|  |  |
| --- | --- |
| 3GPP TS 29.175 V18.2.0 (2024-09) | |
| Technical Specification | |
| 3rd Generation Partnership Project;  Technical Specification Group Core Network and Terminals;  IP Multimedia Subsystem;  IP Multimedia Subsystem (IMS) Application Server (AS) Services  Stage 3  (Release 18) | |
|  | |
|  |  |
|  | |
| The present document has been developed within the 3rd Generation Partnership Project (3GPP TM) and may be further elaborated for the purposes of 3GPP. The present document has not been subject to any approval process by the 3GPPOrganizational Partners and shall not be implemented. This Specification is provided for future development work within 3GPPonly. The Organizational Partners accept no liability for any use of this Specification. Specifications and Reports for implementation of the 3GPP TM system should be obtained via the 3GPP Organizational Partners' Publications Offices. | |

|  |
| --- |
|  |
| ***3GPP***  Postal address  3GPP support office address  650 Route des Lucioles - Sophia Antipolis  Valbonne - FRANCE  Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16  Internet  http://www.3gpp.org |
| ***Copyright Notification***  No part may be reproduced except as authorized by written permission. The copyright and the foregoing restriction extend to reproduction in all media.  © 2024, 3GPP Organizational Partners (ARIB, ATIS, CCSA, ETSI, TSDSI, TTA, TTC).  All rights reserved.  UMTS™ is a Trade Mark of ETSI registered for the benefit of its members  3GPP™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners LTE™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners  GSM® and the GSM logo are registered and owned by the GSM Association |

Contents

Foreword 6

1 Scope 8

2 References 8

3 Definitions, symbols and abbreviations 9

3.1 Definitions 9

3.2 Symbols 9

3.3 Abbreviations 9

4 Overview 9

5 Services offered by the IMS AS 10

5.1 Introduction 10

5.2 Nimsas\_SessionEventControl Service 10

5.2.1 Service Description 10

5.2.2 Service Operations 10

5.2.2.1 Introduction 10

5.2.2.1A Subscribe 11

5.2.2.2 Notify 11

5.2.2.2.1 General 11

5.2.2.2.2 Notification for Session Event 11

5.3 Nimsas\_MediaControl Service 12

5.3.1 Service Description 12

5.3.2 Service Operations 12

5.3.2.1 Introduction 12

5.3.2.2 MediaInstruction 12

5.3.2.2.1 General 12

5.3.2.2.2 Media Instruction 12

6 API Definitions 13

6.1 Nimsas\_SessionEventControl Service API 13

6.1.1 API URI 13

6.1.2 Usage of HTTP 13

6.1.2.1 General 13

6.1.2.2 HTTP standard headers 14

6.1.2.2.1 General 14

6.1.2.2.2 Content type 14

6.1.2.3 HTTP custom headers 14

6.1.3 Resources 14

6.1.3.1 Overview 14

6.1.3.2 Resource: Session Event Subscriptions 15

6.1.3.2.1 Description 15

6.1.3.2.2 Resource Definition 15

6.1.3.2.3 Resource Standard Methods 15

6.1.3.2.4 Resource Custom Operations 15

6.1.4 Custom Operations without associated resources 15

6.1.5 Notifications 15

6.1.5.1 General 15

6.1.5.2 Session Event Notification 16

6.1.5.2.1 Description 16

6.1.5.2.2 Target URI 16

6.1.5.2.3 Standard Methods 16

6.1.6 Data Model 17

6.1.6.1 General 17

6.1.6.2 Structured data types 18

6.1.6.2.1 Introduction 18

6.1.6.2.2 Type: SessionEventNotification 19

6.1.6.2.3 Type: NotificationEvent 19

6.1.6.2.4 Type: SessionInfo 19

6.1.6.2.5 Type: MediaInfo 20

6.1.6.2.6 Type: DcMediaSpec 20

6.1.6.3 Simple data types and enumerations 20

6.1.6.3.1 Introduction 20

6.1.6.3.2 Simple data types 20

6.1.6.3.3 Enumeration: EventType 21

6.1.6.3.4 Enumeration: MediaType 21

6.1.6.3.5 Enumeration: SessionCase 21

6.1.6.3.6 Enumeration: EventInitiator 21

6.1.6.4 Data types describing alternative data types or combinations of data types 22

6.1.6.5 Binary data 22

6.1.7 Error Handling 22

6.1.7.1 General 22

6.1.7.2 Protocol Errors 22

6.1.7.3 Application Errors 22

6.1.8 Feature negotiation 22

6.1.9 Security 22

6.1.10 HTTP redirection 23

6.2 Nimsas\_MediaControl Service API 23

6.2.1 API URI 23

6.2.2 Usage of HTTP 23

6.2.2.1 General 23

6.2.2.2 HTTP standard headers 23

6.2.2.2.1 General 23

6.2.2.2.2 Content type 24

6.2.2.3 HTTP custom headers 24

6.2.3 Resources 24

6.2.3.1 Overview 24

6.2.3.2 Resource: Individual call session 24

6.2.3.2.1 Description 24

6.2.3.2.2 Resource Definition 25

6.2.3.2.3 Resource Standard Methods 25

6.2.3.2.4 Resource Custom Operations 25

6.2.4 Custom Operations without associated resources 26

6.2.5 Notifications 26

6.2.6 Data Model 26

6.2.6.1 General 26

6.2.6.2 Structured data types 27

6.2.6.2.1 Introduction 27

6.2.6.2.2 Type: MediaInstructionData 27

6.2.6.2.3 Type: MediaInstructions 28

6.2.6.2.4 Type: DcMediaSpecification 29

6.2.6.2.5 Type: ArMediaSpecification 29

6.2.6.2.6 Type: Mdc2EndpointInfo 30

6.2.6.2.7 Type: AudioVideoReNegotiationInd 30

6.2.6.3 Simple data types and enumerations 31

6.2.6.3.1 Introduction 31

6.2.6.3.2 Simple data types 31

6.2.6.3.3 Enumeration: MediaInstruction 31

6.2.6.3.4 Enumeration: MediaConnSide 31

6.2.6.4 Data types describing alternative data types or combinations of data types 31

6.2.6.5 Binary data 31

6.2.7 Error Handling 32

6.2.7.1 General 32

6.2.7.2 Protocol Errors 32

6.2.7.3 Application Errors 32

6.2.8 Feature negotiation 32

6.2.9 Security 32

6.2.10 HTTP redirection 32

Annex A (normative): OpenAPI specification 34

A.1 General 34

A.2 Nimsas\_SessionEventControl API 34

A.3 Nimsas\_MediaControl API 37

Annex B (informative): Change history 42

# Foreword

This Technical Specification has been produced by the 3rd Generation Partnership Project (3GPP).

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

x the first digit:

1 presented to TSG for information;

2 presented to TSG for approval;

3 or greater indicates TSG approved document under change control.

y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.

z the third digit is incremented when editorial only changes have been incorporated in the document.

In the present document, modal verbs have the following meanings:

**shall** indicates a mandatory requirement to do something

**shall not** indicates an interdiction (prohibition) to do something

The constructions "shall" and "shall not" are confined to the context of normative provisions, and do not appear in Technical Reports.

The constructions "must" and "must not" are not used as substitutes for "shall" and "shall not". Their use is avoided insofar as possible, and they are not used in a normative context except in a direct citation from an external, referenced, non-3GPP document, or so as to maintain continuity of style when extending or modifying the provisions of such a referenced document.

**should** indicates a recommendation to do something

**should not** indicates a recommendation not to do something

**may** indicates permission to do something

**need not** indicates permission not to do something

The construction "may not" is ambiguous and is not used in normative elements. The unambiguous constructions "might not" or "shall not" are used instead, depending upon the meaning intended.

**can** indicates that something is possible

**cannot** indicates that something is impossible

The constructions "can" and "cannot" are not substitutes for "may" and "need not".

**will** indicates that something is certain or expected to happen as a result of action taken by an agency the behaviour of which is outside the scope of the present document

**will not** indicates that something is certain or expected not to happen as a result of action taken by an agency the behaviour of which is outside the scope of the present document

**might** indicates a likelihood that something will happen as a result of action taken by some agency the behaviour of which is outside the scope of the present document

**might not** indicates a likelihood that something will not happen as a result of action taken by some agency the behaviour of which is outside the scope of the present document

In addition:

**is** (or any other verb in the indicative mood) indicates a statement of fact

**is not** (or any other negative verb in the indicative mood) indicates a statement of fact

The constructions "is" and "is not" do not indicate requirements.

# 1 Scope

The present document specifies the stage 3 protocol and data model for the Nimsas Service Based Interface. It provides stage 3 protocol definitions and message flows, and specifies the API for each service offered by the IMS AS.

The 5G System stage 2 architecture and procedures are specified in TS 23.501 [2] and TS 23.502 [3].

The IP Multimedia Subsystem (IMS) supporting DC architecture and procedures are specified in TS 23.228 [14].

The Technical Realization of the Service Based Architecture and the Principles and Guidelines for Services Definition are specified in TS 29.500 [4] and TS 29.501 [5].

# 2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non‑specific.

- For a specific reference, subsequent revisions do not apply.

- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[1] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".

[2] 3GPP TS 23.501: "System Architecture for the 5G System; Stage 2".

[3] 3GPP TS 23.502: "Procedures for the 5G System; Stage 2".

[4] 3GPP TS 29.500: "5G System; Technical Realization of Service Based Architecture; Stage 3".

[5] 3GPP TS 29.501: "5G System; Principles and Guidelines for Services Definition; Stage 3".

[6] OpenAPI: "OpenAPI Specification Version 3.0.0", <https://spec.openapis.org/oas/v3.0.0>.

[7] 3GPP TR 21.900: "Technical Specification Group working methods".

[8] 3GPP TS 33.501: "Security architecture and procedures for 5G system".

[9] IETF RFC 6749: "The OAuth 2.0 Authorization Framework".

[10] 3GPP TS 29.510: "5G System; Network Function Repository Services; Stage 3".

[11] IETF RFC 7540: "Hypertext Transfer Protocol Version 2 (HTTP/2)".

[12] IETF RFC 8259: "The JavaScript Object Notation (JSON) Data Interchange Format".

[13] IETF RFC 7807: "Problem Details for HTTP APIs".

[14] 3GPP TS 23.228: "IP Multimedia Subsystem (IMS); Stage 2".

[15] 3GPP TS 29.562: "5G System; Home Subscriber Server (HSS) Services; Stage 3".

