	Indicates that the requested QoS feature will be used for control signalling purposes (e.g. an additional conference control channel for the session).  Note that this media type is included in [3GPP_TS_29.214] although it has been removed from [RFC4566] to which [3GPP_TS_29.214] refers.
Text	Indicates that the requested QoS feature will be used for text based services.
Message	Indicates that the requested QoS feature will be used for messaging services/applications (e.g. e-mail, chat).
Image	Indicates that the requested QoS feature will be used for a media stream whose content consists of one or more images that require appropriate hardware to display.
Other	Indicates that the requested QoS feature will be used for other purposes.

### 5.2.3.6 Enumeration: ProtocolType

This enumeration defines possible values for transport protocol used for a media IP flow on an end user connection.

Enumeration	Description
TCP	Used to indicate that TCP is used as transport protocol for a media IP flow.

UDP	Used to indicate that UDP is used as transport protocol for a media IP flow.
SCTP	Used to indicate that SCTP is used as transport protocol for a media IP flow.

### 5.2.3.7 Enumeration: QosEvent

This enumeration defines possible values that can be used to describe an event occurring for applied QoS feature(s) on the end user connection.

Enumeration	Description
AbnormalConnectionTermination	End user connection(s) terminated abnormally because of a fault in the network causing all applied QoS features that were active on the connection(s) to be released as well.
NormalConnectionTermination	End user connection(s) terminated normally – e.g. user(s) have logged off – causing all applied QoS features active on the connection(s) to be automatically released.
AppliedQoSFeatureReleased	Applied QoS feature that was active on the end user connection has been released because the threshold set by one of the service attributes (e.g. elapsed duration) has been reached.
AppliedQoSFeatureRenewed	Applied QoS feature that was active on the end user connection has been renewed and continues with the same settings.

### 5.2.3.8 Enumeration: ReservationPriority

This enumeration defines possible values that can be used to specify reservation priority level for a QoS feature that is applied on the end user connection.

Enumeration	Description
High	Priority level recommended for services that shall be given a high priority for resource reservation during the network congestion.
Medium	Priority level recommended for services that shall be given a medium priority for resource reservation during the network congestion.
Low	Priority level recommended for services that do not require priority for resource reservation during the network congestion.

## 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):

- PredefinedQoSFeature
- AppliedQoSFeatureList
- QoSFeatureData
- QoSFeaturesSubscriptionList
- PredefinedQoSFeaturesSubscriptionList
- AppliedQoSFeaturesSubscriptionList

- PredefinedQosFeaturesSubscription
- AppliedQosFeaturesSubscription

These values indicate the kind of resource that the link points to.

## 5.3 Sequence Diagrams

The following subsections describe the resources, methods and steps involved in typical scenarios.

In a sequence diagram, a step which involves delivering a notification is labeled with “POST or NOTIFY”, where “POST” refers to delivery via the HTTP POST method, and “NOTIFY” refers to delivery using the Notification Channel [REST\_NetAPI\_NotificationChannel].

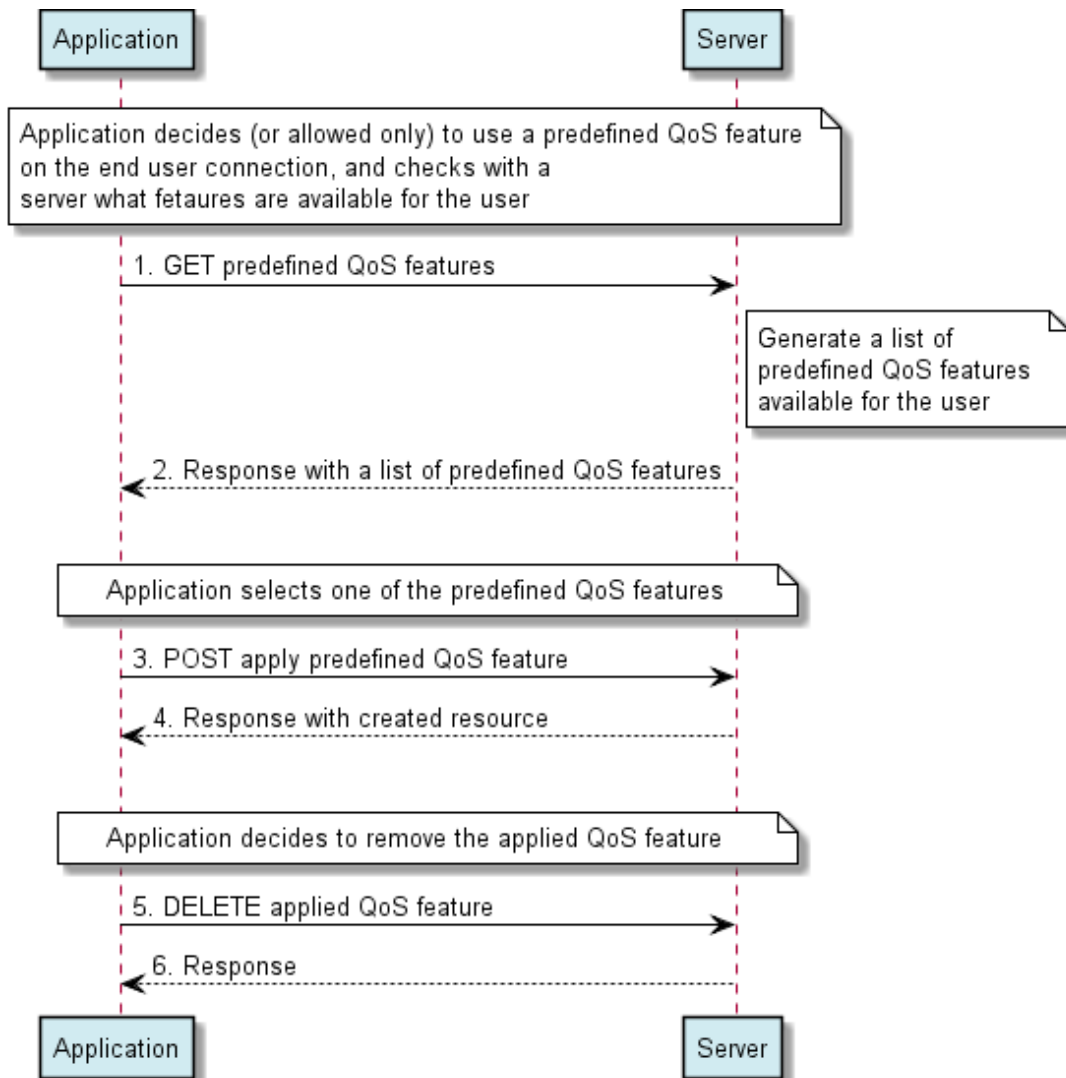
### 5.3.1 Applying a predefined QoS feature on the end user connection

The figure below shows a scenario where a client requested a predefined QoS feature to be applied on the end user connection on a temporary basis.

A predefined QoS features are features that have defined QoS properties (for example: max/min bit rates up/down link, media type, priority, etc), by the operator (service provider) and as such they are packed under different names (e.g. gold, silver, bronze) and then offered to their users.

The resources:

- To retrieve predefined QoS features available for the end user connection, read the resource under **http://{serverRoot}/qos/{apiVersion}/{userId}/predefinedQosFeatures**
- To request a predefined QoS feature to be applied on the end user connection, create a new resource under **http://{serverRoot}/qos/{apiVersion}/{userId}/appliedQosFeatures**
- To remove a predefined QoS feature that has been applied on an end user connection, delete the resource under **http://{serverRoot}/qos/{apiVersion}/{userId}/appliedQosFeatures/{featureId}**



**Figure 2 Applying a predefined QoS feature on the end user connection**

Outline of the flows:

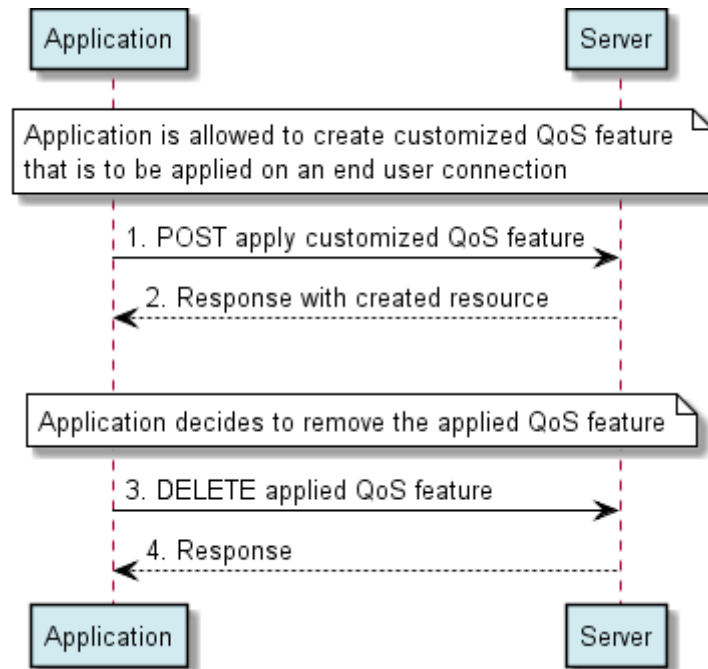
1. The application requests the server to provide a list of predefined QoS features available to a user, by using GET method on the resource.
2. The server returns in the response to the GET request a list of available predefined QoS features with their names, and their properties (this is dependent on the server policy)
3. The application requests the server to apply a selected predefined QoS feature on the end user connection.
4. The server returns in the response to the POST request a resource URL which can be used in subsequent HTTP methods to identify the QoS feature applied on the end user connection.
5. The application decides to remove the QoS feature applied on the end user connection by using DELETE method on the resource identified by the resource URL which has been received in step 4.
6. The server returns an HTTP response.

### 5.3.2 Applying a custom QoS feature on the end user connection

The figure below shows a scenario where a client requested specific properties for a QoS feature that is to be applied on the end user connection on a temporary basis.

The resources:

- To request a specific QoS feature properties to be applied on the end user connection, create a new resource under **http://{serverRoot}/qos/{apiVersion}/{userId}/appliedQosFeatures**
- To remove a temporary QoS feature that has been applied on an end user connection, delete the resource under **http://{serverRoot}/qos/{apiVersion}/{userId}/appliedQosFeatures/{featureId}**



**Figure 3 Applying a custom QoS feature on the end user connection**

Outline of the flows:

1. The application requests the server to apply a QoS feature with specific properties (custom QoS feature) on the end user connection, by using POST request which includes desired QoS feature properties.
2. The server returns in the response to the POST request a resource URL which can be used in subsequent HTTP methods to identify the QoS feature applied on the end user connection.
3. The application decides to remove the QoS feature applied on the end user connection by using DELETE method on the resource identified by the resource URL received in the step 2.
4. The server returns an HTTP response.

### 5.3.3 Updating a QoS feature applied on the end user connection

The figure below shows a scenario where a client updates parameters for the QoS feature that has been applied on the end user connection.

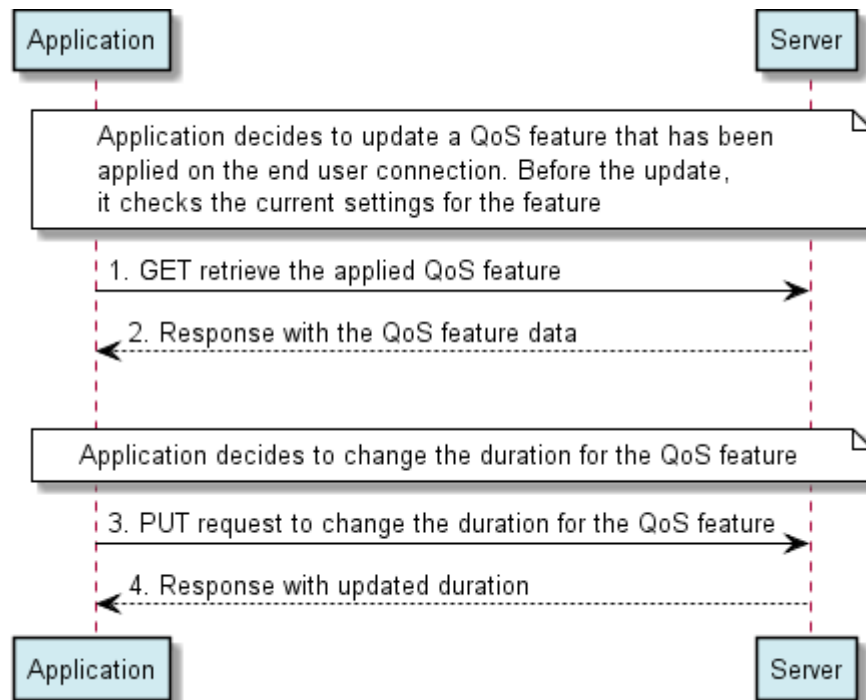
The resources:

- To retrieve a specified QoS features available for an end user connection, read the resource under **http://{serverRoot}/qos/{apiVersion}/{userId}/appliedQosFeatures/{featureId}**
- To update parameters for a QoS feature that has been applied on an end user connection, update the resource under **http://{serverRoot}/qos/{apiVersion}/{userId}/appliedQosFeatures/{featureId}**



- To update an individual QoS parameter (attribute) the following Light-weight Resource is used:  
**http://{serverRoot}/qos/{apiVersion}/{userId}/appliedQoSFeatures/{featureId}/[ResourceRelPath]**

Where [ResourceRelPath] is a light-weight relative resource URL, and it should be replaced with a string corresponding to the parameter that needs to be updated. Those strings are defined in [ResourceRelPath] column in the tables in section 5.2.2.



**Figure 4 Updating duration time for an applied QoS feature**

Outline of the flows:

1. The application requests the server to retrieve a specified QoS feature that has been applied on an end user connection, by using GET method on the resource identified by the resource URL received in the response from the server when the resource was created. Alternatively, if the application is interested in a particular QoS feature parameter then the application can use a Light-weight Resource for retrieving of that particular QoS feature parameter (assuming that such resource is defined and supported by the server).
2. The server in the response to the GET request returns parameters for the specified QoS feature.
3. The application decides to change the duration that was initially specified for the QoS feature by using PUT method on the resource identified by the resource URL received in the response from the server when the resource was created. Alternatively, the application can use PUT method on a Light-weight Resource for updating duration time (if such option is supported by the server). In that case URL parameter [ResourceRelPath] shall be replaced with a string "duration"
4. The server returns an HTTP response confirming updated duration.

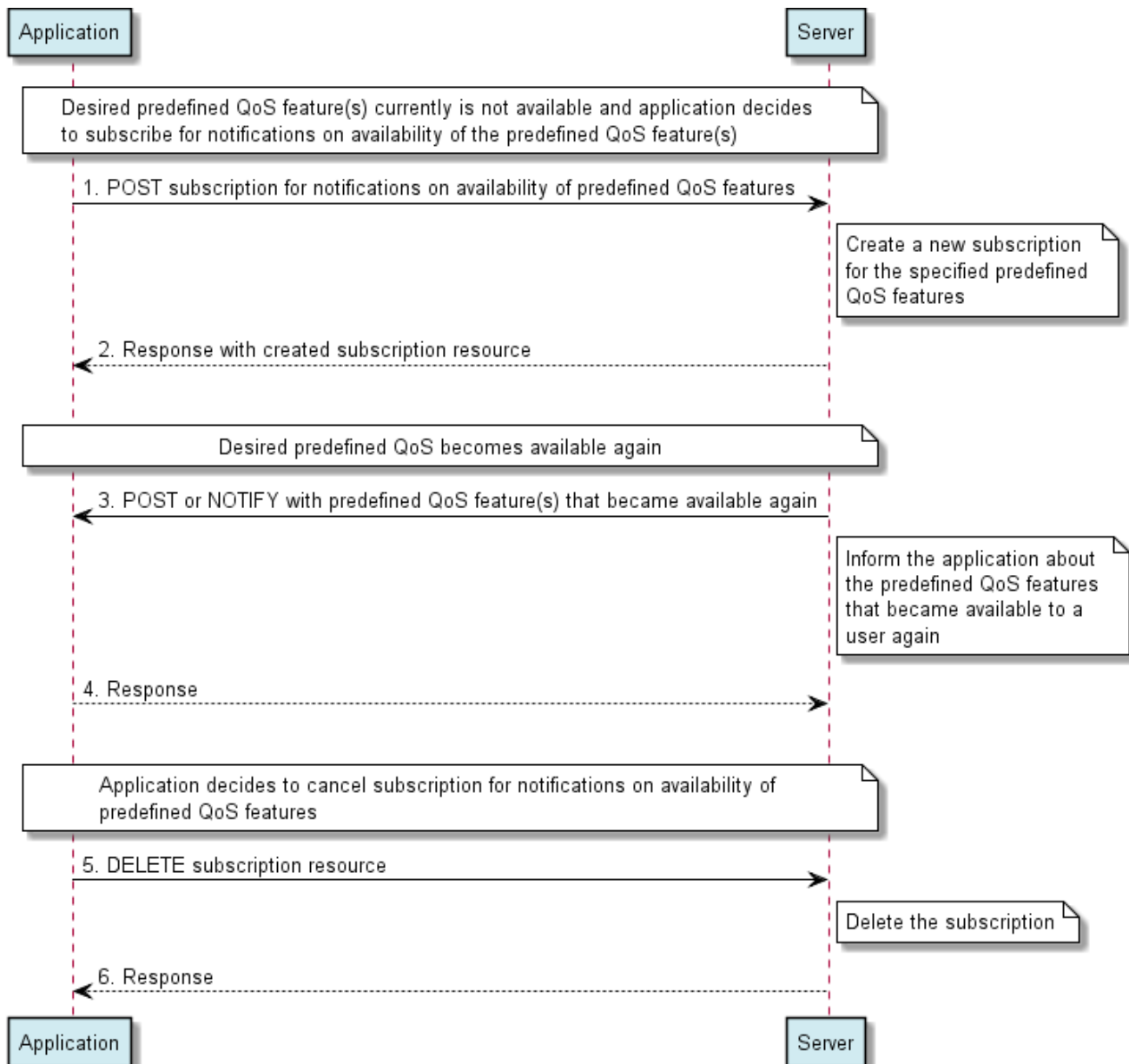
### 5.3.4 Subscription to notifications on availability of predefined QoS features

The figure below shows a scenario for a subscription to notifications on availability of predefined QoS features that are currently not available to the end user due to underlying conditions (e.g. user's location, network condition, time of day, etc).