[16] 3GPP TS 29.571: "5G System; Common Data Types for Service Based Interfaces; Stage 3".

[17] 3GPP TS 29.176: "IP Multimedia Subsystems (IMS); Media Function (MF) Services; Stage 3".

[18] 3GPP TS 24.610:" Communication HOLD (HOLD) using IP Multimedia (IM) Core Network (CN) subsystem; Protocol specification".

[19] 3GPP TS 24.186:" IMS Data Channel applications; Protocol specification".

[20] IETF RFC 8259: "The JavaScript Object Notation (JSON) Data Interchange Format".

# 3 Definitions, symbols and abbreviations

## 3.1 Definitions

For the purposes of the present document, the terms and definitions given in 3GPP TR 21.905 [1] and the following apply. A term defined in the present document takes precedence over the definition of the same term, if any, in 3GPP TR 21.905 [1].

For the purposes of the present document, the following terms and definitions given in 3GPP TS 23.228 [14] apply:

**- Bootstrap data channel**

**- DC Application Server**

## 3.2 Symbols

For the purposes of the present document, the following symbols given in 3GPP TS 23.228 [14] apply:

Cr Reference Point between an AS and an MRFC for media control.

Mr′ Reference Point between an AS and an MRFC for session control.

DC1 Reference Point between an SBI capable IMS AS and DCSF.

DC2 Reference Point between an SBI capable IMS AS and MF.

## 3.3 Abbreviations

For the purposes of the present document, the abbreviations given in 3GPP TR 21.905 [1] and the following apply. An abbreviation defined in the present document takes precedence over the definition of the same abbreviation, if any, in 3GPP TR 21.905 [1].

AR Augmented Reality

DC Data Channel

DCSF Data Channel Signalling Function

MF Media Function

# 4 Overview

The IP Multimedia Subsystem (IMS) architecture supporting the Data Channel services is defined in 3GPP TS 23.228 [14].

The IP Multimedia Subsystem (IMS) Application Server (AS) is enhanced to offer services to DCSF via the Nimsas service-based interface (see 3GPP TS 23.228 [14]) and support the following functionalities:

- The IMS AS interacts with the DCSF via DC1 for event notifications;

- The IMS AS receives the data channel control instructions from the DCSF and accordingly interacts with the MF via DC2 or with MRF via Mr'/Cr for data channel media resource management.

Figure 4-1 provides the reference model (in service-based interface representation and in reference point representation), with focus on the IMS AS.



Figure 4-1: Reference model – IMS AS

DC1 is the reference point between an SBI capable IMS AS and DCSF.

# 5 Services offered by the IMS AS

## 5.1 Introduction

The IMS AS offers to other NFs the following services:

- Nimsas\_SessionEventControl. This service enables the consumer to be notified about session events when served IMS subscribers take part in IMS sessions.

- Nimsas\_MediaControl. This service enables the consumer to control IMS AS handling of media flow within an IMS session.

Table 5.1-1 summarizes the SBI services produced by the IMS AS and the corresponding APIs defined for this specification.

Table 5.1-1: API Descriptions

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Service Name | Clause | Description | OpenAPI Specification File | apiName | Annex |
| Nimsas\_SessionEventControl | 6.1 | IMS AS SessionEventControl Service | TS29175\_Nimsas\_SessionEventControl.yaml | nimsas-sec | A.2 |
| Nimsas\_MediaControl | 6.2 | IMS AS MediaControl Service | TS29175\_Nimsas\_MediaControl.yaml | nimsas-mc | A.3 |

## 5.2 Nimsas\_SessionEventControl Service

### 5.2.1 Service Description

The Nimsas\_SessionEventControl service enables the consumer to be notified about session events when served IMS subscribers take part in IMS sessions.

### 5.2.2 Service Operations

#### 5.2.2.1 Introduction

The Service operations defined for the Nimsas\_SessionEventControl service are as follows:

- Notify: It allows the IMS AS to notify the IMS session control events of a specific IMS subscriber to NFs. This service operation reports the event to the consumer that has subscribed implicitly.

NOTE: Explicit subscription to receive session events is another service operation defined in TS 23.228 [14] which has not specified in this Release. In this Release an implicit subscription is assumed where the IMS AS notifies a configured or discovered the NF consumer of a call event through a Notify.

Table 5.2.2.1-1: Service operations supported by the Nimsas\_SessionEventControl service

|  |  |  |  |
| --- | --- | --- | --- |
| Service Operations | Description | Operation  Semantics | Example Consumer(s) |
| Notify | Notification about IMS session control events of a specific IMS subscriber. | Subscribe/Notify | DCSF |

#### 5.2.2.1A Subscribe

This is a pseudo operation, the IMS AS does not actually provide Subscribe service operation through Nimsas\_SessionEventControl service in this Release. The actual subscription is implicitly subscribed with the SessionEventNotificationUri.

#### 5.2.2.2 Notify

##### 5.2.2.2.1 General

This service operation is invoked by IMS AS and used to enables the IMS AS to notify the NF service consumers (e.g. DCSF) of IMS session events related to a specific served IMS subscriber requesting use of IMS media (e.g. data channel media).

##### 5.2.2.2.2 Notification for Session Event



Figure 5.2.2.2.2-1: Notification for Session Event

1. If the IMS AS observes the IMS session control events of a specific IMS subscriber that requesting use of IMS data channel media, and it determines based on the IMS subscriber's service profile that the events need to be notified to the DCSF, the IMS AS shall send a POST request to the SessionEventNotificationUri as specified in clause 6.1.5.2.2.

The IMS AS discovers the DCSF and its session event notification URI based on local configuration or via NRF, as specified in clause AC.7.1 of 3GPP TS 23.228 [14].

2a. Upon success, the NF Service Consumer responds with "204 No Content". The NF service consumer may invoke the Nimsas\_MediaControl service before sending the session event notification response to IMS AS if the NF service consumer determines that media resources for data channel are needed.

2b. On failure or redirection:

- If the NF Service Consumer does not consider the "SessionEventNotificationUri" as a valid notification URI, the NF Service Consumer shall return "404 Not Found" status code with the ProblemDetails IE providing details of the error.

- In the case of redirection, the NF service consumer shall return 3xx status code, which shall contain a Location header with an URI pointing to the endpoint of another NF service consumer endpoint. A RedirectResponse IE may be included in the content of POST response.

## 5.3 Nimsas\_MediaControl Service

### 5.3.1 Service Description

The Nimsas\_MediaControl service enables the consumer to control IMS AS handling of media flow within an IMS session. The service can be used by the consumer before responding to a Nimsas\_SessionEventControl\_Notity request.

### 5.3.2 Service Operations

#### 5.3.2.1 Introduction

The Service operations defined for the Nimsas\_MediaControl service are as follows:

- MediaInstruction: It provides instructions to the IMS AS for all media flows a consumer wants to control based on its policies for the received IMS session event, and that may require media resource reservation on MF.

NOTE: For the case when a specific media flow needs to be terminated in MF media resource (i.e. termination of a Data Channel media descriptor offered by one of the IMS subscribers) or originated by the MF (i.e. origination of a data channel media descriptor to be offered towards one of the IMS subscribers), the consumer must provide a complete MF media specification including information required by MF to know how to terminate or originate the media flow.

Table 5.3.2.1-1: Service operations supported by the Nimsas\_MediaControl service

|  |  |  |  |
| --- | --- | --- | --- |
| Service Operations | Description | Operation  Semantics | Example Consumer(s) |
| MediaInstruction | Instructions to the IMS AS for all media flows a consumer wants to control based on its policies for the received IMS session event | Request/Response | DCSF |

#### 5.3.2.2 MediaInstruction

##### 5.3.2.2.1 General

This service operation provides instructions to the IMS AS for all media flows a consumer wants to control based on its policies for the received IMS session event, and that may require resource reservation in media resource

The MediaInstruction service operation defined for Nimsas\_MediaControl service is used:

- to enable the NF consumer to instruct the IMS AS how to manage the MF media resource. E.g., setting up the bootstrap data channel with MF both for originating and terminating side, terminating the media flow of the originating UE to MF and updating the MF resource with MDC2 media endpoint information of DC Application Server.

##### 5.3.2.2.2 Media Instruction

Figure 5.3.2.2.2-1 shows a scenario where the NF service consumer (e.g. DCSF) sends a request to the IMS AS to instruct the IMS AS how to manage the MF media resource. The request contains the IMS session identity (/{sessionId}) which is assigned by the IMS AS during the IMS session setup and will be notified to consumer via Nimsas\_SessionEventControl notify service operation. After that, the consumer shall reuse the session ID it received from the IMS AS for referencing the same session.



Figure 5.3.2.2.2-1: Media instruction from NF service consumer

1. The NF service consumer shall send a POST request to the resource representing the individual session resource in the IMS AS. The payload body of the POST request shall contain the media instructions for all the media flows that the NF Service Consumer want to control based on its policies for the received IMS session event via Nimsas\_SessionEventControl notification.

2a. Upon success, the IMS AS responds with "200 OK" with the message body containing a representation of the created media instruction data.

2b. On failure or redirection:

In the case of failure, one of the HTTP status code listed in Table 6.2.3.2.4.2.2-2 shall be returned. For a 4xx/5xx response, the message body shall contain a ProblemDetails structure with the "cause" attribute set to one of the application errors listed in Table 6.2.3.2.4.2.2-2.

In the case of redirection, the NF service producer shall return 3xx status code, which shall contain a Location header with an URI pointing to the endpoint of another NF service producer endpoint.

# 6 API Definitions

## 6.1 Nimsas\_SessionEventControl Service API

### 6.1.1 API URI

The Nimsas\_SessionEventControl service shall use the Nimsas\_SessionEventControl API.

The API URI of the Nimsas\_SessionEventControl API shall be:

**{apiRoot}/<apiName>/<apiVersion>**

The request URIs used in HTTP requests from the NF service consumer towards the NF service producer shall have the Resource URI structure defined in clause 4.4.1 of 3GPP TS 29.501 [5], i.e.:

**{apiRoot}/<apiName>/<apiVersion>/<apiSpecificResourceUriPart>**

with the following components:

- The {apiRoot} shall be set as described in 3GPP TS 29.501 [5].

- The <apiName> shall be "nimsas-sec".

- The <apiVersion> shall be "v1".

- The <apiSpecificResourceUriPart> shall be set as described in clause 6.1.3.

### 6.1.2 Usage of HTTP

#### 6.1.2.1 General

HTTP/2, IETF RFC 7540 [11], shall be used as specified in clause 5 of 3GPP TS 29.500 [4].