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 create a new subscription to notifications on availability of predefined QoS features, create a new resource under **http://{serverRoot}/qos/{apiVersion}/{userId}/subscriptions/predefinedQosFeatures**
- To terminate (remove) an individual subscription to notifications on availability of predefined QoS features, delete the resource under **http://{serverRoot}/qos/{apiVersion}/{userId}/subscriptions/predefinedQosFeatures/{subscriptionId}**
- The notifications from the server on availability of predefined QoS features are done on the notification URL provided by the application during the subscription to notifications.



**Figure 5 Subscription to notifications on availability of predefined QoS features**

Outline of the flows:

1. An application subscribes to notifications on availability of a particular predefined QoS feature(s) by using the POST method to the resource containing subscriptions for predefined QoS features.
2. The application receives the result resource URL containing the subscriptionId.
3. The application receives notification that desired predefined QoS feature(s) has become available again to the end user.
4. The application returns an HTTP response.
5. The application stops receiving notifications by using DELETE method with the resource URL containing the subscriptionId.
6. The server returns an HTTP response.

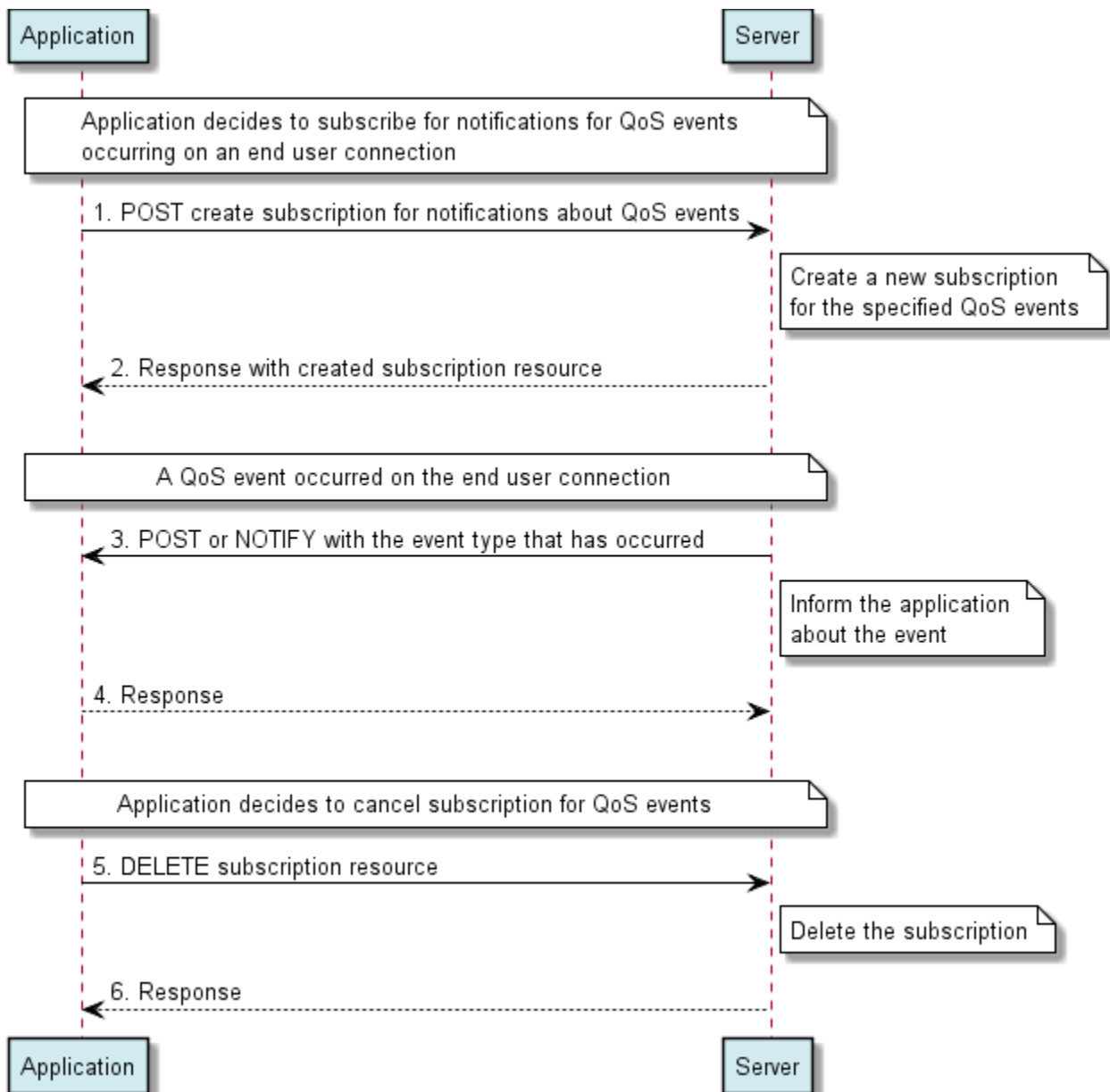
### 5.3.5 Subscription to notifications about events occurring for applied QoS features

The figure below shows a scenario for a subscription for notifications about events relating to QoS features applied on an end user connection.

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 create a new subscription for notifications about QoS events, create a new resource under **http://{serverRoot}/qos/{apiVersion}/{userId}/subscriptions/appliedQosFeatures**
- To terminate (remove) an individual subscription for QoS event notifications, delete the resource under **http://{serverRoot}/qos/{apiVersion}/{userId}/subscriptions/appliedQosFeatures/{subscriptionId}**
- The notifications from the server about QoS events are done on the notification URL provided by the application during the subscription to notifications.



**Figure 6 Subscription to notifications about events occurring for applied QoS features**

Outline of the flows:

1. An application subscribes to notifications about the events occurring for the applied QoS features by using the POST method to the resource containing subscriptions for applied QoS features.
2. The application receives the result resource URL containing the subscription Id.
3. The application receives notification that a certain QoS event has occurred on the end user connection.
4. The application returns an HTTP response.
5. The application stops receiving notifications by using DELETE method with the resource URL containing the subscription Id.
6. The server returns an HTTP response.

### 5.3.6 A limit for applied QoS feature has been reached

The figure below shows a scenario where a limit (either duration timed out, or allocated volume) for a particular QoS feature applied on the end user connection has been reached.

The resources:

- The notification about the action taken by the server is done on the notification URL provided by the application during the subscriptions to notifications.

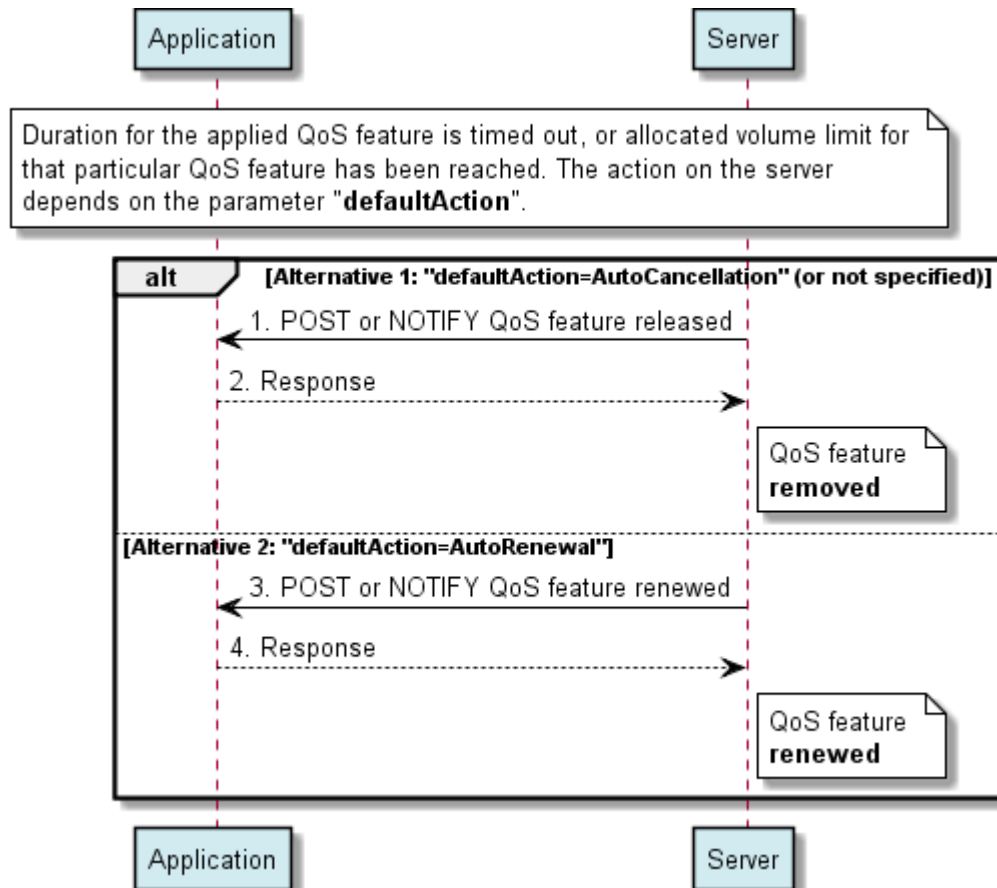


Figure 7 A limit set for applied QoS feature has been reached

Outline of the flows:

A limit for applied QoS feature (either duration timed out, or allocated volume) has been reached and the application receives a notification from the server. There are 2 possible alternatives depending on the settings for “defaultAction” parameter:

Alternative 1: In the request to apply that particular QoS feature, the application had set the parameter “defaultAction” to “AutoCancellation” or the parameter was completely omitted.

1. The application receives a notification from the server that the QoS feature is terminated and the resource has been removed.
2. The application returns an HTTP response.

Alternative 2: In the request to apply that particular QoS feature, the application had set the parameter “defaultAction” to “AutoRenewal”.

3. The application receives a notification from the server that the QoS feature is renewed and continues with the same settings
4. The application returns an HTTP response.

## 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):

- 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 a user identifier (e.g. address, participantAddress, etc.) 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 a user identifier (e.g. address, userId, etc) of type anyURI is in the form of an Anonymous Customer Reference (ACR), it **MUST** be defined according to Appendix H of [REST\_NetAPI\_ACR].
  - The ACR ‘auth’ 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. 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: Predefined QoS features available to a user

The resource used is:

**http://{serverRoot}/qos/{apiVersion}/{userId}/predefinedQoSFeatures**

This resource is used by a client for retrieving a list of predefined QoS features that are available to an end user.

#### 6.1.1 Request URL variables

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

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
userId	Identifier of the user on whose behalf the application acts 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 Quality of Service API, see section 7.

### 6.1.3 GET

This operation is used for retrieving a list of predefined QoS that are available to an end user.

Supported parameters in the query string of the Request URL are:

Name	Type/Values	Optional	Description
currentlyAvailableOnly	xsd:boolean	Yes	Indicates what type the request is. If set to "True" then the request is to retrieve predefined QoS feature that are available to the end user at the time the request is received by the server. If set to "False" then the request is to retrieve predefined QoS features that are generally available to the end user. Default value is "False".
mediaType	MediaType	Yes	Indicates for which media type the predefined QoS features are requested (e.g. Video). If not specified then the predefined QoS features for all media types will be retrieved. Possible values to be used are defined in 5.2.3.5.

#### 6.1.3.1 Example 1: Retrieving a list of predefined QoS features generally available to the end user (Informative)

##### 6.1.3.1.1 Request

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

##### 6.1.3.1.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: nnnn
Date: Mon, 28 Jul 2011 17:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<qos:predefinedQosFeatureList xmlns:qos="urn:oma:xml:rest:netapi:qos:1">
  <predefinedQosFeature>
    <predefinedQosFeatureId>hdv1080</predefinedQosFeatureId>
    <predefinedQosFeatureName>VideoGold</predefinedQosFeatureName>
    <mediaInfo>
      <mediaType>Video</mediaType>
      <bandwidth>
        <minUplinkBitRate>7000000</minUplinkBitRate>
```

```

    <minDownlinkBitRate> 7000000</minDownlinkBitRate>
  </bandwidth>
</mediaInfo>
<reservationPriority>Medium</reservationPriority>
<link rel="PredefinedQoSFeature"
  href="http://example.com/qos/predefinedQoSFeatures/hdv1080"/>
</predefinedQoSFeature>
<predefinedQoSFeature>
  <predefinedQoSFeatureId>dvdv768</predefinedQoSFeatureId>
  <predefinedQoSFeatureName>VideoSilver</predefinedQoSFeatureName>
  <mediaInfo>
    <mediaType>Video</mediaType>
    <bandwidth>
      <minUplinkBitRate>4000000</minUplinkBitRate>
      <minDownlinkBitRate> 4000000</minDownlinkBitRate>
    </bandwidth>
  </mediaInfo>
  <reservationPriority>Medium</reservationPriority>
  <link rel="PredefinedQoSFeature"
    href=" http://example.com/qos/predefinedQoSFeatures/dvdv768"/>
</predefinedQoSFeature>
<predefinedQoSFeature>
  <predefinedQoSFeatureId>audio16</predefinedQoSFeatureId>
  <predefinedQoSFeatureName>AudioGold</predefinedQoSFeatureName>
  <mediaInfo>
    <mediaType>Audio</mediaType>
  </mediaInfo>
  <reservationPriority>Medium</reservationPriority>
  <link rel="PredefinedQoSFeature"
    href="http://example.com/qos/predefinedQoSFeatures/audio16"/>
</predefinedQoSFeature>
<predefinedQoSFeature>
  <predefinedQoSFeatureId>avg8768</predefinedQoSFeatureId>
  <predefinedQoSFeatureName>GamingSilver</predefinedQoSFeatureName>
  <mediaInfo>
    <mediaType>Video</mediaType>
    <bandwidth>
      <minUplinkBitRate>7000000</minUplinkBitRate>
      <minDownlinkBitRate> 7000000</minDownlinkBitRate>
    </bandwidth>
  </mediaInfo>
  <reservationPriority>Low</reservationPriority>
  <link rel="PredefinedQoSFeature"
    href=" http://example.com/qos/predefinedQoSFeatures/avg8768"/>
</predefinedQoSFeature>
<resourceURL>http://example.com/exampleAPI/qos/v1/tel%3A%2B19585550100/predefinedQoSFeatures</resourceURL>
</qos:predefinedQoSFeatureList>

```

### 6.1.3.2 Example 2: Retrieving a list of predefined QoS features currently available to the end user (Informative)

Following is an example where a client is interested in to know which predefined QoS features are available to the end user at that time at that location.

### 6.1.3.2.1 Request

```
GET /exampleAPI/qos/v1/tel%3A%2B19585550100/predefinedQosFeatures?currentlyAvailableOnly="True" HTTP/1.1
Accept: application/xml
Host: example.com
```

### 6.1.3.2.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: nnnn
Date: Mon, 28 Jul 2011 17:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<qos:predefinedQosFeatureList xmlns:qos="urn:oma:xml:rest:netapi:qos:1">
  <predefinedQosFeature>
    <predefinedQosFeatureId>dvdv768</predefinedQosFeatureId>
    <predefinedQosFeatureName>VideoSilver</predefinedQosFeatureName>
    <mediaInfo>
      <mediaType>Video</mediaType>
      <bandwidth>
        <minUplinkBitRate>4000000</minUplinkBitRate>
        <minDownlinkBitRate> 4000000</minDownlinkBitRate>
      </bandwidth>
    </mediaInfo>
    <reservationPriority>Medium</reservationPriority>
    <link rel="PredefinedQosFeature"
      href="http://example.com/qos/predefinedQosFeatures/dvdv768"/>
  </predefinedQosFeature>
  <predefinedQosFeature>
    <predefinedQosFeatureId>audio16</predefinedQosFeatureId>
    <predefinedQosFeatureName>AudioGold</predefinedQosFeatureName>
    <mediaInfo>
      <mediaType>Audio</mediaType>
    </mediaInfo>
    <reservationPriority>Medium</reservationPriority>
    <link rel="PredefinedQosFeature"
      href="http://example.com/qos/predefinedQosFeatures/audio16"/>
  </predefinedQosFeature>
  <resourceURL>http://example.com/exampleAPI/qos/v1/tel%3A%2B19585550100/predefinedQosFeatures</resourceURL>
</qos:predefinedQosFeatureList>
```

### 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 [RFC2616].

### 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 [RFC2616].

### 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 [RFC2616].

## 6.2 Resource: QoS features applied on an end user connection

The resource used is:

**http://{serverRoot}/qos/{apiVersion}/{userId}/appliedQosFeatures**

This resource is used by a client for retrieving a list of applied QoS features an end user connection as well as for requesting a certain QoS feature to be applied to the end user connection.

### 6.2.1 Request URL variables

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

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
userId	Identifier of the user on whose behalf the application acts 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 Quality of Service API, see section 7.

### 6.2.3 GET

This operation is used for retrieving a list of QoS features that have been applied on an end user connection.

#### 6.2.3.1 Example: Retrieving a list of QoS features applied on the end user connection (Informative)

##### 6.2.3.1.1 Request

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

##### 6.2.3.1.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: nnnn
Date: Mon, 28 Jul 2011 17:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<qos:appliedQosFeatureList xmlns:qos="urn:oma:xml:rest:netapi:qos:1">
  <qosFeature>
    <clientCorrelator>v1234</clientCorrelator>
    <predefinedQosFeatureId>hdv1080</predefinedQosFeatureId>
    <duration>5400</duration>
```

```

    <resourceURL>http://example.com/exampleAPI/qos/v1/tel%3A%2B19585550100/appliedQosFeatures/hdv1080</resourceURL>
  </qosFeature>
</qosFeature>
  <clientCorrelator>gm1234</clientCorrelator>
  <media>
    <mediaNumber>1</mediaNumber>
    <mediaType>Application</mediaType>
  </media>
  <duration>6200</duration>
  <resourceURL>http://example.com/exampleAPI/qos/v1/tel%3A%2B19585550100/appliedQosFeatures/app001</resourceURL>
</qosFeature>
<resourceURL>http://example.com/exampleAPI/qos/v1/tel%3A%2B19585550100/appliedQosFeatures</resourceURL>
</qos:appliedQosFeatureList>

```

## 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 [RFC2616].

## 6.2.5 POST

This operation is used by an application to request a certain QoS feature to be applied on an end user connection. In the request, the application can specify either a predefined QoS feature or a custom QoS feature depending on the method supported by server.

To find out predefined QoS features available for a user, the application can use a method described in 6.1.3.

### 6.2.5.1 Example 1: Request to apply a predefined QoS feature on the end user connection (Informative)

In the following example, the predefined QoS feature has predetermined media type, bandwidth information, and reservation priority. The feature is requested for streaming a video from a service provider to the user.

#### 6.2.5.1.1 Request

```

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

```

```

<?xml version="1.0" encoding="UTF-8"?>
<qos:qosFeatureData xmlns:qos="urn:oma:xml:rest:netapi:qos:1">
  <clientCorrelator>v1234</clientCorrelator>
  <predefinedQosFeatureId>hdv1080</predefinedQosFeatureId>
  <media>
    <mediaNumber>1</mediaNumber>
    <ipFlow>
      <flowNumber>1</flowNumber>
      <flowDescription>
        <direction>Downlink</direction>
        <protocol>TCP</protocol>
        <otherPartyIpAddress>
          <ipV4Address>192.0.2.1</ipV4Address>
        </otherPartyIpAddress>
        <otherPartyPortNumber>
          <port>10</port>

```

```

    </otherPartyPortNumber>
    <userIpAddress>
      <ipV4Address>192.0.2.10</ipV4Address>
    </userIpAddress>
    <userPortNumber>
      <port>100</port>
    </userPortNumber>
  </flowDescription>
</ipFlow>
<flowStatus>EnabledDownlink</flowStatus>
</media>
<duration>7200</duration>
</qos:qosFeatureData>

```

### 6.2.5.1.2 Response

HTTP/1.1 201 Created  
 Content-Type: application/xml  
 Location: http://example.com/exampleAPI/qos/v1/tel%3A%2B19585550100/appliedQosFeatures/hdv1080  
 Content-Length: nnnn  
 Date: Mon, 28 Jul 2011 17:51:59 GMT

```

<?xml version="1.0" encoding="UTF-8"?>
<qos:qosFeatureData xmlns:qos="urn:oma:xml:rest:netapi:qos:1">
  <clientCorrelator>v1234</clientCorrelator>
  <predefinedQosFeatureId>hdv1080</predefinedQosFeatureId>
  <media>
    <mediaNumber>1</mediaNumber>
    <ipFlow>
      <flowNumber>1</flowNumber>
      <flowDescription>
        <direction>Downlink</direction>
        <protocol>TCP</protocol>
        <otherPartyIpAddress>
          <ipV4Address>192.0.2.1</ipV4Address>
        </otherPartyIpAddress>
        <otherPartyPortNumber>
          <port>10</port>
        </otherPartyPortNumber>
        <userIpAddress>
          <ipV4Address>192.0.2.10</ipV4Address>
        </userIpAddress>
        <userPortNumber>
          <port>100</port>
        </userPortNumber>
      </flowDescription>
    </ipFlow>
    <flowStatus>EnabledDownlink</flowStatus>
  </media>
  <duration>7200</duration>
  <resourceURL>http://example.com/exampleAPI/qos/v1/tel%3A%2B19585550100/appliedQosFeatures/hdv1080</resourceURL>
</qos:qosFeatureData>

```

### 6.2.5.2 Example 2: Request to apply a predefined QoS feature on the end user connection, response with location of created resource (Informative)

This is the same as Example 1, except that the server returns only a reference to the created resource, rather than a copy of it (defined in [REST\_NetAPI\_Common] as an alternative way of resource creation responses).

#### 6.2.5.2.1 Request

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

```
<?xml version="1.0" encoding="UTF-8"?>
<qos:qosFeatureData xmlns:qos="urn:oma:xml:rest:netapi:qos:1">
  <clientCorrelator>v1234</clientCorrelator>
  <predefinedQosFeatureId>hdv1080</predefinedQosFeatureId>
  <media>
    <mediaNumber>1</mediaNumber>
    <ipFlow>
      <flowNumber>1</flowNumber>
      <flowDescription>
        <direction>Downlink</direction>
        <protocol>TCP</protocol>
        <otherPartyIpAddress>
          <ipV4Address>192.0.2.1</ipV4Address>
        </otherPartyIpAddress>
        <otherPartyPortNumber>
          <port>10</port>
        </otherPartyPortNumber>
        <userIpAddress>
          <ipV4Address>192.0.2.10</ipV4Address>
        </userIpAddress>
        <userPortNumber>
          <port>100</port>
        </userPortNumber>
      </flowDescription>
    </ipFlow>
    <flowStatus>EnabledDownlink</flowStatus>
  </media>
  <duration>7200</duration>
</qos:qosFeatureData>
```

#### 6.2.5.2.2 Response

```
HTTP/1.1 201 Created
Content-Type: application/xml
Location: http://example.com/exampleAPI/qos/v1/tel%3A%2B19585550100/appliedQosFeatures/hdv1080
Content-Length: nnnn
Date: Mon, 28 Jul 2011 17:51:59 GMT
```

```
<?xml version="1.0" encoding="UTF-8"?>
<common:resourceReference xmlns:common="urn:oma:xml:rest:netapi:common:1">
  <resourceURL>http://example.com/exampleAPI/qos/v1/tel%3A%2B19585550100/appliedQosFeatures/hdv1080</resourceURL>
</common:resourceReference>
```

### 6.2.5.3 Example 3: Request to apply a custom QoS feature on the end user connection (Informative)

In the following example for bidirectional communication, rather than using a predefined QoS feature in the request, the application specifies own values for QoS properties such as bandwidth, media type, and reservation priority. There are two different media types included in the request.

#### 6.2.5.3.1 Request

```
POST /exampleAPI/qos/v1/tel%3A%2B19585550100/appliedQosFeatures HTTP/1.1
```

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

```
Content-Length: nnn
```

```
Accept: application/xml
```

```
Host: example.com
```

```
<?xml version="1.0" encoding="UTF-8"?>
<qos:qosFeatureData xmlns:qos="urn:oma:xml:rest:netapi:qos:1">
  <clientCorrelator>vcnf123</clientCorrelator>
  <media>
    <mediaNumber>1</mediaNumber>
    <mediaType>Video</mediaType>
    <bandwidth>
      <minDownlinkBitRate>7000000</minDownlinkBitRate>
      <maxDownlinkBitRate>7000000</maxDownlinkBitRate>
    </bandwidth>
    <ipFlow>
      <flowNumber>1</flowNumber>
      <flowDescription>
        <direction>Downlink</direction>
        <protocol>TCP</protocol>
        <otherPartyIpAddress>
          <ipV4Address>192.0.2.2</ipV4Address>
        </otherPartyIpAddress>
        <otherPartyPortNumber>
          <port>12</port>
        </otherPartyPortNumber>
        <userIpAddress>
          <ipV4Address>192.0.2.12</ipV4Address>
        </userIpAddress>
        <userPortNumber>
          <port>102</port>
        </userPortNumber>
      </flowDescription>
      <flowDescription>
        <direction>Uplink</direction>
        <protocol>TCP</protocol>
        <otherPartyIpAddress>
          <ipV4Address>192.0.2.2</ipV4Address>
        </otherPartyIpAddress>
        <otherPartyPortNumber>
          <port>13</port>
        </otherPartyPortNumber>
        <userIpAddress>
          <ipV4Address>192.0.2.12</ipV4Address>
        </userIpAddress>
        <userPortNumber>
          <port>103</port>
        </userPortNumber>
      </flowDescription>
    </ipFlow>
  </media>
</qos:qosFeatureData>
```



```

        </userPortNumber>
    </flowDescription>
</ipFlow>
</media>
<media>
    <mediaNumber>2</mediaNumber>
    <mediaType>Audio</mediaType>
    <bandwidth>
        <minDownlinkBitRate>48000</minDownlinkBitRate>
        <maxDownlinkBitRate>48000</maxDownlinkBitRate>
    </bandwidth>
    <ipFlow>
        <flowNumber>1</flowNumber>
        <flowDescription>
            <direction>Downlink</direction>
            <protocol>TCP</protocol>
            <otherPartyIpAddress>
                <ipV4Address>192.0.2.2</ipV4Address>
            </otherPartyIpAddress>
            <otherPartyPortNumber>
                <port>14</port>
            </otherPartyPortNumber>
            <userIpAddress>
                <ipV4Address>192.0.2.12</ipV4Address>
            </userIpAddress>
            <userPortNumber>
                <port>104</port>
            </userPortNumber>
        </flowDescription>
        <flowDescription>
            <direction>Uplink</direction>
            <protocol>TCP</protocol>
            <otherPartyIpAddress>
                <ipV4Address>192.0.2.2</ipV4Address>
            </otherPartyIpAddress>
            <otherPartyPortNumber>
                <port>15</port>
            </otherPartyPortNumber>
            <userIpAddress>
                <ipV4Address>192.0.2.12</ipV4Address>
            </userIpAddress>
            <userPortNumber>
                <port>105</port>
            </userPortNumber>
        </flowDescription>
    </ipFlow>
</media>
<reservationPriority>Medium</reservationPriority>
<duration>5400</duration>
</qos:qosFeatureData>

```

### 6.2.5.3.2 Response

```

HTTP/1.1 201 Created
Content-Type: application/xml
Location: http://example.com/exampleAPI/qos/v1/tel%3A%2B19585550100/appliedQosFeatures/app100

```

Content-Length: nnnn

Date: Mon, 28 Jul 2011 17:51:59 GMT

```
<?xml version="1.0" encoding="UTF-8"?>
<qos:qosFeatureData xmlns:qos="urn:oma:xml:rest:netapi:qos:1">
  <clientCorrelator>vcnf123</clientCorrelator>
  <media>
    <mediaNumber>1</mediaNumber>
    <mediaType>Video</mediaType>
    <bandwidth>
      <minDownlinkBitRate>7000000</minDownlinkBitRate>
      <maxDownlinkBitRate>7000000</maxDownlinkBitRate>
    </bandwidth>
    <ipFlow>
      <flowNumber>1</flowNumber>
      <flowDescription>
        <direction>Downlink</direction>
        <protocol>TCP</protocol>
        <otherPartyIpAddress>
          <ipV4Address>192.0.2.2</ipV4Address>
        </otherPartyIpAddress>
        <otherPartyPortNumber>
          <port>12</port>
        </otherPartyPortNumber>
        <userIpAddress>
          <ipV4Address>192.0.2.12</ipV4Address>
        </userIpAddress>
        <userPortNumber>
          <port>102</port>
        </userPortNumber>
      </flowDescription>
      <flowDescription>
        <direction>Uplink</direction>
        <protocol>TCP</protocol>
        <otherPartyIpAddress>
          <ipV4Address>192.0.2.2</ipV4Address>
        </otherPartyIpAddress>
        <otherPartyPortNumber>
          <port>13</port>
        </otherPartyPortNumber>
        <userIpAddress>
          <ipV4Address>192.0.2.12</ipV4Address>
        </userIpAddress>
        <userPortNumber>
          <port>103</port>
        </userPortNumber>
      </flowDescription>
    </ipFlow>
  </media>
  <media>
    <mediaNumber>2</mediaNumber>
    <mediaType>Audio</mediaType>
    <bandwidth>
      <minDownlinkBitRate>48000</minDownlinkBitRate>
      <maxDownlinkBitRate>48000</maxDownlinkBitRate>
    </bandwidth>
  </media>
</qos:qosFeatureData>
```

```

<ipFlow>
  <flowNumber>1</flowNumber>
  <flowDescription>
    <direction>Downlink</direction>
    <protocol>TCP</protocol>
    <otherPartyIpAddress>
      <ipV4Address>192.0.2.2</ipV4Address>
    </otherPartyIpAddress>
    <otherPartyPortNumber>
      <port>14</port>
    </otherPartyPortNumber>
    <userIpAddress>
      <ipV4Address>192.0.2.12</ipV4Address>
    </userIpAddress>
    <userPortNumber>
      <port>104</port>
    </userPortNumber>
  </flowDescription>
  <flowDescription>
    <direction>Uplink</direction>
    <protocol>TCP</protocol>
    <otherPartyIpAddress>
      <ipV4Address>192.0.2.2</ipV4Address>
    </otherPartyIpAddress>
    <otherPartyPortNumber>
      <port>15</port>
    </otherPartyPortNumber>
    <userIpAddress>
      <ipV4Address>192.0.2.12</ipV4Address>
    </userIpAddress>
    <userPortNumber>
      <port>105</port>
    </userPortNumber>
  </flowDescription>
</ipFlow>
</media>
<reservationPriority>Medium</reservationPriority>
<duration>5400</duration>
<resourceURL>http://example.com/exampleAPI/qos/v1/tel%3A%2B19585550100/appliedQosFeatures/app100</resourceURL>
</qos:qosFeatureData>

```

#### 6.2.5.4 Example 4: Application service provider sponsoring user's data usage (Informative)

The following is an example for sponsored data connectivity where a sponsor (e.g. Application service provider) pays for user's data usage in order to allow the user to access the Application service provider's services. The identity of the sponsor is included in the request.

##### 6.2.5.4.1 Request

```

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

```

```

<?xml version="1.0" encoding="UTF-8"?>
<qos:qosFeatureData xmlns:qos="urn:oma:xml:rest:netapi:qos:1">
  <clientCorrelator>v1234</clientCorrelator>
  <media>
    <mediaNumber>1</mediaNumber>
    <mediaType>Video</mediaType>
    <bandwidth>
      <minDownlinkBitRate>7000000</minDownlinkBitRate>
      <maxDownlinkBitRate>7000000</maxDownlinkBitRate>
    </bandwidth>
    <ipFlow>
      <flowNumber>1</flowNumber>
      <flowDescription>
        <direction>Downlink</direction>
        <protocol>TCP</protocol>
        <otherPartyIpAddress>
          <ipV4Address>192.0.2.2</ipV4Address>
        </otherPartyIpAddress>
        <otherPartyPortNumber>
          <port>12</port>
        </otherPartyPortNumber>
        <userIpAddress>
          <ipV4Address>192.0.2.12</ipV4Address>
        </userIpAddress>
        <userPortNumber>
          <port>102</port>
        </userPortNumber>
      </flowDescription>
    </ipFlow>
    <flowStatus>EnabledDownlink</flowStatus>
  </media>
  <reservationPriority>High</reservationPriority>
  <volume>36000000</volume>
  <sponsorId>sp100</sponsorId>
</qos:qosFeatureData>

```

#### 6.2.5.4.2 Response

HTTP/1.1 201 Created  
 Content-Type: application/xml  
 Location: http://example.com/exampleAPI/qos/v1/tel%3A%2B19585550100/appliedQosFeatures/app200  
 Content-Length: nnnn  
 Date: Mon, 28 Jul 2011 17:51:59 GMT

```

<?xml version="1.0" encoding="UTF-8"?>
<qos:qosFeatureData xmlns:qos="urn:oma:xml:rest:netapi:qos:1">
  <clientCorrelator>v1234</clientCorrelator>
  <media>
    <mediaNumber>1</mediaNumber>
    <mediaType>Video</mediaType>
    <bandwidth>
      <minDownlinkBitRate>7000000</minDownlinkBitRate>
      <maxDownlinkBitRate>7000000</maxDownlinkBitRate>
    </bandwidth>
    <ipFlow>

```

```

    <flowNumber>1</flowNumber>
    <flowDescription>
      <direction>Downlink</direction>
      <protocol>TCP</protocol>
      <otherPartyIpAddress>
        <ipV4Address>192.0.2.2</ipV4Address>
      </otherPartyIpAddress>
      <otherPartyPortNumber>
        <port>12</port>
      </otherPartyPortNumber>
      <userIpAddress>
        <ipV4Address>192.0.2.12</ipV4Address>
      </userIpAddress>
      <userPortNumber>
        <port>102</port>
      </userPortNumber>
    </flowDescription>
  </ipFlow>
  <flowStatus>EnabledDownlink</flowStatus>
</media>
<reservationPriority>High</reservationPriority>
<volume>36000000</volume>
<sponsorId>sp100</sponsorId>
<resourceURL>http://example.com/exampleAPI/qos/v1/tel%3A%2B19585550100/appliedQosFeatures/app200</resourceURL>
</qos:qosFeatureData>

```

### 6.2.5.5 Example 4: Request to apply a custom QoS feature on the end user connection, which is not supported by the server (Informative)

In the following example, an application is trying to apply a custom QoS feature on an end user connection however it fails since the server does not support custom QoS features.

#### 6.2.5.5.1 Request