HTTP/2 shall be transported as specified in clause 5.3 of 3GPP TS 29.500 [4].

The OpenAPI [6] specification of HTTP messages and content bodies for the Nimsas\_SessionEventControl API is contained in Annex A.

#### 6.1.2.2 HTTP standard headers

##### 6.1.2.2.1 General

See clause 5.2.2 of 3GPP TS 29.500 [4] for the usage of HTTP standard headers.

##### 6.1.2.2.2 Content type

JSON, IETF RFC 8259 [12], shall be used as content type of the HTTP bodies specified in the present specification as specified in clause 5.4 of 3GPP TS 29.500 [4]. The use of the JSON format shall be signalled by the content type "application/json".

"Problem Details" JSON object shall be used to indicate additional details of the error in a HTTP response body and shall be signalled by the content type "application/problem+json", as defined in IETF RFC 7807 [13].

#### 6.1.2.3 HTTP custom headers

The mandatory HTTP custom header fields specified in clause 5.2.3.2 of 3GPP TS 29.500 [4] shall be supported, and the optional HTTP custom header fields specified in clause 5.2.3.3 of 3GPP TS 29.500 [4] may be supported.

### 6.1.3 Resources

#### 6.1.3.1 Overview

This clause describes the structure for the Resource URIs and the resources and methods used for the service.

Figure 6.1.3.1-1 depicts the resource URIs structure for the Nimsas\_SessionEventControl service API.



Figure 6.1.3.1-1: Resource URI structure of the Nimsas\_SessionEventControl API

Table 6.1.3.1-1 provides an overview of the resources and applicable HTTP methods.

Table 6.1.3.1-1: Resources and methods overview

|  |  |  |  |
| --- | --- | --- | --- |
| Resource purpose/name | Resource URI (relative path after API URI) | HTTP method or custom operation | Description (service operation) |
| Session Event Subscriptions  (Collection) | /session-event-subscriptions | POST | This is a pseudo resource. |

#### 6.1.3.2 Resource: Session Event Subscriptions

##### 6.1.3.2.1 Description

##### 6.1.3.2.2 Resource Definition

Resource URI: **{apiRoot}/nimsas-sec/<apiVersion>/session-event-subscriptions**

This resource shall support the resource URI variables defined in table 6.1.3.2.2-1.

Table 6.1.3.2.2-1: Resource URI variables for this resource

|  |  |  |
| --- | --- | --- |
| Name | Data type | Definition |
| apiRoot | string | See clause 6.1.1 |

##### 6.1.3.2.3 Resource Standard Methods

6.1.3.2.3.1 POST

This method shall support the URI query parameters specified in table 6.1.3.2.3.1-1.

Table 6.1.3.2.3.1-1: URI query parameters supported by the POST method on this resource

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Data type | P | Cardinality | Description | Applicability |
| n/a |  |  |  |  |  |

This method shall support the request data structures specified in table 6.1.3.2.3.1-2 and the response data structures and response codes specified in table 6.1.3.2.3.1-3.

Table 6.1.3.2.3.1-2: Data structures supported by the POST Request Body on this resource

|  |  |  |  |
| --- | --- | --- | --- |
| Data type | P | Cardinality | Description |
| Any |  |  |  |

Table 6.1.3.2.3.1-3: Data structures supported by the POST Response Body on this resource

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data type | P | Cardinality | Response  codes | Description |
| n/a |  |  |  |  |
| NOTE: The mandatory HTTP error status code for the POST method listed in Table 5.2.7.1-1 of 3GPP TS 29.500 [4] also apply. | | | | |

##### 6.1.3.2.4 Resource Custom Operations

None.

### 6.1.4 Custom Operations without associated resources

None in this release of the specification.

### 6.1.5 Notifications

#### 6.1.5.1 General

Notifications shall comply to clause 6.2 of 3GPP TS 29.500 [4] and clause 4.6.2.3 of 3GPP TS 29.501 [5].

Table 6.1.5.1-1: Notifications overview

|  |  |  |  |
| --- | --- | --- | --- |
| Notification | Callback URI | HTTP method or custom operation | Description  (service operation) |
| Session Event Notification | {SessionEventNotificationUri} | POST | Report the session event observed. |

#### 6.1.5.2 Session Event Notification

##### 6.1.5.2.1 Description

The Session Event Notification is used by the IMS AS to report the observed Session Event to notification endpoints.

##### 6.1.5.2.2 Target URI

The Callback URI "{SessionEventNotificationUri}" shall be used with the callback URI variables defined in table 6.1.5.2.2-1.

Table 6.1.5.2.2-1: Callback URI variables

|  |  |
| --- | --- |
| Name | Definition |
| SessionEventNotificationUri | The notification URI of the Service Consumer (i.e., DCSF) to receive the session events.  In this release, the SessionEventNotificationUri of the Service Consumer is locally configured in the IMS AS or discovered by the IMS AS via NRF (in the default notification subscription with the "DC\_SESSION\_EVENT\_NOTIFICATION" notification type). |

##### 6.1.5.2.3 Standard Methods

6.1.5.2.3.1 POST

This method shall support the request data structures specified in table 6.1.5.2.3.1-1 and the response data structures and response codes specified in table 6.1.5.2.3.1-1.

Table 6.1.5.2.3.1-1: Data structures supported by the POST Request Body

|  |  |  |  |
| --- | --- | --- | --- |
| Data type | P | Cardinality | Description |
| SessionEventNotification | M | 1 | The session event notification to notification endpoints. |

Table 6.1.5.2.3.1-2: Data structures supported by the POST Response Body

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data type | P | Cardinality | Response codes | Description |
| n/a |  |  | 204 No Content | This case represents a successful notification of the event. |
| ProblemDetails | O | 0..1 | 404 Not Found | Indicates the session event notification has failed due to application error.  The "cause" attribute may be used to indicate one of the following application errors:  - USER\_NOT\_FOUND, e.g. if the NF Service Consumer (e.g. DCSF) does not serve this service user;  - NOTIFICATION\_URI\_NOT\_FOUND, if the NF Service Consumer (e.g. DCSF) considers the "SessionEventNotificationUri" is not recognized. |
| RedirectResponse | O | 0..1 | 307 Temporary Redirect | Temporary redirection.  (NOTE 2) |
| RedirectResponse | O | 0..1 | 308 Permanent Redirect | Permanent redirection.  (NOTE 2) |
| NOTE 1: The mandatory HTTP error status codes for the POST method listed in Table 5.2.7.1-1 of 3GPP TS 29.500 [4] also apply.  NOTE 2: RedirectResponse may be inserted by an SCP, see clause 6.10.9.1 of 3GPP TS 29.500 [4]. | | | | |

Table 6.1.5.2.3-3: Headers supported by the 307 Response Code on this endpoint

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Data type | P | Cardinality | Description |
| Location | string | M | 1 | A URI pointing to the endpoint of the NF service consumer instance to which the request should be sent. For the case, when a request is redirected to the same target resource via a different SCP, see clause 6.10.9.1 in 3GPP TS 29.500 [4]. |
| 3gpp-Sbi-Target-Nf-Id | string | O | 0..1 | Identifier of the target NF instance ID towards which the notification is redirected |

Table 6.1.5.2.3-4: Headers supported by the 308 Response Code on this endpoint

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Data type | P | Cardinality | Description |
| Location | string | M | 1 | A URI pointing to the endpoint of the NF service consumer instance to which the request should be sent. For the case, when a request is redirected to the same target resource via a different SCP, see clause 6.10.9.1 in 3GPP TS 29.500 [4]. |
| 3gpp-Sbi-Target-Nf-Id | string | O | 0..1 | Identifier of the target NF instance ID towards which the notification is redirected |

### 6.1.6 Data Model

#### 6.1.6.1 General

This clause specifies the application data model supported by the API.

Table 6.1.6.1-1 specifies the data types defined for the Nimsas\_SessionEventControl service based interface protocol.

Table 6.1.6.1-1: Nimsas\_SessionEventControl specific Data Types

|  |  |  |
| --- | --- | --- |
| Data type | Clause defined | Description |
| SessionEventNotification | 6.1.6.2.2 | The session event notification item. |
| NotificationEvent | 6.1.6.2.3 | The notification event information. |
| SessionInfo | 6.1.6.2.4 | The session information related to the notification event. |
| MediaInfo | 6.1.6.2.5 | The media information related to the notification event. |
| DcMediaSpec | 6.1.6.2.6 | The data channel media specification information which includes media description attributes. |
| EventType | 6.1.6.3.3 | The type of the event for which the notification is generated. |
| MediaType | 6.1.6.3.4 | The type of the media for which the notification is generated. |
| SessionCase | 6.1.6.3.5 | The type of the session case. |
| EventInitiator | 6.1.6.3.6 | The initiator of the event for which the notification is generated. |

Table 6.1.6.1-2 specifies data types re-used by the Nimsas\_SessionEventControl service based interface protocol from other specifications, including a reference to their respective specifications and when needed, a short description of their use within the Nimsas\_SessionEventControl service based interface.

Table 6.1.6.1-2: Nimsas\_SessionEventControl re-used Data Types

|  |  |  |
| --- | --- | --- |
| Data type | Reference | Comments |
| ImsPublicId | 3GPP TS 29.562 [15] | IMS Public Identity. |
| DcStream | 3GPP TS 29.571 [16] | Data Channel mapping and configuration information. |
| MaxMessageSize | 3GPP TS 29.571 [16] | Maximum message size of to be expected. |
| DcEndpoint | 3GPP TS 29.571 [16] | local or remote endpoint for the Data Channel. |
| SessionId | 3GPP TS 29.571 [16] | IMS Session Identity. |
| MediaId | 3GPP TS 29.571 [16] | IMS media flow identity. |
| BdcUsedBy | 3GPP TS 29.571 [16] | The party using the bootstrap data channel. |
| AppBindingInfo | 3GPP TS 29.571 [16] | The application binding information of the application data channel. |

#### 6.1.6.2 Structured data types

##### 6.1.6.2.1 Introduction

This clause defines the structures to be used in resource representations.