```

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

```

```

<?xml version="1.0" encoding="UTF-8"?>
<qos:qosFeatureData xmlns:qos="urn:oma:xml:rest:netapi:qos:1">
  <clientCorrelator>v1234</clientCorrelator>
  <media>
    <mediaNumber>1</mediaNumber>
    <mediaType>Video</mediaType>
    <bandwidth>
      <minDownlinkBitRate>7000000</minDownlinkBitRate>
      <maxDownlinkBitRate>7000000</maxDownlinkBitRate>
    </bandwidth>
  </media>
  <ipFlow>
    <flowNumber>1</flowNumber>
    <flowDescription>
      <direction>Downlink</direction>
      <protocol>TCP</protocol>
      <otherPartyIpAddress>

```

```

    <ipV4Address>192.0.2.2</ipV4Address>
  </otherPartyIpAddress>
  <otherPartyPortNumber>
    <port>12</port>
  </otherPartyPortNumber>
  <userIpAddress>
    <ipV4Address>192.0.2.12</ipV4Address>
  </userIpAddress>
  <userPortNumber>
    <port>102</port>
  </userPortNumber>
</flowDescription>
</ipFlow>
<flowStatus>EnabledDownlink</flowStatus>
</media>
<reservationPriority>Medium</reservationPriority>
<duration>5400</duration>
</qos:qosFeatureData>

```

### 6.2.5.5.2 Response

```

HTTP/1.1 403 Forbidden
Content-Type: application/xml
Content-Length: nnnn
Date: Mon, 28 Jul 2011 17:51:59 GMT

```

```

<?xml version="1.0" encoding="UTF-8"?>
<common:requestError xmlns:common="urn:oma:xml:rest:netapi:common:1">
  <policyException>
    <messageId>POL1032</messageId>
    <text>Custom QoS features are not supported</text>
  </policyException>
</common:requestError>

```

## 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 [RFC2616].

## 6.3 Resource: Individual QoS feature applied on an end user connection

The resource used is:

**http://{serverRoot}/qos/{apiVersion}/{userId}/appliedQosFeatures/{featureId}**

This resource is used by a client for managing (retrieve, update, delete) an individual QoS features applied on an end user connection.

### 6.3.1 Request URL variables

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

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
userId	Identifier of the user on whose behalf the application acts Examples: tel:+19585550100, acr:pseudonym123
featureId	Identifier of QoS feature applied on an end user connection

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 Quality of Service API, see section 7.

## 6.3.3 GET

This operation is used for retrieving an individual QoS feature applied on an end user connection.

### 6.3.3.1 Example: Retrieving an individual QoS feature applied on the end user connection (Informative)

#### 6.3.3.1.1 Request

```
GET /exampleAPI/qos/v1/tel%3A%2B19585550100/appliedQosFeatures/hdv1080 HTTP/1.1
Accept: application/xml
Host: example.com
```

#### 6.3.3.1.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: nnnn
Date: Mon, 28 Jul 2011 17:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<qos:qosFeatureData xmlns:qos="urn:oma:xml:rest:netapi:qos:1">
  <clientCorrelator>v1234</clientCorrelator>
  <predefinedQosFeatureId>hdv1080</predefinedQosFeatureId>
  <media>
    <mediaNumber>1</mediaNumber>
    <ipFlow>
      <flowNumber>1</flowNumber>
      <flowDescription>
        <direction>Downlink</direction>
        <protocol>TCP</protocol>
        <otherPartyIpAddress>
          <ipV4Address>192.0.2.1</ipV4Address>
        </otherPartyIpAddress>
        <otherPartyPortNumber>
          <port>10</port>
        </otherPartyPortNumber>
        <userIpAddress>
          <ipV4Address>192.0.2.10</ipV4Address>
        </userIpAddress>
      </flowDescription>
    </ipFlow>
  </media>
</qos:qosFeatureData>
```

```

    </userIpAddress>
    <userPortNumber>
      <port>100</port>
    </userPortNumber>
  </flowDescription>
</ipFlow>
<flowStatus>EnabledDownlink</flowStatus>
</media>
<duration>4200</duration>
<resourceURL>http://example.com/exampleAPI/qos/v1/tel%3A%2B19585550100/appliedQosFeatures/hdv1080</resourceURL>
</qos:qosFeatureData>

```

## 6.3.4 PUT

The operation is used for retrieving or updating attributes for a QoS feature applied on an end user connection.

### 6.3.4.1 Example: Updating attributes for QoS feature applied on the end user connection (Informative).

The application requests changing of duration time for a QoS feature applied on an end user connection.

#### 6.3.4.1.1 Request

```
PUT /exampleAPI/qos/v1/tel%3A%2B19585550100/appliedQosFeatures/hdv1080 HTTP/1.1
```

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

```
Content-Length: nnnn
```

```
Accept: application/xml
```

```
Host: example.com
```

```

<?xml version="1.0" encoding="UTF-8"?>
<qos:qosFeatureData xmlns:qos="urn:oma:xml:rest:netapi:qos:1">
  <clientCorrelator>v1234</clientCorrelator>
  <predefinedQosFeatureId>hdv1080</predefinedQosFeatureId>
  <media>
    <mediaNumber>1</mediaNumber>
    <ipFlow>
      <flowNumber>1</flowNumber>
      <flowDescription>
        <direction>Downlink</direction>
        <protocol>TCP</protocol>
        <otherPartyIpAddress>
          <ipV4Address>192.0.2.1</ipV4Address>
        </otherPartyIpAddress>
        <otherPartyPortNumber>
          <port>10</port>
        </otherPartyPortNumber>
        <userIpAddress>
          <ipV4Address>192.0.2.10</ipV4Address>
        </userIpAddress>
        <userPortNumber>
          <port>100</port>
        </userPortNumber>
      </flowDescription>
    </ipFlow>
    <flowStatus>EnabledDownlink</flowStatus>
  </media>

```



```

<duration>7200</duration>
<resourceURL>http://example.com/exampleAPI/qos/v1/tel%3A%2B19585550100/appliedQosFeatures/hdv1080</resourceURL>
</qos:qosFeatureData>

```

### 6.3.4.1.2 Response

```

HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: nnnn
Date: Mon, 28 Jul 2011 17:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<qos:qosFeatureData xmlns:qos="urn:oma:xml:rest:netapi:qos:1">
  <clientCorrelator>v1234</clientCorrelator>
  <predefinedQosFeatureId>hdv1080</predefinedQosFeatureId>
  <media>
    <mediaNumber>1</mediaNumber>
    <ipFlow>
      <flowNumber>1</flowNumber>
      <flowDescription>
        <direction>Downlink</direction>
        <protocol>TCP</protocol>
        <otherPartyIpAddress>
          <ipV4Address>192.0.2.1</ipV4Address>
        </otherPartyIpAddress>
        <otherPartyPortNumber>
          <port>10</port>
        </otherPartyPortNumber>
        <userIpAddress>
          <ipV4Address>192.0.2.10</ipV4Address>
        </userIpAddress>
        <userPortNumber>
          <port>100</port>
        </userPortNumber>
      </flowDescription>
    </ipFlow>
    <flowStatus>EnabledDownlink</flowStatus>
  </media>
  <duration>7200</duration>
  <resourceURL>http://example.com/exampleAPI/qos/v1/tel%3A%2B19585550100/appliedQosFeatures/hdv1080</resourceURL>
</qos:qosFeatureData>

```

### 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, PUT, DELETE' field in the response as per section 14.7 of [RFC2616].

### 6.3.6 DELETE

This operation is used by an application to remove (terminate) an individual QoS feature applied on an end user connection.

### 6.3.6.1 Example: Removing QoS feature which is applied to the end user connection (Informative)

#### 6.3.6.1.1 Request

```
DELETE /exampleAPI/qos/v1/tel%3A%2B19585550100/appliedQosFeatures/hdv1080 HTTP/1.1
Accept: application/xml
Host: example.com
```

#### 6.3.6.1.2 Response

```
HTTP/1.1 204 No Content
Date: Fri, 28 Jun 2013 17:51:59 GMT
```

## 6.4 Resource: Individual attribute for QoS feature applied on an end user connection

The resource used is:

**http://{serverRoot}/qos/{apiVersion}/{userId}/appliedQosFeatures/{featureId}/[ResourceRelPath]**

This resource is used by a client for managing (retrieve and update) individual parameter(s) for a QoS feature applied on an end user connection.

### 6.4.1 Request URL variables

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

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
userId	Identifier of the user on whose behalf the application acts Examples: tel:+19585550100, acr:pseudonym123
featureId	Identifier of QoS feature applied on an end user connection
[ResourceRelPath]	Relative resource path for a Light-weight Resource, consisting of a relative path down to an element in the data structure. For more information about the applicable values (strings) for this variable, see 6.4.1.1.

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

#### 6.4.1.1 Light-weight relative resource paths

The following table describes the type of Light-weight Resources that can be accessed by using this resource, applicable methods, and the link to a data structure that contains values (strings) for those relative resource paths.

Light-weight Resource type	Method supported	Description
Duration time or Data Volume for QoS feature	GET, PUT	Enables access to duration (lifetime), and data volume parameters for a particular QoS feature applied on an end user connection. See column [ResourceRelPath] for elements "duration" and "volume" in section 5.2.2.4 for possible values for the

		light-weight relative resource path.
Bandwidth Information	GET, PUT	Enables access to bandwidth information for a particular QoS feature applied on an end user connection. See column [ResourceRelPath] for element “bandwidth” in section 5.2.2.5, for possible value of the light-weight relative resource path.
IP flow status	GET, PUT	Enables access to IP flow(s) status parameter on a media number level or IP flow number level for a particular QoS feature applied on an end user connection. See column [ResourceRelPath] for elements “flowStatus” in section 5.2.2.5, and “flowStatus” in section 5.2.2.7 for possible values of the light-weight relative resource path.

## 6.4.2 Response Codes and Error Handling

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

For Policy Exception and Service Exception fault codes applicable to Quality of Service API, see section 7.

### 6.4.3 GET

This operation is used for retrieving duration attribute for QoS feature applied on an end user connection by using Light-weight Resources.

#### 6.4.3.1 Example 1: Retrieving duration time for QoS feature applied on the end user connection (Informative)

##### 6.4.3.1.1 Request

```
GET /exampleAPI/qos/v1/tel%3A%2B19585550100/appliedQosFeatures/hdv1080/duration HTTP/1.1
Accept: application/xml
Host: example.com
```

##### 6.4.3.1.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: nnnn
Date: Mon, 28 Jul 2011 17:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<qos:duration xmlns:qos="urn:oma:xml:rest:netapi:qos:1">7200</qos:duration>
```

#### 6.4.3.2 Example 2: Retrieving IP flow status on media number level for QoS feature applied on the end user connection (Informative)

##### 6.4.3.2.1 Request

```
GET /exampleAPI/qos/v1/tel%3A%2B19585550100/appliedQosFeatures/hdv1080/media/1/flowStatus HTTP/1.1
Accept: application/xml
Host: example.com
```

### 6.4.3.2.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: nnnn
Date: Mon, 28 Jul 2011 17:51:59 GMT
```

```
<?xml version="1.0" encoding="UTF-8"?>
<qos:flowStatus xmlns:qos="urn:oma:xml:rest:netapi:qos:1">EnabledDownlink</qos:flowStatus>
```

### 6.4.3.3 Example 3: Retrieving IP flow status on IP flow number level for QoS feature applied on the end user connection (Informative)

#### 6.4.3.3.1 Request

```
GET /exampleAPI/qos/v1/tel%3A%2B19585550100/appliedQosFeatures/hdv1080/media/1/flow/1/flowStatus HTTP/1.1
Accept: application/xml
Host: example.com
```

#### 6.4.3.3.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: nnnn
Date: Mon, 28 Jul 2011 17:51:59 GMT
```

```
<?xml version="1.0" encoding="UTF-8"?>
<qos:flowStatus xmlns:qos="urn:oma:xml:rest:netapi:qos:1">EnabledDownlink</qos:flowStatus>
```

## 6.4.4 PUT

The operation is used for updating duration time for a QoS feature applied on an end user connection.

### 6.4.4.1 Example 1: Updating duration time for QoS feature applied on the end user connection (Informative)

#### 6.4.4.1.1 Request

```
PUT /exampleAPI/qos/v1/tel%3A%2B19585550100/appliedQosFeatures/hdv1080/duration HTTP/1.1
Content-Type: application/xml
Content-Length: nnnn
Accept: application/xml
Host: example.com
```

```
<?xml version="1.0" encoding="UTF-8"?>
<qos:duration xmlns:qos="urn:oma:xml:rest:netapi:qos:1">9000</qos:duration>
```

#### 6.4.4.1.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: nnnn
Date: Mon, 28 Jul 2011 17:51:59 GMT
```

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

```
<qos:duration xmlns:qos="urn:oma:xml:rest:netapi:qos:1">9000</qos:duration>
```

#### 6.4.4.2 Example 2: Updating IP flow status on IP flow number level for QoS feature (Informative)

##### 6.4.4.2.1 Request

```
PUT /exampleAPI/qos/v1/tel%3A%2B19585550100/appliedQosFeatures/hdv1080/media/1/flow/1/flowStatus HTTP/1.1
```

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

```
Content-Length: nnnn
```

```
Accept: application/xml
```

```
Host: example.com
```

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

```
<qos:flowStatus xmlns:qos="urn:oma:xml:rest:netapi:qos:1">Disabled</qos:flowStatus>
```

##### 6.4.4.2.2 Response

```
HTTP/1.1 200 OK
```

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

```
Content-Length: nnnn
```

```
Date: Mon, 28 Jul 2011 17:51:59 GMT
```

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

```
<qos:flowStatus xmlns:qos="urn:oma:xml:rest:netapi:qos:1">Disabled</qos:flowStatus>
```

## 6.4.5 POST

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET, PUT' field in the response as per section 14.7 of [RFC2616].

## 6.4.6 DELETE

Method not allowed by the resource. The returned HTTP error status is 405. The server should also include the 'Allow: GET, PUT' field in the response as per section 14.7 of [RFC2616].

## 6.5 Resource: All subscriptions to QoS notifications

The resource used is:

**http://{serverRoot}/qos/{apiVersion}/{userId}/subscriptions**

This resource is used for retrieving all subscriptions to notifications related to predefined and applied QoS features.

### 6.5.1 Request URL variables

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

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
userId	Identifier of the user on whose behalf the application acts

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 Quality of Service API, see section 7.

## 6.5.3 GET

This operation is used for retrieving all subscriptions to notifications related to predefined and applied QoS features.

### 6.5.3.1 Example: Retrieving all subscriptions to notifications related to QoS features (Informative)

#### 6.5.3.1.1 Request

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

#### 6.5.3.1.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: nnnn
Date: Fri, 28 Jun 2013 17:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<qos:qosFeaturesSubscriptionList xmlns:qos="urn:oma:xml:rest:netapi:qos:1">
  <predefinedQosFeaturesSubscriptionList>
    <predefinedQosFeaturesSubscription>
      <callbackReference>
        <notifyURL>http://application.example.com/qos/notifications/77777</notifyURL>
        <callbackData>abcd</callbackData>
      </callbackReference>
      <duration>7000</duration>
      <clientCorrelator>p1234</clientCorrelator>
    <predefinedQosFeatureId>audio16</predefinedQosFeatureId><resourceURL>http://example.com/exampleAPI/qos/v1/tel%3A%
2B19585550100/subscriptions/predefinedQosFeatures/sub001</resourceURL>
    </predefinedQosFeaturesSubscription>
  </predefinedQosFeaturesSubscriptionList>
  <resourceURL>http://example.com/exampleAPI/qos/v1/tel%3A%2B19585550100/subscriptions/predefinedQosFeatures</resourceURL>
  <appliedQosFeaturesSubscriptionList>
    <appliedQosFeaturesSubscription>
      <callbackReference>
        <notifyURL>http://application.example.com/qos/notifications/77777</notifyURL>
        <callbackData>efgh</callbackData>
      </callbackReference>
      <duration>6000</duration>
      <clientCorrelator>v1234</clientCorrelator>
      <eventType>AppliedQosFeatureReleased</eventType>
    </appliedQosFeaturesSubscription>
  </appliedQosFeaturesSubscriptionList>
  <resourceURL>http://example.com/exampleAPI/qos/v1/tel%3A%2B19585550100/subscriptions/appliedQosFeatures/sub002</resourceURL>
  </appliedQosFeaturesSubscription>
</qos:qosFeaturesSubscriptionList>
```

```
<resourceURL>http://example.com/exampleAPI/qos/v1/tel%3A%2B19585550100/subscriptions/appliedQosFeatures</resourceURL>
</appliedQosFeaturesSubscriptionList>
<resourceURL>http://example.com/exampleAPI/qos/v1/tel%3A%2B19585550100/subscriptions</resourceURL>
</qos:qosFeaturesSubscriptionList>
```

### 6.5.4 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 [RFC2616].

### 6.5.5 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 [RFC2616].

### 6.5.6 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 [RFC2616].

## 6.6 Resource: All subscriptions to notifications for predefined QoS features

The resource used is:

**http://{serverRoot}/qos/{apiVersion}/{userId}/subscriptions/predefinedQosFeatures**

This resource is used for retrieving all subscriptions to notifications about availability of predefined QoS features to an end user.

### 6.6.1 Request URL variables

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

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
userId	Identifier of the user on whose behalf the application acts 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 Quality of Service API, see section 7.

### 6.6.3 GET

This operation is used for retrieving all subscriptions to notifications related to availability of predefined QoS features to an end user.

### 6.6.3.1 Example: Retrieving all subscriptions to notifications on availability of predefined QoS features (Informative)

#### 6.6.3.1.1 Request

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

#### 6.6.3.1.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: nnnn
Date: Fri, 28 Jun 2013 17:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<qos:predefinedQosFeaturesSubscriptionList xmlns:qos="urn:oma:xml:rest:netapi:qos:1">
  <predefinedQosFeaturesSubscription>
    <callbackReference>
      <notifyURL>http://application.example.com/qos/notifications/77777</notifyURL>
      <callbackData>abcd</callbackData>
    </callbackReference>
    <duration>7000</duration>
    <clientCorrelator>p1234</clientCorrelator>
    <predefinedQosFeatureId>audio16</predefinedQosFeatureId>
  <resourceURL>http://example.com/exampleAPI/qos/v1/tel%3A%2B19585550100/subscriptions/predefinedQosFeatures/sub001</resourceURL>
</predefinedQosFeaturesSubscription>
<resourceURL>http://example.com/exampleAPI/qos/v1/tel%3A%2B19585550100/subscriptions/predefinedQosFeatures</resourceURL>
</qos:predefinedQosFeaturesSubscriptionList>
```

### 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, POST’ field in the response as per section 14.7 of [RFC2616].

### 6.6.5 POST

This operation is used to create a new subscription for notifications on availability of predefined QoS features to an end user. The application can subscribe to these notifications in cases where a requested predefined QoS feature is not available to the user due to some underlying conditions such as network coverage, location, time of the day etc. A Boolean parameter “multipleNotificationsRequested” can be used to signal to the server whether a notification shall be sent only after the first occurrence the feature becomes available or for every occurrence in case of multiple occurrences.

#### 6.6.5.1 Example: Creating a new subscription for notifications on availability of predefined QoS features (Informative)

In the following example, Boolean parameter “multipleNotificationsRequested” is not included which means that the application is requesting just a single notification.

##### 6.6.5.1.1 Request

```
POST /exampleAPI/qos/v1/tel%3A%2B19585550100/subscriptions/predefinedQosFeatures HTTP/1.1
Content-Type: application/xml
Content-Length: nnnn
Accept: application/xml
```



Host: example.com

```
<?xml version="1.0" encoding="UTF-8"?>
<qos:predefinedQoSFeaturesSubscription xmlns:qos="urn:oma:xml:rest:netapi:qos:1">
  <callbackReference>
    <notifyURL>http://application.example.com/qos/notifications/77777</notifyURL>
    <callbackData>abcd</callbackData>
  </callbackReference>
  <duration>7000</duration>
  <clientCorrelator>p1234</clientCorrelator>
  <predefinedQoSFeatureId>audio16</predefinedQoSFeatureId>
</qos:predefinedQoSFeaturesSubscription>
```

### 6.6.5.1.2 Response

HTTP/1.1 201 Created  
 Content-Type: application/xml  
 Location: http://example.com/exampleAPI/qos/v1/tel%3A%2B19585550100/subscriptions/predefinedQoSFeatures/sub001  
 Content-Length: nnnn  
 Date: Fri, 28 Jun 2013 17:51:59 GMT

```
<?xml version="1.0" encoding="UTF-8"?>
<qos:predefinedQoSFeaturesSubscription xmlns:qos="urn:oma:xml:rest:netapi:qos:1">
  <callbackReference>
    <notifyURL>http://application.example.com/qos/notifications/77777</notifyURL>
    <callbackData>abcd</callbackData>
  </callbackReference>
  <duration>7000</duration>
  <clientCorrelator>p1234</clientCorrelator>
  <predefinedQoSFeatureId>audio16</predefinedQoSFeatureId>
<resourceURL>http://example.com/exampleAPI/qos/v1/tel%3A%2B19585550100/subscriptions/predefinedQoSFeatures/sub001</resourceURL>
</qos:predefinedQoSFeaturesSubscription>
```

## 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, POST’ field in the response as per section 14.7 of [RFC2616].

## 6.7 Resource: Individual subscription to notifications for predefined QoS features

The resource used is:

**http://{serverRoot}/qos/{apiVersion}/{userId}/subscriptions/predefinedQoSFeatures/{subscriptionId}**

This resource is used to manage (retrieve, delete) an individual subscription to notifications on availability of predefined QoS features that are available to an end user.

### 6.7.1 Request URL variables

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

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
userId	Identifier of the user on whose behalf the application acts Examples: tel:+19585550100, acr:pseudonym123
subscriptionId	Identifier of the subscription generated by the server.

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 Quality of Service API, see section 7.

## 6.7.3 GET

This operation is used to retrieve an individual subscription for notifications on availability of predefined QoS features to an end user.

### 6.7.3.1 Example: Retrieving an individual subscriptions to notifications on availability of predefined QoS features (Informative)

#### 6.7.3.1.1 Request

```
GET /exampleAPI/qos/v1/tel%3A%2B19585550100/subscriptions/predefinedQosFeatures/sub001 HTTP/1.1
Accept: application/xml
Host: example.com
```

#### 6.7.3.1.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: nnnn
Date: Fri, 28 Jun 2013 17:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<qos:predefinedQosFeaturesSubscription xmlns:qos="urn:oma:xml:rest:netapi:qos:1">
  <callbackReference>
    <notifyURL>http://application.example.com/qos/notifications/77777</notifyURL>
    <callbackData>abcd</callbackData>
  </callbackReference>
  <duration>7000</duration>
  <clientCorrelator>p1234</clientCorrelator>
  <predefinedQosFeatureId>audio16</predefinedQosFeatureId>
  <resourceURL>http://example.com/exampleAPI/qos/v1/tel%3A%2B19585550100/subscriptions/predefinedQosFeatures/sub001</resourceURL>
</qos:predefinedQosFeaturesSubscription>
```

## 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, DELETE' field in the response as per section 14.7 of [RFC2616].

## 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, DELETE' field in the response as per section 14.7 of [RFC2616].

## 6.7.6 DELETE

### 6.7.6.1 Example: Cancelling a subscription to notifications on availability of predefined QoS features (Informative)

#### 6.7.6.1.1 Request

```
DELETE /exampleAPI/qos/v1/tel%3A%2B19585550100/subscriptions/predefinedQoSFeatures/sub001 HTTP/1.1
Accept: application/xml
Host: example.com
```

#### 6.7.6.1.2 Response

```
HTTP/1.1 204 No Content
Date: Fri, 28 Jun 2013 17:51:59 GMT
```

## 6.8 Resource: All subscriptions to notifications for QoS features applied on an end user connection

The resource used is:

**http://{serverRoot}/qos/{apiVersion}/{userId}/subscriptions/appliedQoSFeatures**

This resource is used for retrieving a list of subscriptions to notifications about events occurring for QoS features applied to an end user connection.

### 6.8.1 Request URL variables

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

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
userId	Identifier of the user on whose behalf the application acts Examples: tel:+19585550100, acr:pseudonym123

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

### 6.8.2 Response Codes and Error Handling

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

For Policy Exception and Service Exception fault codes applicable to Quality of Service API, see section 7.

### 6.8.3 GET

This operation is used for retrieving all subscriptions to notifications about events occurring for QoS features applied to an end user connection.

### 6.8.3.1 Example: Retrieving all subscriptions to notifications about events occurring for applied QoS features (Informative)

#### 6.8.3.1.1 Request

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

#### 6.8.3.1.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: nnnn
Date: Fri, 28 Jun 2013 17:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<qos:appliedQosFeaturesSubscriptionList xmlns:qos="urn:oma:xml:rest:netapi:qos:1">
  <appliedQosFeaturesSubscription>
    <callbackReference>
      <notifyURL>http://application.example.com/qos/notifications/77777</notifyURL>
      <callbackData>efgh</callbackData>
    </callbackReference>
    <duration>6000</duration>
    <clientCorrelator>v1234</clientCorrelator>
    <eventType>AppliedQosFeatureReleased</eventType>
  <resourceURL>http://example.com/exampleAPI/qos/v1/tel%3A%2B19585550100/subscriptions/appliedQosFeatures/sub002</resourceURL>
  </appliedQosFeaturesSubscription>
  <resourceURL>http://example.com/exampleAPI/qos/v1/tel%3A%2B19585550100/subscriptions/appliedQosFeatures</resourceURL>
</qos:appliedQosFeaturesSubscriptionList>
```

### 6.8.4 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 [RFC2616].

### 6.8.5 POST

This operation is used to create a new subscription for notifications about events occurring for a QoS features applied to an end user connection.

#### 6.8.5.1 Example: Creating a new subscription for notifications about events occurring for applied QoS features (Informative)

##### 6.8.5.1.1 Request

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

<?xml version="1.0" encoding="UTF-8"?>
<qos:appliedQosFeaturesSubscription xmlns:qos="urn:oma:xml:rest:netapi:qos:1">
  <callbackReference>
    <notifyURL>http://application.example.com/qos/notifications/77777</notifyURL>
```

```

    <callbackData>efgh</callbackData>
  </callbackReference>
  <duration>6000</duration>
  <clientCorrelator>v1234</clientCorrelator>
  <eventType>AppliedQoSFeatureReleased</eventType>
</qos:appliedQoSFeaturesSubscription>

```

### 6.8.5.1.2 Response

```

HTTP/1.1 201 Created
Content-Type: application/xml
Location: http://example.com/exampleAPI/qos/v1/tel%3A%2B19585550100/subscriptions/appliedQoSFeatures/sub002
Content-Length: nnnn
Date: Fri, 28 Jun 2013 17:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<qos:appliedQoSFeaturesSubscription xmlns:qos="urn:oma:xml:rest:netapi:qos:1">
  <callbackReference>
    <notifyURL>http://application.example.com/qos/notifications/77777</notifyURL>
    <callbackData>efgh</callbackData>
  </callbackReference>
  <duration>6000</duration>
  <clientCorrelator>v1234</clientCorrelator>
  <eventType>AppliedQoSFeatureReleased</eventType>
<resourceURL>http://example.com/exampleAPI/qos/v1/tel%3A%2B19585550100/subscriptions/appliedQoSFeatures/sub002</resourceURL>
</qos:appliedQoSFeaturesSubscription>

```

### 6.8.6 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 [RFC2616].

## 6.9 Resource: Individual subscription to notifications for QoS features applied on an end user connection

The resource used is:

**http://{serverRoot}/qos/{apiVersion}/{userId}/appliedQoSFeatures/{subscriptionId}**

This resource is used to manage (retrieve, delete) an individual subscription to notifications about events occurring for QoS features applied to an end user connection.

### 6.9.1 Request URL variables

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

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
userId	Identifier of the user on whose behalf the application acts Examples: tel:+19585550100, acr:pseudonym123

subscriptionId	Identifier of the subscription generated by the server.
----------------	---

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

## 6.9.2 Response Codes and Error Handling

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

For Policy Exception and Service Exception fault codes applicable to Quality of Service API, see section 7.

### 6.9.3 GET

This operation is used to retrieve an individual subscription for notifications about events occurring for QoS features applied to an end user connection.

#### 6.9.3.1 Example: Retrieving an individual subscription to notifications about events occurring for applied QoS features (Informative)

##### 6.9.3.1.1 Request

```
GET /exampleAPI/qos/v1/tel%3A%2B19585550100/subscriptions/appliedQosFeatures/sub002 HTTP/1.1
Accept: application/xml
Host: example.com
```

##### 6.9.3.1.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/xml
Content-Length: nnnn
Date: Fri, 28 Jun 2013 17:51:59 GMT

<?xml version="1.0" encoding="UTF-8"?>
<qos:appliedQosFeaturesSubscription xmlns:qos="urn:oma:xml:rest:netapi:qos:1">
  <callbackReference>
    <notifyURL>http://application.example.com/qos/notifications/77777</notifyURL>
    <callbackData>efgh</callbackData>
  </callbackReference>
  <duration>6000</duration>
  <clientCorrelator>v1234</clientCorrelator>
  <eventType>AppliedQosFeatureReleased</eventType>
  <resourceURL>http://example.com/exampleAPI/qos/v1/tel%3A%2B19585550100/subscriptions/appliedQosFeatures/sub002</resourceURL>
</qos:appliedQosFeaturesSubscription>
```

### 6.9.4 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 [RFC2616].

### 6.9.5 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 [RFC2616].

## 6.9.6 DELETE

### 6.9.6.1 Example: Cancelling a subscription to notifications about events occurring for applied QoS features (Informative)

#### 6.9.6.1.1 Request

```
DELETE /exampleAPI/qos/v1/tel%3A%2B19585550100/subscriptions/appliedQoSFeatures/sub002 HTTP/1.1
Accept: application/xml
Host: example.com
```

#### 6.9.6.1.2 Response

```
HTTP/1.1 204 No Content
Date: Fri, 28 Jun 2013 17:51:59 GMT
```

## 6.10 Resource: Client notification about availability of predefined QoS features

This resource is a callback URL provided by the client for notification about availability of predefined QoS features. The RESTful QoS 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 to obtain the notifications, the client needs to use the mechanisms described in [REST\_NetAPI\_NotificationChannel], instead of the mechanism described below in section 6.10.5.

### 6.10.1 Request URL variables

Client provided if any.

### 6.10.2 Response Codes and Error Handling

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

For Policy Exception and Service Exception fault codes applicable to Quality of Service, see section 7.

### 6.10.3 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 [RFC2616].

### 6.10.4 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 [RFC2616].

### 6.10.5 POST

This operation is used to notify the client that a particular predefined QoS feature is available to the end user.

### 6.10.5.1 Example: Notifying a client that a predefined QoS feature is available to the end user (Informative)

#### 6.10.5.1.1 Request

```
POST /qos/notifications/77777 HTTP/1.1
Content-Type: application/xml
Content-Length: nnnn
Accept: application/xml
Host: example.com

<?xml version="1.0" encoding="UTF-8"?>
<qos:predefinedQoSAvailabilityNotification xmlns:qos="urn:oma:xml:rest:netapi:qos:1">
  <callbackData>abcd</callbackData>
  <predefinedQoSFeatureId> audio16</predefinedQoSFeatureId>
  <link rel="PredefinedQoSFeaturesSubscription"
    href="http://example.com/exampleAPI/qos/v1/tel%3A%2B19585550100/subscriptions/predefinedQoSFeatures/sub001"/>
</qos:predefinedQoSAvailabilityNotification>
```

#### 6.10.5.1.2 Response

```
HTTP/1.1 204 No Content
Date: Fri, 28 Jun 2013 17:51:59 GMT
```

## 6.10.6 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 [RFC2616].

## 6.11 Resource: Client notification about an event occurred for QoS features applied on an end user connection

This resource is a callback URL provided by the client for notification about events occurring for QoS features applied on an end user connection. The RESTful QoS 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 to obtain the notifications, the client needs to use the mechanisms described in [REST\_NetAPI\_NotificationChannel], instead of the mechanism described below in section 6.11.5.

### 6.11.1 Request URL variables

Client provided if any.

### 6.11.2 Response Codes and Error Handling

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

For Policy Exception and Service Exception fault codes applicable to Quality of Service, see section 7.

### 6.11.3 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 [RFC2616].



## 6.11.4 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 [RFC2616].

## 6.11.5 POST

This operation is used to notify the client about an event occurred for QoS features applied to the end user.

### 6.11.5.1 Example: Notifying a client about an event occurring for applied QoS features (Informative)

In the following example, beside the event type the server has included also a reference to the subscription and a reference to the QoS feature for which the notification applies.

#### 6.11.5.1.1 Request

```
POST /qos/notifications/77777 HTTP/1.1
Content-Type: application/xml
Content-Length: nnnn
Accept: application/xml
Host: example.com

<?xml version="1.0" encoding="UTF-8"?>
<qos:appliedQosFeaturesNotification xmlns:qos="urn:oma:xml:rest:netapi:qos:1">
  <callbackData>efgh</callbackData>
  <eventType>AppliedQosFeatureReleased</eventType>
  <eventDescription>Duration time for the feature has elapsed</eventDescription>
  <link rel="AppliedQoSfeaturesSubscription"
    href="http://example.com/exampleAPI/qos/v1/tel%3A%2B19585550100/subscriptions/appliedQosFeatures/sub002"/>
  <link rel="QoSFeatureData"
    href="http://example.com/exampleAPI/qos/v1/tel%3A%2B19585550100/appliedQosFeatures/hdv1080"/>
</qos:appliedQosFeaturesNotification>
```

#### 6.11.5.1.2 Response

```
HTTP/1.1 204 No Content
Date: Fri, 28 Jun 2013 17:51:59 GMT
```

## 6.11.6 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 [RFC2616].

## 7. Fault definitions

### 7.1 Service Exceptions

For common Service Exceptions refer to [REST\_NetAPI\_Common]. The following additional Service Exception codes are defined for the RESTful Quality of Service API.

#### 7.1.1 SVC0340: Insufficient connection resources

Name	Description
MessageID	SVC0340
Text	Insufficient connection resources to fulfil the request
Variables	None
HTTP status code(s)	500 Internal Server Error

#### 7.1.2 SVC0341: Unknown QoS feature identifier

Name	Description
MessageID	SVC0341
Text	Unknown QoS feature identifier
Variables	None
HTTP status code(s)	400 Bad Request

#### 7.1.3 SVC0342: End user is not online

Name	Description
MessageID	SVC0342
Text	End user is not online
Variables	None
HTTP status code(s)	400 Bad Request

#### 7.1.4 SVC0343: Specified and temporary QoS feature conflict

Name	Description
MessageID	SVC0343
Text	Specified and temporary QoS feature conflict
Variables	None
HTTP status code(s)	400 Bad Request

### 7.1.5 SVC1010: Insufficient information to identify IP flow

Name	Description
MessageID	SVC1010
Text	Specified IP flow information insufficient to identify the IP flow
Variables	None
HTTP status code(s)	400 Bad Request

### 7.1.6 SVC1011: Specified IP flow does not exist

Name	Description
MessageID	SVC1011
Text	Specified IP flow does not exist
Variables	None
HTTP status code(s)	400 Bad Request

### 7.1.7 SVC1012: Duplicate media/flow number

Name	Description
MessageID	SVC1012
Text	Value %1 specified for %2 is a duplicate.
Variables	%1 – value for “mediaNumber” or “flowNumber” %2 – a string “mediaNumber” or “flowNumber”
HTTP status code(s)	400 Bad Request

## 7.2 Policy Exceptions

For common Policy Exceptions refer to [REST\_NetAPI\_Common]. The following additional Policy Exception codes are defined for the RESTful Quality of Service API.