##### 6.1.6.2.2 Type: SessionEventNotification

Table 6.1.6.2.2-1: Definition of type SessionEventNotification

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description |
| notificationEvent | NotificationEvent | M | 1 | The NotificationEvent corresponds to a report on one subscribed event. |
| sessionId | SessionId | M | 1 | The session ID is the identity of the IMS session for which the event relates to.  This IE contains the information in the Call-ID header which is the typical header of SIP message. |
| sessionInfo | SessionInfo | C | 0..1 | The sessionInfo is the IMS session related information, e.g. the calling and called identity. It shall be contained when the attribute eventyType of the notificationEvent is one of the "SESSION\_ESTABLISHMENT\_REQUEST",  "MEDIA\_CHANGE\_REQUEST". |
| mediaInfoList | map(MediaInfo) | C | 1..N | Media info list includes a list of media related information, e.g. media identity, media type and media specifications.  The mediaId attribute within the MediaInfo data type is the key of the map.  It shall be contained when the attribute eventyType of the notificationEvent is one of the "SESSION\_ESTABLISHMENT\_REQUEST", "SESSION\_ESTABLISHMENT\_PROGRESS",  "SESSION\_ESTABLISHMENT\_ALERTING",  "SESSION\_ESTABLISHMENT\_SUCCESS",  "MEDIA\_CHANGE\_REQUEST",  "MEDIA\_CHANGE\_SUCCESS", or  "MEDIA\_CHANGE\_FAILURE". |

##### 6.1.6.2.3 Type: NotificationEvent

Table 6.1.6.2.3-1: Definition of type NotificationEvent

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description |
| eventType | EventType | M | 1 | The event type of the event for which the notification is generated. |
| eventInitiator | EventInitiator | C | 0..1 | Event initiator indicates initiator of the event, e.g. "served IMS subscriber" vs "remote IMS subscriber".  It shall be contained when the eventType is "SESSION\_ESTABLISHMENT\_REQUEST" or "MEDIA\_CHANGE\_REQUEST". |

##### 6.1.6.2.4 Type: SessionInfo

Table 6.1.6.2.4-1: Definition of type SessionInfo

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description |
| callingIdentity | ImsPublicId | O | 0..1 | The public identity of the calling IMS subscriber. |
| calledIdentity | ImsPublicId | O | 0..1 | The public identity of the called IMS Subscriber. |
| sessionCase | SessionCase | O | 0..1 | Indicator to indicate that whether this is an originating or terminating IMS session. |

##### 6.1.6.2.5 Type: MediaInfo

Table 6.1.6.2.5-1: Definition of type MediaInfo

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description |
| mediaId | MediaId | M | 1 | The media ID uniquely identifies this media item within the list. The identity is allocated by IMS AS. |
| mediaType | MediaType | M | 1 | The media type of the media for which the notification is generated. |
| dcMediaSpec | DcMediaSpec | C | 0..1 | The data channel media specification includes the relevant media attributes of interest to the consumer.  It shall be contained if the mediaType is set to "DC". |
| mediaSuspended | boolean | C | 0..1 | This IE indicates whether the media is suspended or not.  It shall be contained if the media is suspended or resumed as specified in 3GPP TS 24.610 [18] and 3GPP TS 24.186 [19].  - true: the media is suspended.  false: the media is resumed. |

##### 6.1.6.2.6 Type: DcMediaSpec

Table 6.1.6.2.6-1: Definition of type DcMediaSpec

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description |
| streams | map(DcStream) | M | 1..N | Represents the data channel mapping and configuration information when originating/terminating data channel media flows on the Mb interface.  The streamId attribute within the DcStream data type is the key of the map.  (NOTE) |
| maxMessageSize | MaxMessageSize | O | 0..1 | Represents the maximum message size of to be expected.  (NOTE) |
| receivedDcEndpoint | DcEndpoint | M | 0..1 | Represents the SCTP and DTLS endpoint information for the Data Channel received from the SDP offer/answer.  (NOTE) |
| bdcUsedby | BdcUsedBy | C | 0..1 | Represents the party using the bootstrap data channel. It shall be contained when the mediaId in the data type MediaInfo represents bootstrap data channel and the streamId set to 100 and 110. |
| appBindingInfo | AppBindingInfo | C | 0..1 | Represents the application binding information of the application data channel. It shall be contained when the mediaId in the data type MediaInfo represents application data channel. |
| NOTE: The attribute and the value are derived from the SDP received by the IMS AS in a SIP INVITE or a re-INVITE request and the corresponding response related to an IMS Data Channel. | | | | |

#### 6.1.6.3 Simple data types and enumerations

##### 6.1.6.3.1 Introduction

This clause defines simple data types and enumerations that can be referenced from data structures defined in the previous clauses.

##### 6.1.6.3.2 Simple data types

The simple data types defined in table 6.1.6.3.2-1 shall be supported.

Table 6.1.6.3.2-1: Simple data types

|  |  |  |  |
| --- | --- | --- | --- |
| Type Name | Type Definition | Description | Applicability |
|  |  |  |  |

##### 6.1.6.3.3 Enumeration: EventType

The enumeration EventType represents the type of the event for which the notification is generated. It shall comply with the provisions defined in table 6.1.6.3.3-1.

Table 6.1.6.3.3-1: Enumeration EventType

|  |  |
| --- | --- |
| Enumeration value | Description |
| "SESSION\_ESTABLISHMENT\_REQUEST" | Session establishment request event. |
| "SESSION\_ESTABLISHMENT\_PROGRESS" | Session establishment progress event. |
| "SESSION\_ESTABLISHMENT\_ALERTING" | Session establishment alerting event. |
| "SESSION\_ESTABLISHMENT\_SUCCESS" | Session establishment success event. |
| "SESSION\_ESTABLISHMENT\_FAILURE" | Session establishment failure event. |
| "MEDIA\_CHANGE\_REQUEST" | Media change request event. |
| "MEDIA\_CHANGE\_SUCCESS" | Media change success event. |
| "MEDIA\_CHANGE\_FAILURE" | Media change failure event. |
| "SESSION\_TERMINATION" | Session termination event. |

##### 6.1.6.3.4 Enumeration: MediaType

The enumeration MediaType represents the type of the media for which the notification is generated. It shall comply with the provisions defined in table 6.1.6.3.4-1.

Table 6.1.6.3.4-1: Enumeration MediaType

|  |  |
| --- | --- |
| Enumeration value | Description |
| "DC" | Data channel media type. |
| "AUDIO" | Audio media type. |
| "VIDEO" | Video media type. |

##### 6.1.6.3.5 Enumeration: SessionCase

The enumeration SessionCase represents the type of the session to for which the notification is generated. It shall comply with the provisions defined in table 6.1.6.3.5-1.

Table 6.1.6.3.5-1: Enumeration SessionCase

|  |  |
| --- | --- |
| Enumeration value | Description |
| "ORIGINATING\_IMS\_SESSION" | Originating IMS session. |
| "TERMINATING\_IMS\_SESSION" | Terminating IMS session. |

##### 6.1.6.3.6 Enumeration: EventInitiator

The enumeration eventInitiator represents the initiator of the event for which the notification is generated. It shall comply with the provisions defined in table 6.1.6.3.6-1.

Table 6.1.6.3.6-1: Enumeration EventInitiator

|  |  |
| --- | --- |
| Enumeration value | Description |
| "SERVED\_IMS\_SUBSCRIBER" | Served IMS subscriber. |
| "REMOTE\_IMS\_SUBSCRIBER" | Remote IMS subscriber. |

#### 6.1.6.4 Data types describing alternative data types or combinations of data types

None in this release of the specification.

#### 6.1.6.5 Binary data

None in this release of the specification.

### 6.1.7 Error Handling

#### 6.1.7.1 General

For the Nimsas\_SessionEventControl API, HTTP error responses shall be supported as specified in clause 4.8 of 3GPP TS 29.501 [5]. Protocol errors and application errors specified in table 5.2.7.2-1 of 3GPP TS 29.500 [4] shall be supported for an HTTP method if the corresponding HTTP status codes are specified as mandatory for that HTTP method in table 5.2.7.1-1 of 3GPP TS 29.500 [4].

In addition, the requirements in the following clauses are applicable for the Nimsas\_SessionEventControl API.

#### 6.1.7.2 Protocol Errors

No specific procedures for the Nimsas\_SessionEventControl service are specified.

#### 6.1.7.3 Application Errors

The application errors defined for the Nimsas\_SessionEventControl service are listed in Table 6.1.7.3-1.

Table 6.1.7.3-1: Application errors

|  |  |  |
| --- | --- | --- |
| Application Error | HTTP status code | Description |
| USER\_NOT\_FOUND | 404 Not Found | The service user is not served by this DCSF, e.g. no data channel configurations for this service user is found. |
| NOTIFICATION\_URI\_NOT\_FOUND | 404 Not Found | The session event notification URI is not recognized by the NF service consumer (i.e. DCSF). |

### 6.1.8 Feature negotiation

The optional features in table 6.1.8-1 are defined for the Nimsas\_SessionEventControl API. They shall be negotiated using the extensibility mechanism defined in clause 6.6 of 3GPP TS 29.500 [4].

Table 6.1.8-1: Supported Features

|  |  |  |
| --- | --- | --- |
| Feature number | Feature Name | Description |
|  |  |  |

### 6.1.9 Security

As indicated in 3GPP TS 33.501 [8] and 3GPP TS 29.500 [4], the access to the Nimsas\_SessionEventControl API may be authorized by means of the OAuth2 protocol (see IETF RFC 6749 [9]), based on local configuration, using the "Client Credentials" authorization grant, where the NRF (see 3GPP TS 29.510 [10]) plays the role of the authorization server.

If OAuth2 is used, an NF Service Consumer, prior to consuming services offered by the Nimsas\_SessionEventControl API, shall obtain a "token" from the authorization server, by invoking the Access Token Request service, as described in 3GPP TS 29.510 [10], clause 5.4.2.2.

NOTE: When multiple NRFs are deployed in a network, the NRF used as authorization server is the same NRF that the NF Service Consumer used for discovering the Nimsas\_SessionEventControl service.

The Nimsas\_SessionEventControl API defines a single scope "nimsas-sessioneventcontrol" for the entire service, and it does not define any additional scopes at resource or operation level.

### 6.1.10 HTTP redirection

An HTTP request may be redirected to a different IMS AS service instance when using direct or indirect communications (see 3GPP TS 29.500 [4]).

An SCP that reselects a different NF consumer (e.g. DCSF) instance will return the NF Instance ID of the new NF consumer instance in the 3gpp-Sbi-Producer-Id header, as specified in clause 6.10.3.4 of 3GPP TS 29.500 [4].

If an NF consumer (e.g. DCSF) redirects a service request to a different NF consumer using an 307 Temporary Redirect or 308 Permanent Redirect status code, the identity of the new NF consumer towards which the service request is redirected shall be indicated in the 3gpp-Sbi-Target-Nf-Id header of the 307 Temporary Redirect or 308 Permanent Redirect response as specified in clause 6.10.9.1 of 3GPP TS 29.500 [4].