### 7.2.1 POL1032: Custom QoS features not supported

Name	Description
MessageID	POL1032
Text	Custom QoS features are not supported
Variables	None
HTTP status code(s)	403 Forbidden

### 7.2.2 POL1033: Specifying volume limits not supported

Name	Description
------	-------------

MessageID	POL1033
Text	Specifying volume limits for QoS features is not supported
Variables	None
HTTP status code(s)	403 Forbidden

### 7.2.3 POL1034: Subscriptions to predefined QoS features not supported

Name	Description
MessageID	POL1034
Text	Subscriptions to notifications on availability of predefined QoS features are not supported
Variables	None
HTTP status code(s)	403 Forbidden

### 7.2.4 POL1035: Multiple notifications for predefined QoS features not supported

Name	Description
MessageID	POL1035
Text	Multiple notifications for predefined QoS features are not supported
Variables	None
HTTP status code(s)	403 Forbidden

### 7.2.5 POL1036: Sponsored QoS features not supported

Name	Description
MessageID	POL1036
Text	Sponsored QoS features are not supported.
Variables	None
HTTP status code(s)	403 Forbidden

## Appendix A. Change History

(Informative)

### A.1 Approved Version History

Reference	Date	Description
n/a	n/a	No prior version

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

Document Identifier	Date	Sections	Description
Draft Versions: REST_NetAPI_QoS-V1_0	31 Jan 2013	All	First baseline
	01 Oct 2013	1, 2, 3, 4, 5	Implemented CRs: OMA-ARC-REST-QoS-2013-0012R01-CR_TS_input_for_section_4 OMA-ARC-REST-QoS-2013-0013R01-CR_TS_section_5_resource_definitions OMA-ARC-REST-QoS-2013-0014R01-CR_TS_section_5_data_type_definitions OMA-ARC-REST-QoS-2013-0015R01-CR_TS_section_5_sequence_diagrams
	29 Nov 2013	4,5	Implemented CR OMA-ARC-REST-QoS-2013-0020R02-CR_TS_subscriptions_for_predefined_QoS_features
	18 Feb 2014	5.2	Implemented CRs OMA-ARC-REST-QoS-2014-0001-CR_QoS_TS_update_of_data_types OMA-ARC-REST-QoS-2014-0002-CR_QoS_TS_Editor_notes_resolution
	10 Mar 2014	4.1, 5, 5.1, 5.2, 5.3	Implemented CRs OMA-ARC-REST-QoS-2014-0003-CR_TS_fixing_resource_and_data_type_names OMA-ARC-REST-QoS-2014-0005R01-CR_TS_more_califications_for_sections_4_and_5 OMA-ARC-REST-QoS-2014-0006-CR_TS_subscription_to_predefinedQoS_sequence_flow
	24 Mar 2014	6	Implemented CR OMA-ARC-REST-QoS-2014-0008R01-CR_TS_input_for_section_6
	16 Apr 2014	2.1, 6, 7, Appendix B, C, D,E,G	Implemented CRs: OMA-ARC-REST-QoS-2014-0011-CR_TS_Appendix_B_with_SCR_tables OMA-ARC-REST-QoS-2014-0012-CR_TS_input_for_section_7 OMA-ARC-REST-QoS-2014-0013-CR_TS_input_for_Appendix_D_for_JSON_examples OMA-ARC-REST-QoS-2014-0014-CR_TS_input_for_Appendix_E_mapping_to_ParlayX OMA-ARC-REST-QoS-2014-0015-CR_TS_input_for_Appendix_G_authorization_aspects OMA-ARC-REST-QoS-2014-0016-CR_TS_input_for_Appendix_C OMA-ARC-REST-QoS-2014-0019-CR_TS_adding_mediaType_filter
	06 May 2014	3.2, 5, 5.1, 5.2	Implemented CR OMA-ARC-REST-QoS-2014-0020-CR_TS_adding_child_resources
	14 May 2014	5, 6, Appendix B, D	Implemented CRs: OMA-ARC-REST-QoS-2014-0022R04-CR_TS_further_info_for_child_resources OMA-ARC-REST-QoS-2014-0023-CR_TS_input_for_section_5_2_4 OMA-ARC-REST-QoS-2014-0024-CR_TS_small_fixes OMA-ARC-REST-QoS-2014-0025-CR_TS_JSON_files

Document Identifier	Date	Sections	Description
	18 Jun 2014	5, 6, Appendix B, D, G	Implemented CRs: OMA-ARC-REST-QoS-2014-0026- CR_TS_CONRR_editorial_comments_fixing OMA-ARC-REST-QoS-2014-0027- CR_TS_CONRR_technical_comments_resolution
	03 Sep 2014	Many	Implemented CRs: OMA-ARC-REST-QoS-2014-0029- CR_TS_subscription_lists_for_predefined_and_applied_qos_features OMA-ARC-REST-QoS-2014-0030- CR_TS_resolving_some_of_editor_notes OMA-ARC-REST-QoS-2014-0031R01- CR_TS_xml_examples_for_sponsored_data OMA-ARC-REST-QoS-2014-0036R01- CR_TS_CONRR_issue_B028_protocols OMA-ARC-REST-QoS-2014-0037- CR_TS_CONRR_issue_B040_scope_values
	08 Oct 2014	5.2.3.2	Implemented CR: OMA-ARC-REST-QoS-2014-0038- CR_TS_Editor_Note_for_FlowStatus
	15 Oct 2014	5.2	Implemented CR: OMA-ARC-REST-QoS-2014-0042R01- CR_TS_data_types_for_multiple_media_and_flows
	24 Oct 2014	Many	Implemented CR: OMA-ARC-REST-QoS-2014-0044- CR_TS_further_updates_for_multiple_media_and_flows
	05 Nov 2014	Many	Implemented CRs: OMA-ARC-REST-QoS-2014-0046R04- CR_TS_more_changes_for_data_types OMA-ARC-REST-QoS-2014-0048- CR_TS_small_changes_for_SCR
Candidate Version: REST_NetAPI_QoS-V1_0	25 Nov 2014	n/a	Status changed to Candidate by TP TP Ref # OMA-TP-2014-0266- INP_REST_NetAPI_QoS_V1_0_ERP_and_ETR_for_Candidate_Ap proval

## Appendix B. Static Conformance Requirements (Normative)

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

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

Item	Function	Reference	Requirement
REST-QOS-SUPPORT-S-001-M	Support for the RESTful for Quality of Service API	5, 6	
REST-QOS-SUPPORT-S-002-M	Support for the XML request & response format	6	
REST-QOS-SUPPORT-S-003-M	Support for the JSON request & response format	6	

#### B.1.1 SCR for REST.QoS.PredefinedQoSFeatures Server

Item	Function	Reference	Requirement
REST-QOS-PREDEFINED-S-001-M	Support for access to predefined QoS features	6.1	
REST-QOS-PREDEFINED-S-002-M	Retrieve predefined QoS features available to an user - GET	6.1.3	

#### B.1.2 SCR for REST.QoS.AppliedQoSFeatures Server

Item	Function	Reference	Requirement
REST-QOS-APPLIED-S-001-M	Support for handling of applied QoS features	6.2	
REST-QOS-APPLIED-S-002-M	Retrieve a list of QoS features applied on an end user connection - GET	6.2.3	
REST-QOS-APPLIED-S-003-M	Apply a QoS feature on an end user connection - POST	6.2.5	

#### B.1.3 SCR for REST.QoS.IndAppliedQoSFeature Server

Item	Function	Reference	Requirement
REST-QOS-INDAPPLIED-S-001-M	Support for access to an individual QoS feature applied on an end user connection	6.3	
REST-QOS-INDAPPLIED-S-002-M	Retrieve an individual QoS feature applied on an end user connection - GET	6.3.3	
REST-QOS-INDAPPLIED-S-003-M	Update an individual QoS feature applied on an end user connection - PUT	6.3.4	
REST-QOS-INDAPPLIED-S-004-M	Cancel/remove an individual QoS feature applied on an end user connection - DELETE	6.3.6	

#### B.1.4 SCR for REST.QoS.IndAppliedQoSFeature.Attribute Server

Item	Function	Reference	Requirement
REST-QOS-INDAPPLIED-ATTR-S-001-O	Support for handling of individual attributes for a	6.4	REST-QOS-INDAPPLIED-ATTR-S-003-O

Item	Function	Reference	Requirement
	QoS feature applied on an end user connection		
REST-QOS-INDAPPLIED-ATTR-S-002-O	Retrieve an individual attribute for a QoS feature applied on an end user connection - GET	6.4.3	
REST-QOS-INDAPPLIED-ATTR-S-003-O	Update an individual attribute for a QoS feature applied on an end user connection - PUT	6.4.4	

### B.1.5 SCR for REST.QoS.Subscriptions Server

Item	Function	Reference	Requirement
REST-QOS-SUBSCR-S-001-O	Support for subscriptions for QoS features	6.5	REST-QOS-SUBSCR-PREDEF-S-001-O OR REST-QOS-SUBSCR-APPL-S-001-M
REST-QOS-SUBSCR-S-002-O	Retrieve a list of active subscriptions for both, predefined and applied QoS features - GET	6.5.3	

### B.1.6 SCR for REST.QoS.Subscriptions.Predefined Server

Item	Function	Reference	Requirement
REST-QOS-SUBSCR-PREDEF-S-001-O	Support for subscriptions to predefined QoS features availability notifications	6.6	REST-QOS-SUBSCR-PREDEF-S-003-O
REST-QOS-SUBSCR-PREDEF-S-002-O	Retrieve a list of active subscriptions to predefined QoS features availability notifications - GET	6.6.3	
REST-QOS-SUBSCR-PREDEF-S-003-O	Create a new subscription to predefined QoS features availability notifications - POST	6.6.5	

### B.1.7 SCR for REST.QoS.IndSubscription.Predefined Server

Item	Function	Reference	Requirement
REST-QOS-INDSUBSCR-PREDEF-S-001-O	Support for access to an individual subscription to predefined QoS features availability notifications	6.7	REST-QOS-INDSUBSCR-PREDEF-S-003-O
REST-QOS-INDSUBSCR-PREDEF-S-002-O	Retrieve an individual subscriptions to predefined QoS features availability notifications - GET	6.7.3	
REST-QOS-INDSUBSCR-PREDEF-S-003-O	Cancel a subscription and stop corresponding notifications on availability of predefined QoS features - DELETE	6.7.6	



### B.1.8 SCR for REST.QoS.Subscriptions.Applied Server

Item	Function	Reference	Requirement
REST-QOS-SUBSCR-APPL-S-001-M	Support for subscriptions to applied QoS features event notifications	6.8	
REST-QOS-SUBSCR-APPL-S-002-O	Retrieve a list of active subscriptions to applied QoS features event notifications - GET	6.8.3	
REST-QOS-SUBSCR-APPL-S-003-M	Create a new subscription to applied QoS features event notifications - POST	6.8.5	

### B.1.9 SCR for REST.QoS.IndSubscription.Applied Server

Item	Function	Reference	Requirement
REST-QOS-INDSUBSCR-APPL-S-001-M	Support for access to an individual subscription to applied QoS features event notifications	6.9	
REST-QOS-INDSUBSCR-APPL-S-002-O	Retrieve an individual subscription to applied QoS features event notifications - GET	6.9.3	
REST-QOS-INDSUBSCR-APPL-S-003-M	Cancel a subscription and stop corresponding notifications to applied QoS features event notifications - DELETE	6.9.6	

### B.1.10 SCR for REST.QoS.Notifications.Predefined Server

Item	Function	Reference	Requirement
REST-QOS-NOTIF-PREDEF-S-001-O	Support for notifications on availability of predefined QoS features	6.10	REST-QOS-NOTIF-PREDEF-S-002-O
REST-QOS-NOTIF-PREDEF-S-002-O	Notification that predefined QoS feature(s) is available to a user - POST	6.10.5	

### B.1.11 SCR for REST.QoS.Notifications.Applied Server

Item	Function	Reference	Requirement
REST-QOS-NOTIF-APPL-S-001-M	Support for notifications on availability of applied QoS features	6.11	
REST-QOS-NOTIF-APPL-S-002-M	Notification about an event occurred for QoS feature(s) applied to an end user connection - POST	6.11.5	

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

This specification does not define any API request based on application/x-www-form-urlencoded MIME type.

## Appendix D. JSON examples (Informative)

JSON (JavaScript Object Notation) is a Light-weight, 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 REST invocations from browsers or other processors with JavaScript engines. Further information on JSON can be found at [RFC4627].

The following examples show the request and response for various operations using the JSON data format. 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 Retrieving a list of predefined QoS features generally available to the end user (section 6.1.3.1)

Request:

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

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnnn
Date: Mon, 28 Jul 2011 17:51:59 GMT

{"predefinedQosFeatureList": {
  "predefinedQosFeature": [
    {
      "link": {
        "href": "http://example.com/qos/predefinedQosFeatures/hdv1080",
        "rel": "PredefinedQosFeature"
      },
      "mediaInfo": {
        "bandwidth": {
          "minDownlinkBitRate": "7000000",
          "minUplinkBitRate": "7000000"
        },
        "mediaType": "Video"
      },
      "predefinedQosFeatureId": "hdv1080",
      "predefinedQosFeatureName": "VideoGold",
      "reservationPriority": "Medium"
    },
    {
      "link": {
        "href": "http://example.com/qos/predefinedQosFeatures/dvdv768",
        "rel": "PredefinedQosFeature"
      },
      "mediaInfo": {
        "bandwidth": {
          "minDownlinkBitRate": "4000000",
```

```

        "minUplinkBitRate": "4000000"
    },
    "mediaType": "Video"
  },
  "predefinedQosFeatureId": "dvdv768",
  "predefinedQosFeatureName": "VideoSilver",
  "reservationPriority": "Medium"
},
{
  "link": {
    "href": "http://example.com/qos/predefinedQosFeatures/audio16",
    "rel": "PredefinedQosFeature"
  },
  "mediaInfo": { "mediaType": "Audio"},
  "predefinedQosFeatureId": "audio16",
  "predefinedQosFeatureName": "AudioGold",
  "reservationPriority": "Medium"
},
{
  "link": {
    "href": "http://example.com/qos/predefinedQosFeatures/avg8768",
    "rel": "PredefinedQosFeature"
  },
  "mediaInfo": {
    "bandwidth": {
      "minDownlinkBitRate": "7000000",
      "minUplinkBitRate": "7000000"
    },
    "mediaType": "Video"
  },
  "predefinedQosFeatureId": "avg8768",
  "predefinedQosFeatureName": "GamingSilver",
  "reservationPriority": "Low"
}
],
"resourceURL": "http://example.com/exampleAPI/qos/v1/tel%3A%2B19585550100/predefinedQosFeatures"
}}

```

## D.2 Retrieving a list of predefined QoS features currently available to the end user (section 6.1.3.2)

Request:

```

GET /exampleAPI/qos/v1/tel%3A%2B19585550100/predefinedQosFeatures?currentlyAvailableOnly="True" HTTP/1.1
Accept: application/json
Host: example.com

```

Response:

```

HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnnn
Date: Mon, 28 Jul 2011 17:51:59 GMT

```

```

{"predefinedQosFeatureList": {
  "predefinedQosFeature": [
    {
      "link": {
        "href": " http://example.com/qos/predefinedQosFeatures/dvdv768",
        "rel": "PredefinedQosFeature"
      },
      "mediaInfo": {
        "bandwidth": {
          "minDownlinkBitRate": "4000000",
          "minUplinkBitRate": "4000000"
        },
        "mediaType": "Video"
      },
      "predefinedQosFeatureId": "dvdv768",
      "predefinedQosFeatureName": "VideoSilver",
      "reservationPriority": "Medium"
    },
    {
      "link": {
        "href": "http://example.com/qos/predefinedQosFeatures/audio16",
        "rel": "PredefinedQosFeature"
      },
      "mediaInfo": {"mediaType": "Audio"},
      "predefinedQosFeatureId": "audio16",
      "predefinedQosFeatureName": "AudioGold",
      "reservationPriority": "Medium"
    }
  ],
  "resourceURL": "http://example.com/exampleAPI/qos/v1/tel%3A%2B19585550100/predefinedQosFeatures"
}}

```

### D.3 Retrieving a list of QoS features applied on the end user connection (section 6.2.3.1)

Request:

```

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

```

Response:

```

HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnnn
Date: Mon, 28 Jul 2011 17:51:59 GMT

{"appliedQosFeatureList": {
  "qosFeature": [
    {
      "clientCorrelator": "v1234",
      "duration": "5400",
      "predefinedQosFeatureId": "hdv1080",

```

```

    "resourceURL": "http://example.com/exampleAPI/qos/v1/tel%3A%2B19585550100/appliedQosFeatures/hdv1080"
  },
  {
    "clientCorrelator": "gm1234",
    "duration": "6200",
    "media": {
      "mediaNumber": "1",
      "mediaType": "Application"
    },
    "resourceURL": "http://example.com/exampleAPI/qos/v1/tel%3A%2B19585550100/appliedQosFeatures/app001"
  }
],
"resourceURL": "http://example.com/exampleAPI/qos/v1/tel%3A%2B19585550100/appliedQosFeatures"
}}

```

## D.4 Request to apply a predefined QoS feature on the end user connection (section 6.2.5.1)

Request:

```

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

```

```

{"qosFeatureData": {
  "clientCorrelator": "v1234",
  "duration": "7200",
  "media": {
    "flowStatus": "EnabledDownlink",
    "ipFlow": {
      "flowDescription": {
        "otherPartyIpAddress": {"ipV4Address": "192.0.2.1"},
        "otherPartyPortNumber": {"port": "10"},
        "direction": "Downlink",
        "protocol": "TCP",
        "userIpAddress": {"ipV4Address": "192.0.2.10"},
        "userPortNumber": {"port": "100"}
      },
      "flowNumber": "1"
    },
    "mediaNumber": "1"
  },
  "predefinedQosFeatureId": "hdv1080"
}}

```

Response:

```

HTTP/1.1 201 Created
Content-Type: application/json
Location: http://example.com/exampleAPI/qos/v1/tel%3A%2B19585550100/appliedQosFeatures/hdv1080
Content-Length: nnnn

```

Date: Mon, 28 Jul 2011 17:51:59 GMT

```
{
  "qosFeatureData": {
    "clientCorrelator": "v1234",
    "duration": "7200",
    "media": {
      "flowStatus": "EnabledDownlink",
      "ipFlow": {
        "flowDescription": {
          "otherPartyIpAddress": {"ipV4Address": "192.0.2.1"},
          "otherPartyPortNumber": {"port": "10"},
          "direction": "Downlink",
          "protocol": "TCP",
          "userIpAddress": {"ipV4Address": "192.0.2.10"},
          "userPortNumber": {"port": "100"}
        },
        "flowNumber": "1"
      },
      "mediaNumber": "1"
    },
    "predefinedQosFeatureId": "hdv1080",
    "resourceURL": "http://example.com/exampleAPI/qos/v1/tel%3A%2B19585550100/appliedQosFeatures/hdv1080"
  }
}
```

## D.5 Request to apply a predefined QoS feature on the end user connection, response with location of created resource (section 6.2.5.2)

Request:

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

```
{
  "qosFeatureData": {
    "clientCorrelator": "v1234",
    "duration": "7200",
    "media": {
      "flowStatus": "EnabledDownlink",
      "ipFlow": {
        "flowDescription": {
          "otherPartyIpAddress": {"ipV4Address": "192.0.2.1"},
          "otherPartyPortNumber": {"port": "10"},
          "direction": "Downlink",
          "protocol": "TCP",
          "userIpAddress": {"ipV4Address": "192.0.2.10"},
          "userPortNumber": {"port": "100"}
        },
        "flowNumber": "1"
      },
      "mediaNumber": "1"
    },
  },
}
```

```
"predefinedQosFeatureId": "hdv1080"
}}
```

Response:

```
HTTP/1.1 201 Created
Content-Type: application/json
Location: http://example.com/exampleAPI/qos/v1/tel%3A%2B19585550100/appliedQosFeatures/hdv1080
Content-Length: nnnn
Date: Mon, 28 Jul 2011 17:51:59 GMT

{"resourceReference": {"resourceURL":
"http://example.com/exampleAPI/qos/v1/tel%3A%2B19585550100/appliedQosFeatures/hdv1080"}}
```

## D.6 Request to apply a custom QoS feature on the end user connection (section 6.2.5.3)

Request:

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

{"qosFeatureData": {
  "clientCorrelator": "vcnf123",
  "duration": "5400",
  "media": [
    {
      "bandwidth": {
        "maxDownlinkBitRate": "7000000",
        "minDownlinkBitRate": "7000000"
      },
      "ipFlow": {
        "flowDescription": [
          {
            "otherPartyIpAddress": {"ipV4Address": "192.0.2.2"},
            "otherPartyPortNumber": {"port": "12"},
            "direction": "Downlink",
            "protocol": "TCP",
            "userIpAddress": {"ipV4Address": "192.0.2.12"},
            "userPortNumber": {"port": "102"}
          },
          {
            "otherPartyIpAddress": {"ipV4Address": "192.0.2.2"},
            "otherPartyPortNumber": {"port": "13"},
            "direction": "Uplink",
            "protocol": "TCP",
            "userIpAddress": {"ipV4Address": "192.0.2.12"},
            "userPortNumber": {"port": "103"}
          }
        ],
        "flowNumber": "1"
      }
    }
  ]
}
```



```

    },
    "mediaNumber": "1",
    "mediaType": "Video"
  },
  {
    "bandwidth": {
      "maxDownlinkBitRate": "48000",
      "minDownlinkBitRate": "48000"
    },
    "ipFlow": {
      "flowDescription": [
        {
          "otherPartyIpAddress": {"ipV4Address": "192.0.2.2"},
          "otherPartyPortNumber": {"port": "14"},
          "direction": "Downlink",
          "protocol": "TCP",
          "userIpAddress": {"ipV4Address": "192.0.2.12"},
          "userPortNumber": {"port": "104"}
        },
        {
          "otherPartyIpAddress": {"ipV4Address": "192.0.2.2"},
          "otherPartyPortNumber": {"port": "15"},
          "direction": "Uplink",
          "protocol": "TCP",
          "userIpAddress": {"ipV4Address": "192.0.2.12"},
          "userPortNumber": {"port": "105"}
        }
      ],
      "flowNumber": "1"
    },
    "mediaNumber": "2",
    "mediaType": "Audio"
  }
],
"reservationPriority": "Medium"
}}

```

## Response:

```

HTTP/1.1 201 Created
Content-Type: application/json
Location: http://example.com/exampleAPI/qos/v1/tel%3A%2B19585550100/appliedQosFeatures/app100
Content-Length: nnnn
Date: Mon, 28 Jul 2011 17:51:59 GMT

```

```

{"qosFeatureData": {
  "clientCorrelator": "vcnf123",
  "duration": "5400",
  "media": [
    {
      "bandwidth": {
        "maxDownlinkBitRate": "7000000",
        "minDownlinkBitRate": "7000000"
      },
      "ipFlow": {
        "flowDescription": [

```

```

    {
      "otherPartyIpAddress": {"ipV4Address": "192.0.2.2"},
      "otherPartyPortNumber": {"port": "12"},
      "direction": "Downlink",
      "protocol": "TCP",
      "userIpAddress": {"ipV4Address": "192.0.2.12"},
      "userPortNumber": {"port": "102"}
    },
    {
      "otherPartyIpAddress": {"ipV4Address": "192.0.2.2"},
      "otherPartyPortNumber": {"port": "13"},
      "direction": "Uplink",
      "protocol": "TCP",
      "userIpAddress": {"ipV4Address": "192.0.2.12"},
      "userPortNumber": {"port": "103"}
    }
  ],
  "flowNumber": "1"
},
"mediaNumber": "1",
"mediaType": "Video"
},
{
  "bandwidth": {
    "maxDownlinkBitRate": "48000",
    "minDownlinkBitRate": "48000"
  },
  "ipFlow": {
    "flowDescription": [
      {
        "otherPartyIpAddress": {"ipV4Address": "192.0.2.2"},
        "otherPartyPortNumber": {"port": "14"},
        "direction": "Downlink",
        "protocol": "TCP",
        "userIpAddress": {"ipV4Address": "192.0.2.12"},
        "userPortNumber": {"port": "104"}
      },
      {
        "otherPartyIpAddress": {"ipV4Address": "192.0.2.2"},
        "otherPartyPortNumber": {"port": "15"},
        "direction": "Uplink",
        "protocol": "TCP",
        "userIpAddress": {"ipV4Address": "192.0.2.12"},
        "userPortNumber": {"port": "105"}
      }
    ]
  },
  "flowNumber": "1"
},
"mediaNumber": "2",
"mediaType": "Audio"
}
],
"reservationPriority": "Medium",
"resourceURL": "http://example.com/exampleAPI/qos/v1/tel%3A%2B19585550100/appliedQosFeatures/app100"
}}

```

## D.7 Application service provider sponsoring user's data usage (section 6.2.5.4)

Request:

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

{"qosFeatureData": {
  "clientCorrelator": "v1234",
  "media": {
    "bandwidth": {
      "maxDownlinkBitRate": "7000000",
      "minDownlinkBitRate": "7000000"
    },
    "flowStatus": "EnabledDownlink",
    "ipFlow": {
      "flowDescription": {
        "otherPartyIpAddress": {"ipV4Address": "192.0.2.2"},
        "otherPartyPortNumber": {"port": "12"},
        "direction": "Downlink",
        "protocol": "TCP",
        "userIpAddress": {"ipV4Address": "192.0.2.12"},
        "userPortNumber": {"port": "102"}
      },
      "flowNumber": "1"
    },
    "mediaNumber": "1",
    "mediaType": "Video"
  },
  "reservationPriority": "High",
  "sponsorId": "sp100",
  "volume": "36000000"
}}
```

Response:

```
HTTP/1.1 201 Created
Content-Type: application/json
Location: http://example.com/exampleAPI/qos/v1/tel%3A%2B19585550100/appliedQosFeatures/app100
Content-Length: nnnn
Date: Mon, 28 Jul 2011 17:51:59 GMT

{"qosFeatureData": {
  "clientCorrelator": "v1234",
  "media": {
    "bandwidth": {
      "maxDownlinkBitRate": "7000000",
      "minDownlinkBitRate": "7000000"
    },
    "flowStatus": "EnabledDownlink",
    "ipFlow": {
```

```

    "flowDescription": {
      "otherPartyIpAddress": {"ipV4Address": "192.0.2.2"},
      "otherPartyPortNumber": {"port": "12"},
      "direction": "Downlink",
      "protocol": "TCP",
      "userIpAddress": {"ipV4Address": "192.0.2.12"},
      "userPortNumber": {"port": "102"}
    },
    "flowNumber": "1"
  },
  "mediaNumber": "1",
  "mediaType": "Video"
},
"reservationPriority": "High",
"resourceURL": "http://example.com/exampleAPI/qos/v1/tel%3A%2B19585550100/appliedQosFeatures/app200",
"sponsorId": "sp100",
"volume": "36000000"
}}

```

## D.8 Request to apply a custom QoS feature on the end user connection, which is not supported by the server (section 6.2.5.5)

Request:

```

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

```

```

{"qosFeatureData": {
  "clientCorrelator": "v1234",
  "duration": "5400",
  "media": {
    "bandwidth": {
      "maxDownlinkBitRate": "7000000",
      "minDownlinkBitRate": "7000000"
    },
    "flowStatus": "EnabledDownlink",
    "ipFlow": {
      "flowDescription": {
        "otherPartyIpAddress": {"ipV4Address": "192.0.2.2"},
        "otherPartyPortNumber": {"port": "12"},
        "direction": "Downlink",
        "protocol": "TCP",
        "userIpAddress": {"ipV4Address": "192.0.2.12"},
        "userPortNumber": {"port": "102"}
      },
      "flowNumber": "1"
    },
    "mediaNumber": "1",
    "mediaType": "Video"
  },
}

```

```
"reservationPriority": "Medium"
}}
```

Response:

```
HTTP/1.1 403 Forbidden
Content-Type: application/json
Content-Length: nnnn
Date: Mon, 28 Jul 2011 17:51:59 GMT

{"requestError": {"policyException": {
  "messageId": "POL1032",
  "text": "Custom QoS features are not supported"
}}}
```

## D.9 Retrieving an individual QoS feature applied on the end user connection (section 6.3.3.1)

Request:

```
GET /exampleAPI/qos/v1/tel%3A%2B19585550100/appliedQosFeatures/hdv1080 HTTP/1.1
Accept: application/json
Host: example.com
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnnn
Date: Mon, 28 Jul 2011 17:51:59 GMT

{"qosFeatureData": {
  "clientCorrelator": "v1234",
  "duration": "4200",
  "media": {
    "flowStatus": "EnabledDownlink",
    "ipFlow": {
      "flowDescription": {
        "otherPartyIpAddress": {"ipV4Address": "192.0.2.1"},
        "otherPartyPortNumber": {"port": "10"},
        "direction": "Downlink",
        "protocol": "TCP",
        "userIpAddress": {"ipV4Address": "192.0.2.10"},
        "userPortNumber": {"port": "100"}
      },
      "flowNumber": "1"
    },
    "mediaNumber": "1"
  },
  "predefinedQosFeatureId": "hdv1080",
  "resourceURL": "http://example.com/exampleAPI/qos/v1/tel%3A%2B19585550100/appliedQosFeatures/hdv1080"
}}
```

## D.10 Updating attributes for QoS feature applied on the end user connection (section 6.3.4.1)

Request:

```
PUT /exampleAPI/qos/v1/tel%3A%2B19585550100/appliedQosFeatures/hdv1080 HTTP/1.1
Content-Type: application/json
Content-Length: nnn
Accept: application/json
Host: example.com

{"qosFeatureData": {
  "clientCorrelator": "v1234",
  "duration": "7200",
  "media": {
    "flowStatus": "EnabledDownlink",
    "ipFlow": {
      "flowDescription": {
        "otherPartyIpAddress": {"ipV4Address": "192.0.2.1"},
        "otherPartyPortNumber": {"port": "10"},
        "direction": "Downlink",
        "protocol": "TCP",
        "userIpAddress": {"ipV4Address": "192.0.2.10"},
        "userPortNumber": {"port": "100"}
      },
      "flowNumber": "1"
    },
    "mediaNumber": "1"
  },
  "predefinedQosFeatureId": "hdv1080",
  "resourceURL": "http://example.com/exampleAPI/qos/v1/tel%3A%2B19585550100/appliedQosFeatures/hdv1080"
}}
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnnn
Date: Mon, 28 Jul 2011 17:51:59 GMT

{"qosFeatureData": {
  "clientCorrelator": "v1234",
  "duration": "7200",
  "media": {
    "flowStatus": "EnabledDownlink",
    "ipFlow": {
      "flowDescription": {
        "otherPartyIpAddress": {"ipV4Address": "192.0.2.1"},
        "otherPartyPortNumber": {"port": "10"},
        "direction": "Downlink",
        "protocol": "TCP",
        "userIpAddress": {"ipV4Address": "192.0.2.10"},
        "userPortNumber": {"port": "100"}
      },
      "flowNumber": "1"
    }
  }
}
```

```
    },  
    "mediaNumber": "1"  
  },  
  "predefinedQosFeatureId": "hdv1080",  
  "resourceURL": "http://example.com/exampleAPI/qos/v1/tel%3A%2B19585550100/appliedQosFeatures/hdv1080"  
}}
```

## D.11 Removing a QoS feature which is applied to the end user connection (section 6.3.6.1)

Request:

```
DELETE /exampleAPI/qos/v1/tel%3A%2B19585550100/appliedQosFeatures/hdv1080 HTTP/1.1  
Accept: application/json  
Host: example.com
```

Response:

```
HTTP/1.1 204 No Content  
Date: Fri, 28 Jun 2013 17:51:59 GMT
```

## D.12 Retrieving duration time for QoS feature applied on the end user connection (section 6.4.3.1)

Request:

```
GET /exampleAPI/qos/v1/tel%3A%2B19585550100/appliedQosFeatures/hdv1080/duration HTTP/1.1  
Accept: application/json  
Host: example.com
```

Response:

```
HTTP/1.1 200 OK  
Content-Type: application/json  
Content-Length: nnnn  
Date: Mon, 28 Jul 2011 17:51:59 GMT
```

```
{"duration": "7200"}
```

## D.13 Retrieving IP flow status on media number level for QoS feature applied on the end user connection (section 6.4.3.2)

Request:

```
GET /exampleAPI/qos/v1/tel%3A%2B19585550100/appliedQosFeatures/hdv1080/media/1/flowStatus HTTP/1.1  
Accept: application/json  
Host: example.com
```

Response:

```
HTTP/1.1 200 OK
```

```
Content-Type: application/json
Content-Length: nnnn
Date: Mon, 28 Jul 2011 17:51:59 GMT
```

```
{"flowStatus": "EnabledDownlink"}
```

## D.14 Retrieving IP flow status on IP flow number level for QoS feature applied on the end user connection (section 6.4.3.3)

Request:

```
GET /exampleAPI/qos/v1/tel%3A%2B19585550100/appliedQosFeatures/hdv1080/media/1/flowStatus HTTP/1.1
Accept: application/json
Host: example.com
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnnn
Date: Mon, 28 Jul 2011 17:51:59 GMT
```

```
{"flowStatus": "EnabledDownlink"}
```

## D.15 Updating duration time for QoS feature applied on the end user connection (section 6.4.4.1)

Request:

```
PUT /exampleAPI/qos/v1/tel%3A%2B19585550100/appliedQosFeatures/hdv1080/duration HTTP/1.1
Content-Type: application/json
Content-Length: nnn
Accept: application/json
Host: example.com
```

```
{"duration": "9000"}
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnnn
Date: Mon, 28 Jul 2011 17:51:59 GMT
```

```
{"duration": "9000"}
```

## D.16 Updating IP flow status on IP flow number level for QoS feature applied on the end user connection (section 6.4.4.2)

Request:



```

PUT /exampleAPI/qos/v1/tel%3A%2B19585550100/appliedQosFeatures/hdv1080/media/1/flow/1/flowStatus HTTP/1.1
Content-Type: application/json
Content-Length: nnn
Accept: application/json
Host: example.com

{"flowStatus": "Disabled"}

```

Response:

```

HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnnn
Date: Mon, 28 Jul 2011 17:51:59 GMT

{"flowStatus": "Disabled"}

```

## D.17 Retrieving all subscriptions to notifications related to QoS features (section 6.5.3.1)

Request:

```

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

```

Response:

```

HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnnn
Date: Fri, 28 Jun 2013 17:51:59 GMT