## 6.2 Nimsas\_MediaControl Service API

### 6.2.1 API URI

The Nimsas\_MediaControl service shall use the Nimsas\_MediaControl API.

The API URI of the Nimsas\_MediaControl API shall be:

**{apiRoot}/<apiName>/<apiVersion>**

The request URIs used in HTTP requests from the NF service consumer towards the NF service producer shall have the Resource URI structure defined in clause 4.4.1 of 3GPP TS 29.501 [5], i.e.:

**{apiRoot}/<apiName>/<apiVersion>/<apiSpecificResourceUriPart>**

with the following components:

- The {apiRoot} shall be set as described in 3GPP TS 29.501 [5].

- The <apiName> shall be "nimsas-mc".

- The <apiVersion> shall be "v1".

- The <apiSpecificResourceUriPart> shall be set as described in clause 6.2.3.

### 6.2.2 Usage of HTTP

#### 6.2.2.1 General

HTTP/2, IETF RFC 7540 [11], shall be used as specified in clause 5 of 3GPP TS 29.500 [4].

HTTP/2 shall be transported as specified in clause 5.3 of 3GPP TS 29.500 [4].

The OpenAPI [6] specification of HTTP messages and content bodies for the Nimsas\_MediaControl API is contained in Annex A.

#### 6.2.2.2 HTTP standard headers

##### 6.2.2.2.1 General

See clause 5.2.2 of 3GPP TS 29.500 [4] for the usage of HTTP standard headers.

##### 6.2.2.2.2 Content type

JSON, IETF RFC 8259 [12], shall be used as content type of the HTTP bodies specified in the present specification as specified in clause 5.4 of 3GPP TS 29.500 [4]. The use of the JSON format shall be signalled by the content type "application/json".

"Problem Details" JSON object shall be used to indicate additional details of the error in a HTTP response body and shall be signalled by the content type "application/problem+json", as defined in IETF RFC 7807 [13].

#### 6.2.2.3 HTTP custom headers

The mandatory HTTP custom header fields specified in clause 5.2.3.2 of 3GPP TS 29.500 [4] shall be supported, and the optional HTTP custom header fields specified in clause 5.2.3.3 of 3GPP TS 29.500 [4] may be supported.

### 6.2.3 Resources

#### 6.2.3.1 Overview

This clause describes the structure for the Resource URIs and the resources and methods used for the service.

Figure 6.2.3.1-1 depicts the resource URIs structure for the Nimsas\_MediaControl service API.



Figure 6.2.3.1-1: Resource URI structure of the Nimsas\_MediaControl API

Table 6.2.3.1-1 provides an overview of the resources and applicable HTTP methods.

Table 6.2.3.1-1: Resources and methods overview

|  |  |  |  |
| --- | --- | --- | --- |
| Resource purpose/name | Resource URI (relative path after API URI) | HTTP method or custom operation | Description (service operation) |
| Individual call session  (Document) | /call-sessions/{sessionId}/media-instruction | media-instruction  (POST) | Instructs the IMS AS on how to control the media resources. |

#### 6.2.3.2 Resource: Individual call session

##### 6.2.3.2.1 Description

This resource represents an individual call session created in the IMS AS.

This resource is modelled with the Document resource archetype (see clause C.1 of 3GPP TS 29.501 [5]).

##### 6.2.3.2.2 Resource Definition

Resource URI: **{apiRoot}/nimsas-mc/<apiVersion>/call-sessions/{sessionId}**

This resource shall support the resource URI variables defined in table 6.2.3.2.2-1.

Table 6.2.3.2.2-1: Resource URI variables for this resource

|  |  |  |
| --- | --- | --- |
| Name | Data type | Definition |
| apiRoot | string | See clause 6.2.1 |
| sessionId | SessionId | Session ID assigned by the IMS AS during the IMS session setup and will be notified to consumer via Nimsas\_SessionEventControl service. The consumer shall reuse the session ID it received from the IMS AS for referencing the same session. |

##### 6.2.3.2.3 Resource Standard Methods

None.

##### 6.2.3.2.4 Resource Custom Operations

6.2.3.2.4.1 Overview

Table 6.2.3.2.4.1-1: Custom operations

|  |  |  |  |
| --- | --- | --- | --- |
| Operation Name | Custom operation URI | Mapped HTTP method | Description |
| media-instruction | {resourceUri}/media-instruction | POST | Update call session service operation. |

6.2.3.2.4.2 Operation: media-instruction

6.2.3.2.4.2.1 Description

6.2.3.2.4.2.2 Operation Definition

This custom operation updates an individual call session resource and provide a set of media instruction information for a given IMS session, towards IMS AS.

This operation shall support the request data structures specified in table 6.2.3.2.4.2.2-1 and the response data structure and response codes specified in table 6.2.3.2.4.2.2-2.

Table 6.2.3.2.4.2.2-1: Data structures supported by the POST Request Body on this resource

|  |  |  |  |
| --- | --- | --- | --- |
| Data type | P | Cardinality | Description |
| MediaInstructionData | M | 1 | Representation of the updates to apply to the call session. |

Table 6.2.3.2.4.2.2-2: Data structures supported by the POST Response Body on this resource

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data type | P | Cardinality | Response  codes | Description |
| MediaInstructionData | C | 0..1 | 200 OK | Successful update of the call session, when the IMS AS needs to return information in the response. |
| n/a |  |  | 204 No Content | Successful update of the call session, when the IMS AS does not need to return information in the response. |
| RedirectResponse | O | 0..1 | 307 Temporary Redirect | Temporary redirection.  (NOTE 2) |
| RedirectResponse | O | 0..1 | 308 Permanent Redirect | Permanent redirection.  (NOTE 2) |
| ProblemDetails | O | 0..1 | 400 Bad Request | Indicates the media instruction has failed due to application error.  The "cause" attribute may be used to indicate one of the following application errors:  - MEDIA\_ID\_NOT\_FOUND |
| NOTE 1: The mandatory HTTP error status codes for the POST method listed in table  5.2.7.1-1 of 3GPP TS 29.500 [4] also apply.  NOTE 2: RedirectResponse may be inserted by an SCP, see clause 6.10.9.1 of 3GPP TS 29.500 [4]. | | | | |

Table 6.2.3.3.4.2.2-3: Headers supported by the 307 Response Code on this resource

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Data type | P | Cardinality | Description |
| Location | string | M | 1 | A URI pointing to the endpoint of the NF service consumer instance to which the request should be sent. For the case, when a request is redirected to the same target resource via a different SCP, see clause 6.10.9.1 in 3GPP TS 29.500 [4]. |
| 3gpp-Sbi-Target-Nf-Id | string | O | 0..1 | Identifier of the target NF instance ID towards which the request is redirected. |

Table 6.2.3.3.4.2.2-4: Headers supported by the 308 Response Code on this resource

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Data type | P | Cardinality | Description |
| Location | string | M | 1 | A URI pointing to the endpoint of the NF service consumer instance to which the request should be sent. For the case, when a request is redirected to the same target resource via a different SCP, see clause 6.10.9.1 in 3GPP TS 29.500 [4]. |
| 3gpp-Sbi-Target-Nf-Id | string | O | 0..1 | Identifier of the target SMF (service) instance ID towards which the request is redirected. |

### 6.2.4 Custom Operations without associated resources

None in this release of the specification.

### 6.2.5 Notifications

None in this release of the specification.

### 6.2.6 Data Model

#### 6.2.6.1 General

This clause specifies the application data model supported by the API.

Table 6.2.6.1-1 specifies the data types defined for the Nimsas\_MediaControl service based interface protocol.

Table 6.2.6.1-1: Nimsas\_MediaControl specific Data Types

|  |  |  |
| --- | --- | --- |
| Data type | Clause defined | Description |
| ArMediaSpecification | 6.2.6.2.5 | The AR media specification information needed for AR communication services from application layer. |
| AudioVideoReNegotiationInd | 6.2.6.2.7 | The audio/video media re-negotiation indication information to connect the UE's audio/video media stream to MF. |
| DcMediaSpecification | 6.2.6.2.4 | The data channel media specification information needed for data channel media stream from application layer. |
| Mdc2EndpointInfo | 6.2.6.2.6 | The MDC2 media endpoint information. |
| MediaInstruction | 6.2.6.3.3 | The media instruction for handling a specific media flow. |
| MediaInstructionData | 6.2.6.2.2 | The media instruction data for a specific session. |
| MediaInstructions | 6.2.6.2.3 | The media instructions for a specific media flow. |
| MediaConnSide | 6.2.6.3.4 | The side that the audio or video media need to be connected to MF. E.g., both the originating and terminating side. |

Table 6.2.6.1-2 specifies data types re-used by the Nimsas\_MediaControl service based interface protocol from other specifications, including a reference to their respective specifications and when needed, a short description of their use within the Nimsas\_MediaControl service based interface.

Table 6.2.6.1-2: Nimsas\_MediaControl re-used Data Types

|  |  |  |
| --- | --- | --- |
| Data type | Reference | Comments |
| DcStream | 3GPP TS 29.571 [16] | Data Channel mapping and configuration information. |
| MdcEndpoint | 3GPP TS 29.571 [16] | The MDC1 and MDC2 Endpoint information. |
| MaxMessageSize | 3GPP TS 29.571 [16] | Maximum SCTP user message size. |
| Mdc2Protocol | 3GPP TS 29.176 [17] | Transport layer protocols for MDC2 interface. |
| MediaId | 3GPP TS 29.571 [16] | IMS media flow identity. |
| MediaProxy | 3GPP TS 29.571 [16] | Media proxy configuration applicable to the media flow. |
| MediaResourceType | 3GPP TS 29.571 [16] | IMS media resource type. |
| ReplaceHttpUrl | 3GPP TS 29.571 [16] | A list of replacement HTTP URL per stream ID. |
| SessionId | 3GPP TS 29.571 [16] | IMS session identity. |
| Uri | 3GPP TS 29.571 [16] | URI |
| BdcUsedBy | 3GPP TS 29.571 [16] | The party using the bootstrap data channel. |
| AppBindingInfo | 3GPP TS 29.571 [16] | The application binding information of the application data channel. |

#### 6.2.6.2 Structured data types

##### 6.2.6.2.1 Introduction

This clause defines the structures to be used in resource representations.

##### 6.2.6.2.2 Type: MediaInstructionData

Table 6.2.6.2.2-1: Definition of type MediaInstructionData

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description |
| sessionId | SessionId | M | 1 | The session ID is the identity of the IMS session for which the MediaInstructions applies. |
| mediaInstructionSet | map(MediaInstructions) | M | 1..N | The mediaInstructionSet includes a set of instructions for each media flow to control.  The key of the map shall be a (unique) valid JSON string per clause 7 of IETF RFC 8259 [20], with a maximum of 32 characters. |

##### 6.2.6.2.3 Type: MediaInstructions

Table 6.2.6.2.3-1: Definition of type MediaInstructions

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description |
| mediaId | MediaId | M | 1 | The media ID uniquely identifies this media item within the list and used by the producer (IMS AS) and the consumer for referencing purposes. The consumer reuses the media ID it received from the IMS AS for referencing the same media. This field will be null for instructions related to originating new media. |
| mediaResourceType | MediaResourceType | M | 1 | The mediaResourceType identify the Media Resource types the Media instruction is intended for. |
| mediaInstruction | MediaInstruction | C | 0..1 | mediaInstruction includes instructions to the producer (IMS AS) for handling the media. |
| dcMediaSpecification | DcMediaSpecification | C | 0..1 | dcMediaSpecification includes the description of additional media specification information which are needed for data channel media stream from application layer.  It shall be contained if the mediaResourceType is set to "DC". |
| arMediaSpecification | ArMediaSpecification | C | 0..1 | arMediaSpecification includes the description of additional media specification information which are needed for AR communication services from application layer.  It shall be contained if the mediaResourceType is set to "AR". |
| mediaProcessingUrl | Uri | C | 0..1 | The mediaProcessingURL indicates the address where MF receive service-related media instructions. This field shall be contained in the response when originating a new media. |
| audioVideoReNegotiationInd | AudioVideoReNegotiationInd | C | 0..1 | Represents the audio/video media re-negotiation indication information to connect the UE's audio/video media stream to MF which is required by the application data channel.  It shall be included if the mediaResourceType is set to "AUDIO" or "VIDEO". |

##### 6.2.6.2.4 Type: DcMediaSpecification

Table 6.2.6.2.4-1: Definition of type DcMediaSpecification

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description |
| mediaProxyConfig | MediaProxy | M | 1 | Represents the media proxy configuration on the MF.  The value shall be set to "HTTP\_PROXY" if the mediaId represents the bootstrap data channel. |
| replaceHttpUrls | map(ReplaceHttpUrl) | C | 1..N | Represents a list of replacement HTTP URL per stream ID allocated by the application layer for the specific IMS subscriber when requesting the application list (e.g. graphical user interface) via the MDC1 interface.  It shall be included if the mediaId represents a bootstrap data channel and streamId is 0 or 100.  The streamId attribute within the ReplaceHttpUrl data type is the key of the map. |
| mdc1EndpointDcsf | MdcEndpoint | C | 0..1 | Represents the MDC1 media endpoint information reserved on DCSF, e.g., the IP address and port number of DCSF.  It shall be included in the request of the media instruction if the mediaId represents a bootstrap data channel.  (NOTE) |
| mdc1EndpointMf | MdcEndpoint | C | 0..1 | Represents the negotiated MDC1 media endpoint information on MF, e.g., the IP address and port number of DC capable MF.  It shall be included in the request of the media instruction if the mediaId represents a bootstrap data channel.  (NOTE) |
| mdc2EndpointInfo | Mdc2EndpointInfo | C | 0..1 | Represents the negotiated MDC2 media endpoint information, e.g., the protocol stack, IP address and port number.  It shall be included if in the request of the media instruction the mediaId represents anapplication data channel. |
| streams | map(DcStream) | M | 1..N | Represents the data channel mapping and configuration information when originating/terminating data channel media flows on the Mb interface.  The streamId attribute within the DcStream data type is the key of the map. |
| bdcUsedby | BdcUsedBy | C | 0..1 | Represents the party using the bootstrap data channel. It shall be contained when the mediaId in the data type MediaInstructions represents bootstrap data channel and the streamId sets to 100 and 110. |
| appBindingInfo | AppBindingInfo | C | 0..1 | Represents the application binding information of the application data channel. It shall be contained when the mediaId in the data type MediaInstrutions represents application data channel. |
| NOTE: The MDC1 interface shall use "TCP/TLS/HTTP" protocol stack and the corresponding IP, TCP, TLS and HTTP related attributes in MdcEndpoint should be included. | | | | |

##### 6.2.6.2.5 Type: ArMediaSpecification

Table 6.2.6.2.5-1: Definition of type ArMediaSpecification

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description |
| mediaProcessingSpec | string | M | 1 | It specifies how the AR media stream should be processed. It indicates how the MF assists in the AR media rendering function. |

##### 6.2.6.2.6 Type: Mdc2EndpointInfo

Table 6.2.6.2.6-1: Definition of type Mdc2EndpointInfo

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description |
| mdc2Protocol | Mdc2Protocol | C | 0..1 | Represents the transport layer protocol or protocol stack for MDC2 interface when the "HTTP\_PROXY" media proxy configuration is used in MF.  It shall be included if the value of the mediaProxyConfig within the data type DcMediaSpecification is set to "HTTP\_PROXY". |
| mdc2EndpointDcAs | MdcEndpoint | C | 0..1 | Represents the negotiated MDC2 media endpoint information on DC AS.  It shall be included in the POST request message for the media control.  The attributes "tlsId", "sctpPort" and "fingerprint" within the MdcEndpoint shall be present when the value of the mediaProxyConfig within the DcMediaSpecification is set to "UDP\_PROXY".  (NOTE) |
| mdc2EndpointMf | MdcEndpoint | C | 0..1 | Represents the negotiated MDC2 media endpoint information on MF.  It shall be included in the successful POST response message for the media control API.  The attributes "tlsId", "sctpPort" and "fingerprint" within the MdcEndpoint shall not be present when the value of the mediaProxyConfig within the data type DcMediaSpecification is set to "UDP\_PROXY".  (NOTE) |
| NOTE: When the mdc2Protocol is set to "UDP/DTLS/SCTP", "TCP/TLS" or "SCTP/DTLS", the attributes "tlsId" and "fingerprint" within the data type MdcEndpoint shall be included, and the "securitySetup" may be included. When the value of mdc2Protocol is set to " UDP/DTLS/SCTP", the attribute "sctpPort" within the data type MdcEndpoint shall be included. | | | | |

##### 6.2.6.2.7 Type: AudioVideoReNegotiationInd

Table 6.2.6.2.7-1: Definition of type AudioVideoReNegotiationInd

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description |
| reNegotiation | boolean | M | 1 | When present, the IE shall indicate whether the audio or video media re-negotiation to connect the UE's media stream to MF is required by the application data channel. The media type will be represented by the value of the mediaResourceType.  - true: media re-negotiation to connect the UE's audio or video media stream to MF is required.  - false: media re-negotiation to connect the UE's audio or video media stream to MF is not required. |
| mediaConnSide | MediaConnSide | C | 0..1 | Represent the audio or video media connection side that the application data channel required.  It shall be included if the reNegotiation present and the attribute value is set to true. |

#### 6.2.6.3 Simple data types and enumerations

##### 6.2.6.3.1 Introduction

This clause defines simple data types and enumerations that can be referenced from data structures defined in the previous clauses.

##### 6.2.6.3.2 Simple data types

The simple data types defined in table 6.2.6.3.2-1 shall be supported.

Table 6.2.6.3.2-1: Simple data types

|  |  |  |
| --- | --- | --- |
| Type Name | Type Definition | Description |
|  |  |  |

##### 6.2.6.3.3 Enumeration: MediaInstruction

The enumeration MediaInstruction represents the instructions to the producer (IMS AS) for handling the media. It shall comply with the provisions defined in table 6.2.6.3.3-1.

Table 6.2.6.3.3-1: Enumeration MediaInstruction

|  |  |
| --- | --- |
| Enumeration value | Description |
| "TERMINATE\_MEDIA" | Terminate the offered media descriptor of the UE in the media resource. |
| "ORIGINATE\_MEDIA" | Originate and offer a media descriptor from the mediaResource to the UE. |
| "TERMINATE\_AND\_ORIGINATE\_MEDIA" | Terminate the offered media flow in the mediaResource from the UE and originate the same media flow from the mediaResource to the other UE. |
| "UPDATE\_MEDIA" | Update a media flow of the mediaResource previously allocated by the instructions "TerminateMedia", "OriginateMedia" and "TerminateAndOriginateMedia". |
| "DELETE\_MEDIA" | Delete a media flow of the mediaResource previously allocated by the instructions "TerminateMedia", "OriginateMedia" and "TerminateAndOriginateMedia". |
| "REJECT\_MEDIA" | Remove an offered media flow. |

##### 6.2.6.3.4 Enumeration: MediaConnSide

The enumeration MediaConnSide represents the audio or video media connection side. It shall comply with the provisions defined in table 6.2.6.3.4-1.

Table 6.2.6.3.4-1: Enumeration MediaConnSide

|  |  |
| --- | --- |
| Enumeration value | Description |
| "BOTH" | Both the originating and terminating side of the audio or video media stream shall be connected to the MF. |

#### 6.2.6.4 Data types describing alternative data types or combinations of data types

None in this release of the specification.

#### 6.2.6.5 Binary data

None in this release of the specification.

### 6.2.7 Error Handling

#### 6.2.7.1 General

For the Nimsas\_MediaControl API, HTTP error responses shall be supported as specified in clause 4.8 of 3GPP TS 29.501 [5]. Protocol errors and application errors specified in table 5.2.7.2-1 of 3GPP TS 29.500 [4] shall be supported for an HTTP method if the corresponding HTTP status codes are specified as mandatory for that HTTP method in table 5.2.7.1-1 of 3GPP TS 29.500 [4].

In addition, the requirements in the following clauses are applicable for the Nimsas\_MediaControl API.

#### 6.2.7.2 Protocol Errors

No specific procedures for the Nimsas\_MediaControl service are specified.

#### 6.2.7.3 Application Errors

The application errors defined for the Nimsas\_MediaControl service are listed in table 6.2.7.3-1.

Table 6.2.7.3-1: Application errors

|  |  |  |
| --- | --- | --- |
| Application Error | HTTP status code | Description |
| MEDIA\_ID\_NOT\_FOUND | 400 Bad Request | Indicates that the requested mediaId is not found in the IMS AS. |

### 6.2.8 Feature negotiation

The optional features in table 6.2.8-1 are defined for the Nimsas\_MediaControl API. They shall be negotiated using the extensibility mechanism defined in clause 6.6 of 3GPP TS 29.500 [4].