{"qosFeaturesSubscriptionList": {
  "appliedQosFeaturesSubscriptionList": {
    "appliedQosFeaturesSubscription": {
      "callbackReference": {
        "callbackData": "efgh",
        "notifyURL": "http://application.example.com/qos/notifications/77777"
      },
      "clientCorrelator": "v1234",
      "duration": "6000",
      "eventType": "AppliedQosFeatureReleased",
      "resourceURL": "http://example.com/exampleAPI/qos/v1/tel%3A%2B19585550100/subscriptions/appliedQosFeatures/sub002"
    },
    "resourceURL": "http://example.com/exampleAPI/qos/v1/tel%3A%2B19585550100/subscriptions/appliedQosFeatures"
  },
  "predefinedQosFeaturesSubscriptionList": {
    "predefinedQosFeaturesSubscription": {
      "callbackReference": {
        "callbackData": "abcd",
        "notifyURL": "http://application.example.com/qos/notifications/77777"
      },
      "clientCorrelator": "p1234",

```

```

    "duration": "7000",
    "predefinedQosFeatureId": "audio16",
    "resourceURL": "http://example.com/exampleAPI/qos/v1/tel%3A%2B19585550100/subscriptions/predefinedQosFeatures/sub001"
  },
  "resourceURL": "http://example.com/exampleAPI/qos/v1/tel%3A%2B19585550100/subscriptions/predefinedQosFeatures"
},
"resourceURL": "http://example.com/exampleAPI/qos/v1/tel%3A%2B19585550100/subscriptions"
}}

```

## D.18 Retrieving all subscriptions to notifications on availability of predefined QoS features (section 6.6.3.1)

Request:

```

GET /exampleAPI/qos/v1/tel%3A%2B19585550100/subscriptions/predefinedQosFeatures HTTP/1.1
Accept: application/json
Host: example.com

```

Response:

```

HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnnn
Date: Fri, 28 Jun 2013 17:51:59 GMT

{"predefinedQosFeaturesSubscriptionList": {
  "predefinedQosFeaturesSubscription": {
    "callbackReference": {
      "callbackData": "abcd",
      "notifyURL": "http://application.example.com/qos/notifications/77777"
    },
    "clientCorrelator": "p1234",
    "duration": "7000",
    "predefinedQosFeatureId": "audio16",
    "resourceURL": "http://example.com/exampleAPI/qos/v1/tel%3A%2B19585550100/subscriptions/predefinedQosFeatures/sub001"
  },
  "resourceURL": "http://example.com/exampleAPI/qos/v1/tel%3A%2B19585550100/subscriptions/predefinedQosFeatures "
}}

```

## D.19 Creating a new subscription for notifications on availability of predefined QoS features (section 6.6.5.1)

Request:

```

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

{"predefinedQosFeaturesSubscription": {
  "callbackReference": {
    "callbackData": "abcd",

```

```
"notifyURL": "http://application.example.com/qos/notifications/77777"  
},  
"clientCorrelator": "p1234",  
"duration": "7000",  
"predefinedQosFeatureId": "audio16"  
}}
```

**Response:**

```
HTTP/1.1 201 Created  
Content-Type: application/json  
Location: http://example.com/exampleAPI/qos/v1/tel%3A%2B19585550100/subscriptions/predefinedQosFeatures/sub001  
Content-Length: nnnn  
Date: Fri, 28 Jun 2013 17:51:59 GMT  
  
{  
  "predefinedQosFeaturesSubscription": {  
    "callbackReference": {  
      "callbackData": "abcd",  
      "notifyURL": "http://application.example.com/qos/notifications/77777"  
    },  
    "clientCorrelator": "p1234",  
    "duration": "7000",  
    "predefinedQosFeatureId": "audio16",  
    "resourceURL": " http://example.com/exampleAPI/qos/v1/tel%3A%2B19585550100/subscriptions/predefinedQosFeatures/sub001 "  
  }  
}
```

## D.20 Retrieving an individual subscriptions to notifications on availability of predefined QoS features (section 6.7.3.1)

**Request:**

```
GET /exampleAPI/qos/v1/tel%3A%2B19585550100/subscriptions/predefinedQosFeatures/sub001 HTTP/1.1  
Accept: application/json  
Host: example.com
```

**Response:**

```
HTTP/1.1 200 OK  
Content-Type: application/json  
Content-Length: nnnn  
Date: Fri, 28 Jun 2013 17:51:59 GMT  
  
{  
  "predefinedQosFeaturesSubscription": {  
    "callbackReference": {  
      "callbackData": "abcd",  
      "notifyURL": "http://application.example.com/qos/notifications/77777"  
    },  
    "clientCorrelator": "p1234",  
    "duration": "7000",  
    "predefinedQosFeatureId": "audio16",  
    "resourceURL": " http://example.com/exampleAPI/qos/v1/tel%3A%2B19585550100/subscriptions/predefinedQosFeatures/sub001 "  
  }  
}
```

## D.21 Cancelling a subscription to notifications on availability of predefined QoS features (section 6.7.6.1)

Request:

```
DELETE /exampleAPI/qos/v1/tel%3A%2B19585550100/subscriptions/predefinedQosFeatures/sub001 HTTP/1.1
Accept: application/json
Host: example.com
```

Response:

```
HTTP/1.1 204 No Content
Date: Fri, 28 Jun 2013 17:51:59 GMT
```

## D.22 Retrieving all subscriptions to notifications about events occurring for applied QoS features (section 6.8.3.1)

Request:

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

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnnn
Date: Fri, 28 Jun 2013 17:51:59 GMT

{"appliedQosFeaturesSubscriptionList": {
  "appliedQosFeaturesSubscription": {
    "callbackReference": {
      "callbackData": "efgh",
      "notifyURL": "http://application.example.com/qos/notifications/77777"
    },
    "clientCorrelator": "v1234",
    "duration": "6000",
    "eventType": "AppliedQosFeatureReleased",
    "resourceURL": "http://example.com/exampleAPI/qos/v1/tel%3A%2B19585550100/subscriptions/appliedQosFeatures/sub002 "
  },
  "resourceURL": "http://example.com/exampleAPI/qos/v1/tel%3A%2B19585550100/subscriptions/appliedQosFeatures"
}}
```

## D.23 Creating a new subscription for notifications about events occurring for applied QoS features (section 6.8.5.1)

Request:

```
POST /exampleAPI/qos/v1/tel%3A%2B19585550100/subscriptions/appliedQosFeatures HTTP/1.1
Content-Type: application/json
Content-Length: nnnn
```

```
Accept: application/json
Host: example.com
```

```
{"appliedQosFeaturesSubscription": {
  "callbackReference": {
    "callbackData": "efgh",
    "notifyURL": "http://application.example.com/qos/notifications/77777"
  },
  "clientCorrelator": "v1234",
  "duration": "6000",
  "eventType": "AppliedQosFeatureReleased"
}}
```

Response:

```
HTTP/1.1 201 Created
Content-Type: application/json
Location: http://example.com/exampleAPI/qos/v1/tel%3A%2B19585550100/subscriptions/appliedQosFeatures/sub002
Content-Length: nnnn
Date: Fri, 28 Jun 2013 17:51:59 GMT

{"appliedQosFeaturesSubscription": {
  "callbackReference": {
    "callbackData": "efgh",
    "notifyURL": "http://application.example.com/qos/notifications/77777"
  },
  "clientCorrelator": "v1234",
  "duration": "6000",
  "eventType": "AppliedQosFeatureReleased",
  "resourceURL": "http://example.com/exampleAPI/qos/v1/tel%3A%2B19585550100/subscriptions/appliedQosFeatures/sub002 "
}}
```

## D.24 Retrieving an individual subscription to notifications about events occurring for applied QoS features (section 6.9.3.1)

Request:

```
GET /exampleAPI/qos/v1/tel%3A%2B19585550100/subscriptions/appliedQosFeatures/sub002 HTTP/1.1
Accept: application/json
Host: example.com
```

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: nnnn
Date: Fri, 28 Jun 2013 17:51:59 GMT

{"appliedQosFeaturesSubscription": {
  "callbackReference": {
    "callbackData": "efgh",
    "notifyURL": "http://application.example.com/qos/notifications/77777"
  },
  "clientCorrelator": "v1234",
```

```
"duration": "6000",
"eventType": "AppliedQoSFeatureReleased",
"resourceURL": "http://example.com/exampleAPI/qos/v1/tel%3A%2B19585550100/subscriptions/appliedQoSFeatures/sub002 "
}}
```

## D.25 Cancelling a subscription to notifications about events occurring for applied QoS features (section 6.9.6.1)

Request:

```
DELETE /exampleAPI/qos/v1/tel%3A%2B19585550100/subscriptions/appliedQoSFeatures/sub002 HTTP/1.1
Accept: application/json
Host: example.com
```

Response:

```
HTTP/1.1 204 No Content
Date: Fri, 28 Jun 2013 17:51:59 GMT
```

## D.26 Notifying a client that a predefined QoS feature is available to the end user (section 6.10.5.1)

Request:

```
POST /qos/notifications/77777 HTTP/1.1
Content-Type: application/json
Content-Length: nnnn
Accept: application/json
Host: example.com

{"predefinedQoSAvailabilityNotification": {
  "callbackData": "abcd",
  "link": {
    "href": "http://example.com/exampleAPI/qos/v1/tel%3A%2B19585550100/subscriptions/predefinedQoSFeatures/sub001",
    "rel": "PredefinedQoSfeaturesSubscription"
  },
  "predefinedQoSFeatureId": " audio16"
}}
```

Response:

```
HTTP/1.1 204 No Content
Date: Fri, 28 Jun 2013 17:51:59 GMT
```

## D.27 Notifying a client about an event occurring for applied QoS features (section 6.11.5.1)

Request:

```
POST /qos/notifications/77777 HTTP/1.1
Content-Type: application/json
```

Content-Length: nnnn  
Accept: application/json  
Host: example.com

```
{
  "appliedQosFeaturesNotification": {
    "callbackData": "efgh",
    "eventDescription": "Duration time for the feature has elapsed",
    "eventType": "AppliedQosFeatureReleased",
    "link": [
      {
        "href": "http://example.com/exampleAPI/qos/v1/tel%3A%2B19585550100/subscriptions/appliedQosFeatures/sub002",
        "rel": "AppliedQoSfeaturesSubscription"
      },
      {
        "href": "http://example.com/exampleAPI/qos/v1/tel%3A%2B19585550100/appliedQosFeatures/hdv1080",
        "rel": "QoSFeatureData"
      }
    ]
  }
}
```

**Response:**

HTTP/1.1 204 No Content  
Date: Fri, 28 Jun 2013 17:51:59 GMT

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

This specification does not have a baseline specification; however some of the resources/methods defined in this specification are equivalent to the operations defined in Parlay X Application-driven Quality of Service specification [3GPP\_TS\_29.199-17].

The table below illustrates the mapping between REST resources/methods defined in this specification and Parlay X equivalent operations.

REST Resource	REST Method	REST Section reference	Parlay X equivalent operation
Applied QoS features on a temporary basis	GET	6.2.3	getQoSStatus 1)
	POST	6.2.5	applyQoSFeature 2)
Individual applied QoS feature on a temporary basis	PUT	6.3.4	modifyQoSFeature
	DELETE	6.3.6	removeQoSFeature
All subscriptions to applied QoS features event notifications	POST	6.8.5	startQoSNotification
Individual subscription to applied QoS features event notifications	DELETE	6.9.6	stopQoSNotification
Client notification about an event occurred for applied QoS features	POST	6.11.5	notifyQoSEvent

**Table 1 Parlay X operations mapping**

- 1) Note: The ParlayX operation getQoSStatus is similar to but not quite the same as this REST method because GET method supports retrieval of QoS features that are applied on an end user connection on a temporary basis only.
- 2) The ParlayX operation applyQoSFeature is similar to but not quite the same as this REST method because POST method supports requests to apply QoS features on an end user connection on a temporary basis only.



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

The following table lists all Quality of Services data structure elements that can be accessed individually as Light-weight Resources. For each Light-weight Resource, the following information is provided: corresponding root element name, root element type and [ResourceRelPath] string.

Type of Light-weight Resources (and references to data structures)	Element/attribute that can be accessed as Light-weight Resource	Root element name for the Light-weight Resource	Root element type for the Light-weight Resource	[ResourceRelPath] string that needs to be appended to the corresponding Heavy-weight Resource URL
Duration time and data volume for QoS feature (5.2.2.4)	duration	duration	xsd:unsignedInt	duration
	volume	volume	xsd:unsignedInt	volume
Bandwidth information for QoS feature (5.2.2.5)	bandwidth	bandwidth	BandwidthInformation	media/{mediaNumber}/bandwidth
IP Flow status On media number level (5.2.2.5) On IP flow number level (5.2.2.7)	flowStatus	flowStatus	FlowStatus	media/{mediaNumber}/flowStatus
	flowStatus	flowStatus	FlowStatus	media/{mediaNumber}/flow/{flowNumber}/flowStatus

Note: When appending [ResourceRelPath] string to its Heavy-weight Resource URL, variables within curly brackets “{}” such as: {mediaNumber} and {flowNumber} have to be replaced with their real value.

## Appendix G. Authorization aspects (Normative)

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

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

The RESTful Quality of Service API MAY support the authorization framework defined in [Autho4API\_10].

A RESTful Quality of Service 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 Quality of Service 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_qos.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_qos.subscr	Provide access to all defined operations regarding subscriptions to notifications for predefined and applied QoS features.	No
oma_rest_qos.subscr_predef	Provide access to all defined operations regarding subscriptions to notifications for predefined QoS features.	No

**Table 2 Scope values for RESTful Quality of Service API**

##### G.1.1.2 Downscoping

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

- “oma\_rest\_qos.subscr”
- “oma\_rest\_qos.subscr\_predef”

### 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 Quality of Service API map to the REST resources and methods of this API. In these tables, the root “oma\_rest\_qos.” of scope values is omitted for readability reasons.

Resource	URL Base URL: http://{serverRoot}/qos/{apiVersion} {userId}	Section reference	HTTP verbs			
			GET	PUT	POST	DELETE
Predefined QoS features available to a user	/predefinedQosFeatures	6.1	all_{apiVersion}	n/a	n/a	n/a

Table 3 Required scope values for: Retrieval of available predefined QoS features

Resource	URL Base URL: http://{serverRoot}/qos/{apiVersion} {userId}	Section reference	HTTP verbs			
			GET	PUT	POST	DELETE
QoS features applied on an end user connection	/appliedQosFeatures	6.2	all_{apiVersion}	n/a	all_{apiVersion}	n/a
Individual QoS feature applied on an end user connection	/appliedQosFeatures/{featureId}	6.3	all_{apiVersion}	all_{apiVersion}	n/a	all_{apiVersion}
Individual attribute for QoS feature applied on an end user connection	/appliedQosFeatures/{featureId}/{ResourceRelPath}	6.4	all_{apiVersion}	all_{apiVersion}	n/a	n/a

Table 4 Required scope values for: Management of applied QoS features

Resource	URL Base URL: http://{serverRoot}/qos/{apiVersion}/ {userId}	Section reference	HTTP verbs			
			GET	PUT	POST	DELETE
All subscriptions to QoS notifications	/subscriptions	6.5	all_{apiVersion}, or subscr	n/a	n/a	n/a

Table 5 Required scope values for: Retrieval of all subscriptions regarding QoS features

Resource	URL Base URL: http://{serverRoot}/qos/{apiVersion} /{userId}	Section reference	HTTP verbs			
			GET	PUT	POST	DELETE
All subscriptions to notifications for predefined QoS features	/subscriptions/predefinedQosFeatures	6.6	all_{apiVersion}, or subscr, or subscr_predef	n/a	all_{apiVersion}, or subscr, or subscr_predef	n/a
Individual subscription to notifications for predefined QoS features	/subscriptions/predefinedQosFeatures/{subscriptionId}	6.7	all_{apiVersion}, or subscr, or subscr_predef	n/a	n/a	all_{apiVersion}, or subscr, or subscr_predef

**Table 6 Required scope values for: Management of subscriptions to notifications for predefined QoS features**

Resource	URL Base URL: http://{serverRoot}/qos/{apiVersion} /{userId}	Section reference	HTTP verbs			
			GET	PUT	POST	DELETE
All subscriptions to notifications for QoS features applied on an end user connection	/subscriptions/appliedQosFeatures	6.8	all_{apiVersion}, or subscr	n/a	all_{apiVersion}, or subscr	n/a
Individual subscription to notifications for QoS features applied on an end user connection	/subscriptions/appliedQosFeatures/{subscriptionId}	6.9	all_{apiVersion}, or subscr	n/a	n/a	all_{apiVersion}, or subscr

**Table 7 Required scope values for: Management of subscriptions to notifications for applied QoS features**

Resource	URL Base URL: <Specified by the client>	Section reference	HTTP verbs			
			GET	PUT	POST	DELETE
Client notification about availability of predefined QoS features	Specified by client when the subscription is created or provisioned	6.10	n/a	n/a	all_{apiVersion}, or subscr, or subscr_predef	n/a

**Table 8 Required scope values for: Notifications for predefined QoS features**

Resource	URL Base URL: <Specified by the client>	Section reference	HTTP verbs			
			GET	PUT	POST	DELETE
Client notification about an event occurred for QoS features applied on an end user connection	Specified by client when the subscription is created or provisioned	6.11	n/a	n/a	all_{apiVersion}, or subscr	n/a

**Table 9 Required scope values for: Notifications for applied QoS features**

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

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

An 'acr' URI of the form 'acr:auth', where 'auth' 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:auth' in a resource URL in place of a {userId} when the the RESTful Quality of Service API is used in combination with [Autho4API\_10].

In the case the RESTful Quality of Service API supports [Autho4API\_10], the server:

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