Table 6.2.8-1: Supported Features

|  |  |  |
| --- | --- | --- |
| Feature number | Feature Name | Description |
|  |  |  |

### 6.2.9 Security

As indicated in 3GPP TS 33.501 [8] and 3GPP TS 29.500 [4], the access to the Nimsas\_MediaControl API may be authorized by means of the OAuth2 protocol (see IETF RFC 6749 [9]), based on local configuration, using the "Client Credentials" authorization grant, where the NRF (see 3GPP TS 29.510 [10]) plays the role of the authorization server.

If OAuth2 is used, an NF Service Consumer, prior to consuming services offered by the Nimsas\_MediaControl API, shall obtain a "token" from the authorization server, by invoking the Access Token Request service, as described in 3GPP TS 29.510 [10], clause 5.4.2.2.

NOTE: When multiple NRFs are deployed in a network, the NRF used as authorization server is the same NRF that the NF Service Consumer used for discovering the Nimsas\_MediaControl service.

The Nimsas\_MediaControl API defines a single scope " nimsas-mc " for the entire service, and it does not define any additional scopes at resource or operation level.

### 6.2.10 HTTP redirection

An HTTP request may be redirected to a different IMS AS service instance when using direct or indirect communications (see 3GPP TS 29.500 [4]).

An SCP that reselects a different IMS AS producer instance will return the NF Instance ID of the new IMS AS producer instance in the 3gpp-Sbi-Producer-Id header, as specified in clause 6.10.3.4 of 3GPP TS 29.500 [4].

If an IMS AS redirects a service request to a different IMS AS using an 307 Temporary Redirect or 308 Permanent Redirect status code, the identity of the new IMS AS towards which the service request is redirected shall be indicated in the 3gpp-Sbi-Target-Nf-Id header of the 307 Temporary Redirect or 308 Permanent Redirect response as specified in clause 6.10.9.1 of 3GPP TS 29.500 [4].

Annex A (normative):  
OpenAPI specification

# A.1 General

This Annex specifies the formal definition of the API(s) defined in the present specification. It consists of OpenAPI specifications in YAML format.

This Annex takes precedence when being discrepant to other parts of the specification with respect to the encoding of information elements and methods within the API(s).

NOTE 1: The semantics and procedures, as well as conditions, e.g. for the applicability and allowed combinations of attributes or values, not expressed in the OpenAPI definitions but defined in other parts of the specification also apply.

Informative copies of the OpenAPI specification files contained in this 3GPP Technical Specification are available on a Git-based repository that uses the GitLab software version control system (see clause 5.3.1 of 3GPP TS 29.501 [5] and clause 5B of 3GPP TR 21.900 [7]).

# A.2 Nimsas\_SessionEventControl API

openapi: 3.0.0

info:

title: 'IMS AS SessionEventControl Service'

version: 1.0.0

description: |

Nimsas\_SessionEventControl Service.

© 2024, 3GPP Organizational Partners (ARIB, ATIS, CCSA, ETSI, TSDSI, TTA, TTC).

All rights reserved.

externalDocs:

description: >

3GPP TS 29.175 V18.1.0; IP Multimedia Subsystem; IP Multimedia Subsystem (IMS) Application

Server (AS) Services; Stage 3

url: 'https://www.3gpp.org/ftp/Specs/archive/29\_series/29.175/'

servers:

- url: '{apiRoot}/nimsas-sec/v1'

variables:

apiRoot:

default: https://example.com

description: apiRoot as defined in clause 4.4 of 3GPP TS 29.501

security:

- {}

- oAuth2ClientCredentials:

- nimsas-sec

paths:

/session-event-subscriptions:

post:

# This is a pseudo operation, NF service consumers shall NOT invoke this method!

summary: Nimsas\_SessionEventControl Subscribe service operation, pseudo operation

tags:

- Session Event Subscriptions (Collection)

operationId: Subscribe

requestBody:

required: true

content:

application/json:

# Unspecified schema for the JSON body, since this is not used by consumer or producer.

schema: {}

responses:

default:

$ref: 'TS29571\_CommonData.yaml#/components/responses/default'

callbacks:

eeNotification:

'{SessionEventNotificationUri}':

# The notification URI is provided by local configuration or via NRF.

post:

requestBody:

required: true

content:

application/json:

schema:

$ref: '#/components/schemas/SessionEventNotification'

responses:

'204':

description: No Content, notification was succesfull.

'307':

description: Temporary Redirect

content:

application/json:

schema:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/RedirectResponse'

headers:

Location:

description: >

The URI pointing to the resource located on the redirect target NF

service consumer.

required: true

schema:

type: string

'308':

description: Permanent Redirect

content:

application/json:

schema:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/RedirectResponse'

headers:

Location:

description: >

The URI pointing to the resource located on the redirect target NF

service consumer.

required: true

schema:

type: string

'400':

$ref: 'TS29571\_CommonData.yaml#/components/responses/400'

'401':

$ref: 'TS29571\_CommonData.yaml#/components/responses/401'

'403':

$ref: 'TS29571\_CommonData.yaml#/components/responses/403'

'404':

$ref: 'TS29571\_CommonData.yaml#/components/responses/404'

'411':

$ref: 'TS29571\_CommonData.yaml#/components/responses/411'

'413':

$ref: 'TS29571\_CommonData.yaml#/components/responses/413'

'415':

$ref: 'TS29571\_CommonData.yaml#/components/responses/415'

'429':

$ref: 'TS29571\_CommonData.yaml#/components/responses/429'

'500':

$ref: 'TS29571\_CommonData.yaml#/components/responses/500'

'502':

$ref: 'TS29571\_CommonData.yaml#/components/responses/502'

'503':

$ref: 'TS29571\_CommonData.yaml#/components/responses/503'

default:

$ref: 'TS29571\_CommonData.yaml#/components/responses/default'

components:

securitySchemes:

oAuth2ClientCredentials:

type: oauth2

flows:

clientCredentials:

tokenUrl: '{nrfApiRoot}/oauth2/token'

scopes:

nimsas-sec: Access to the Nimsas\_SessionEventControl API

schemas:

# STRUCTURED DATA TYPES

SessionEventNotification:

description: The session event notification item.

type: object

required:

- notificationEvent

- sessionId

properties:

notificationEvent:

$ref: '#/components/schemas/NotificationEvent'

sessionId:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/SessionId'

sessionInfo:

$ref: '#/components/schemas/SessionInfo'

mediaInfoList:

description: >

Contains a media related information. The mediaId attribute within the MediaInfo

data type is the key of the map.

type: object

additionalProperties:

$ref: '#/components/schemas/MediaInfo'

minItems: 1

NotificationEvent:

description: The notification event information.

type: object

required:

- eventType

properties:

eventType:

$ref: '#/components/schemas/EventType'

eventInitiator:

$ref: '#/components/schemas/EventInitiator'

SessionInfo:

description: The session information related to the notification event.

type: object

properties:

callingIdentity:

$ref: 'TS29562\_Nhss\_imsSDM.yaml#/components/schemas/ImsPublicId'

calledIdentity:

$ref: 'TS29562\_Nhss\_imsSDM.yaml#/components/schemas/ImsPublicId'

sessionCase:

$ref: '#/components/schemas/SessionCase'

MediaInfo:

description: The media information related to the notification event.

type: object

required:

- mediaId

- mediaType

properties:

mediaId:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/MediaId'

mediaType:

$ref: '#/components/schemas/MediaType'

dcMediaSpec:

$ref: '#/components/schemas/DcMediaSpec'

mediaSuspended:

type: boolean

DcMediaSpec:

description: >

The data channel media specification information contains media description attributes.

type: object

required:

- streams

- receivedDcEndpoint

properties:

streams:

description: >

Contains a data channel mapping and configuration information. The streamId

attribute within the DcStream data type is the key of the map.

type: object

additionalProperties:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/DcEndpoint'

minItems: 1

maxMessageSize:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/MaxMessageSize'

receivedDcEndpoint:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/DcEndpoint'

bdcUsedBy:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/BdcUsedBy'

appBindingInfo:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/AppBindingInfo'

# ENUMS

EventType:

description: The type of the event for which the notification is generated.

anyOf:

- type: string

enum:

- SESSION\_ESTABLISHMENT\_REQUEST

- SESSION\_ESTABLISHMENT\_PROGRESS

- SESSION\_ESTABLISHMENT\_ALERTING

- SESSION\_ESTABLISHMENT\_SUCCESS

- SESSION\_ESTABLISHMENT\_FAILURE

- MEDIA\_CHANGE\_REQUEST

- MEDIA\_CHANGE\_SUCCESS

- MEDIA\_CHANGE\_FAILURE

- SESSION\_TERMINATION

- type: string

description: >

This string provides forward-compatibility with future extensions to the enumeration

and is not used to encode content defined in the present version of this API.

MediaType:

description: The type of the media for which the notification is generated.

anyOf:

- type: string

enum:

- DC

- AUDIO

- VIDEO

- type: string

description: >

This string provides forward-compatibility with future extensions to the enumeration

and is not used to encode content defined in the present version of this API.

SessionCase:

description: The type of the session case.

anyOf:

- type: string

enum:

- ORIGINATING\_IMS\_SESSION

- TERMINATING\_IMS\_SESSION

- type: string

description: >

This string provides forward-compatibility with future extensions to the enumeration

and is not used to encode content defined in the present version of this API.

EventInitiator:

description: The initiator of the event for which the notification is generated.

anyOf:

- type: string

enum:

- SERVED\_IMS\_SUBSCRIBER

- REMOTE\_IMS\_SUBSCRIBER

- type: string

description: >

This string provides forward-compatibility with future extensions to the enumeration

and is not used to encode content defined in the present version of this API.

# A.3 Nimsas\_MediaControl API

openapi: 3.0.0

info:

title: 'IMS AS MediaControl Service'

version: 1.0.0

description: |

Nimsas\_MediaControl Service.

© 2024, 3GPP Organizational Partners (ARIB, ATIS, CCSA, ETSI, TSDSI, TTA, TTC).

All rights reserved.

externalDocs:

description: >

3GPP TS 29.175 V18.1.0; IP Multimedia Subsystem; IP Multimedia Subsystem (IMS) Application

Server (AS) Services; Stage 3

url: 'https://www.3gpp.org/ftp/Specs/archive/29\_series/29.175'

servers:

- url: '{apiRoot}/nimsas-mc/v1'

variables:

apiRoot:

default: https://example.com

description: apiRoot as defined in clause 4.4 of 3GPP TS 29.501

security:

- {}

- oAuth2ClientCredentials:

- nimsas-mc

paths:

/call-sessions/{sessionId}/media-instruction:

post:

summary: Update

operationId: UpdateCallSession

tags:

- Individual call session (Document)

parameters:

- name: sessionId

in: path

description: Session ID assigned by the IMS AS during the IMS session setup.

required: true

schema:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/SessionId'

requestBody:

description: Representation of the updates to apply to the call session.

required: true

content:

application/json:

schema:

$ref: '#/components/schemas/MediaInstructionData'

responses:

'200':

description: >

Successful update of the call session, when the IMS AS needs to return information

in the response.

content:

application/json:

schema:

$ref: '#/components/schemas/MediaInstructionData'

'204':

description: >

Successful update of the call session, when the IMS AS does not need to return

information in the response.

'307':

description: Temporary Redirect

content:

application/json:

schema:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/RedirectResponse'

headers:

Location:

description: The URI pointing to the resource located on the redirect target IMS AS.

schema:

type: string

'308':

description: Permanent Redirect

content:

application/json:

schema:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/RedirectResponse'

headers:

Location:

description: The URI pointing to the resource located on the redirect target IMS AS.

schema:

type: string

'400':

$ref: 'TS29571\_CommonData.yaml#/components/responses/400'

'401':

$ref: 'TS29571\_CommonData.yaml#/components/responses/401'

'403':

$ref: 'TS29571\_CommonData.yaml#/components/responses/403'

'404':

$ref: 'TS29571\_CommonData.yaml#/components/responses/404'

'406':

$ref: 'TS29571\_CommonData.yaml#/components/responses/406'

'411':

$ref: 'TS29571\_CommonData.yaml#/components/responses/411'

'413':

$ref: 'TS29571\_CommonData.yaml#/components/responses/413'

'415':

$ref: 'TS29571\_CommonData.yaml#/components/responses/415'

'429':

$ref: 'TS29571\_CommonData.yaml#/components/responses/429'

'500':

$ref: 'TS29571\_CommonData.yaml#/components/responses/500'

'501':

$ref: 'TS29571\_CommonData.yaml#/components/responses/501'

'503':

$ref: 'TS29571\_CommonData.yaml#/components/responses/503'

default:

$ref: 'TS29571\_CommonData.yaml#/components/responses/default'

components:

securitySchemes:

oAuth2ClientCredentials:

type: oauth2

flows:

clientCredentials:

tokenUrl: '{nrfApiRoot}/oauth2/token'

scopes:

nimsas-mc: Access to the Nimsas\_MediaControl API.

schemas:

# STRUCTURED DATA TYPES

MediaInstructionData:

description: The media instruction data for a specific session.

type: object

required:

- sessionId

- mediaInstructionSet

properties:

sessionId:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/SessionId'

mediaInstructionSet:

description: >

Contains a set of instructions for each media flow to control. The mediaId

attribute within the MediaInstructions data type is the key of the map.

type: object

additionalProperties:

$ref: '#/components/schemas/MediaInstructions'

minProperties: 1

MediaInstructions:

description: The media instructions for a specific media flow.

type: object

required:

- mediaId

- mediaResourceType

properties:

mediaId:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/MediaId'

mediaResourceType:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/MediaResourceType'

mediaInstruction:

$ref: '#/components/schemas/MediaInstruction'

dcMediaSpecification:

$ref: '#/components/schemas/DcMediaSpecification'

arMediaSpecification:

$ref: '#/components/schemas/ArMediaSpecification'

mediaProcessingUrl:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/Uri'

audioVideoReNegotiationInd:

$ref: '#/components/schemas/AudioVideoReNegotiationInd'

DcMediaSpecification:

description: The data channel media specification information.

type: object

required:

- streams

- mediaProxyConfig

properties:

mediaProxyConfig:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/MediaProxy'

replaceHttpUrls:

description: >

Contains a list of replacement HTTP URLs. The streamId

attribute within the ReplaceHttpUrl data type is the key of the map.

type: object

additionalProperties:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/ReplaceHttpUrl'

minProperties: 1

mdc1EndpointDcsf:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/MdcEndpoint'

mdc1EndpointMf:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/MdcEndpoint'

mdc2EndpointInfo:

$ref: '#/components/schemas/Mdc2EndpointInfo'

streams:

description: >

Contains a data channel mapping and configuration information. The streamId

attribute within the DcStream data type is the key of the map.

type: object

additionalProperties:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/DcStream'

minProperties: 1

bdcUsedBy:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/BdcUsedBy'

appBindingInfo:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/AppBindingInfo'

ArMediaSpecification:

description: The AR media specification information.

type: object

required:

- mediaProcessingSpec

properties:

mediaProcessingSpec:

type: string

Mdc2EndpointInfo:

description: Represents the negotiated MDC2 media endpoint information.

type: object

properties:

mdc2Protocol:

$ref: 'TS29176\_Nmf\_MRM.yaml#/components/schemas/Mdc2Protocol'

mdc2EndpointDcAs:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/MdcEndpoint'

mdc2EndpointMf:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/MdcEndpoint'

AudioVideoReNegotiationInd:

description: The audio/video media re-negotiation indication information.

type: object

required:

- reNegotiation

properties:

reNegotiation:

type: boolean

mediaConnSide:

$ref: '#/components/schemas/MediaConnSide'

# ENUMS

MediaInstruction:

description: The media instruction for handling a specific media flow.

anyOf:

- type: string

enum:

- TERMINATE\_MEDIA

- ORIGINATE\_MEDIA

- TERMINATE\_AND\_ORIGINATE\_MEDIA

- UPDATE\_MEDIA

- DELETE\_MEDIA

- REJECT\_MEDIA

- type: string

description: >

This string provides forward-compatibility with future extensions to the enumeration

and is not used to encode content defined in the present version of this API.

MediaConnSide:

description: The audio or video media connection side.

anyOf:

- type: string

enum:

- BOTH

- type: string

description: >

This string provides forward-compatibility with future extensions to the enumeration

and is not used to encode content defined in the present version of this API.

Annex B (informative):  
Change history

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Change history** | | | | | | | |
| **Date** | **Meeting** | **TDoc** | **CR** | **Rev** | **Cat** | **Subject/Comment** | **New version** |
| 2023-04 | CT4#115e | C4-231514 |  |  |  | TS skeleton | 0.0.0 |
| 2023-04 | CT4#115e | C4-231551 |  |  |  | Inclusion of pCRs agreed at CT4#115e, including C4-231214, C4-231217, C4-231224, C4-231225, C4-231514, C4-231515, C4-231516, C4-231517. | 0.1.0 |
| 2023-05 | CT4#116 | C4-232553 |  |  |  | Inclusion of pCRs agreed at CT4#116, including C4-232142, C4-232145, C4-232504, C4-232505. | 0.2.0 |
| 2023-09 | CT4#117 | C4-233810 |  |  |  | Inclusion of pCRs agreed at CT4#117, including C4-233250,C4-233254,C4-233255,C4-233256,C4-233700,C4-233701,C4-233704,C4-233856. | 0.3.0 |
| 2023-10 | CT4#118 | C4-234583 |  |  |  | Inclusion of pCRs agreed at CT4#118, including C4-234172, C4-234174, C4-234175, C4-234177, C4-234545, C4-234547. | 0.4.0 |
| 2023-12 | CT4#119 | C4-235660 |  |  |  | Inclusion of pCRs agreed at CT4#119, including C4-235221, C4-235464, C4-235465, C4-235466, C4-235467, C4-235468, C4-235471 | 0.5.0 |
| 2023-12 | CT#102 | CP-233022 |  |  |  | TS presented for information | 1.0.0 |
| 2024-03 | CT4#121 | C4-240852 |  |  |  | Inclusion of pCRs agreed at CT4#121, including C4-240405, C4-240406, C4-240407, C4-240409, C4-240410, C4-240470, C4-240704, C4-240705, C4-240708, C4-240716 | 1.1.0 |
| 2024-03 | CT#103 | CP-240025 |  |  |  | TS presented for approval | 2.0.0 |
| 2024-03 | CT#103 |  |  |  |  | TS approved in CT#103 | 18.0.0 |
| 2024-06 | CT#104 | CP-241044 | 0001 |  | F | Support of SDP attributes a=3gpp-bdc-used-by and a=3gpp-req-app | 18.1.0 |
| 2024-06 | CT#104 | CP-241044 | 0002 | 1 | F | Correction on the Nimsas\_MediaControl service | 18.1.0 |
| 2024-06 | CT#104 | CP-241044 | 0003 | 1 | F | Correction on the Nimsas\_SessionEventControl service | 18.1.0 |
| 2024-06 | CT#104 | CP-241044 | 0004 |  | F | Correction on the Nimsas\_MediaControl OpenAPI | 18.1.0 |
| 2024-06 | CT#104 | CP-241044 | 0006 | 1 | F | Update the presence condition and cardinality for mediaProxyConfig | 18.1.0 |
| 2024-06 | CT#104 | CP-241044 | 0007 | 1 | F | Update the DcMediaSpecification for Nimsas\_SessionEventControl Service API | 18.1.0 |
| 2024-06 | CT#104 | CP-241044 | 0008 | 1 | F | Update the DcMediaSpecification Datatype for MdcEndpoint | 18.1.0 |
| 2024-06 | CT#104 | CP-241044 | 0009 | 1 | F | Update the SessionEventNotificationUri for session event notification | 18.1.0 |
| 2024-06 | CT#104 | CP-241044 | 0010 | 1 | F | Report the media HOLD to DCSF | 18.1.0 |
| 2024-06 | CT#104 | CP-241044 | 0011 | 2 | F | Add the Media re-negotiation indication to Nimsas\_MediaControl API to support AR | 18.1.0 |
| 2024-06 | CT#104 |  | 0012 |  | F | 29.175 Rel-18 API version and External doc update | 18.1.0 |
| 2024-09 | CT#105 | CP-242048 | 0013 |  | F | Corrections and Updates to Notify Service Operation | 18.2.0 |
| 2024-09 | CT#105 | CP-242048 | 0014 |  | F | Remove the SecuritySetup in clause 6.1.6.1 | 18.2.0 |
| 2024-09 | CT#105 | CP-242048 | 0015 | 1 | F | Update the sessionInfo description for Nimsas\_SessionEventControl API | 18.2.0 |