Developer's Guide

AudioCodes One Voice™ Operations Center

AudioCodes Routing Manager (ARM) REST Application Programming Interface (API)

Version 9.2

audiocodes

Table of Contents

Notice

Information contained in this document is believed to be accurate and reliable at the time of printing. However, due to ongoing product improvements and revisions, AudioCodes cannot guarantee accuracy of printed material after the Date Published nor can it accept responsibility for errors or omissions. Updates to this document can be downloaded from https://www.audiocodes.com/library/technical-documents.

This document is subject to change without notice.

Date Published: 05-04-2021

WEEE EU Directive

Pursuant to the WEEE EU Directive, electronic and electrical waste must not be disposed of with unsorted waste. Please contact your local recycling authority for disposal of this product.

Customer Support

Customer technical support and services are provided by AudioCodes or by an authorized AudioCodes Service Partner. For more information on how to buy technical support for AudioCodes products and for contact information, please visit our website at https://www.audiocodes.com/services-support/maintenance-and-support.

Stay in the Loop with AudioCodes

Abbreviations and Terminology

Each abbreviation, unless widely used, is spelled out in full when first used.

Related Documentation

Document Name
ARM User's Manual

Document Revision Record

LTRT	Description
41960	Initial document release for Version 7.2.
41961	Update of template with new AudioCodes logo.
41962	Updates for ARM Versions 7.4, 7.6, 7.8, 8.0, 8.2, 8.4 and 8.6
41963	Updates for ARM Versions 8.8, 9.0 and 9.2
41964	Updates to the Identity API for retrieving user items.

Documentation Feedback

AudioCodes continually strives to produce high quality documentation. If you have any comments (suggestions or errors) regarding this document, please fill out the Documentation Feedback form on our Web site at https://online.audiocodes.com/documentation-feedback.

1 REST API Overview

Why Restful API: Distributed systems over the WEB are widely spread both by the enterprise and core applications, and are the fundamental method used in the emerging Cloud Services.

WEB API is the main interface being used by such WEB services for communication between the elements distributed over the WEB, because it complies with the infrastructure and constraints defined by the WEB.

What is REST: REST is an architectural style and recommendations for WEB API, and not a strict standard and therefore it allows a high level of flexibility in implementation.

A Restful API is designed in a way that will accommodate to WEB infrastructure and fit the Web constraints.

REST adopts a fixed set of operations on named resources, where the representation of each resource is the same for retrieving and setting information. In other words, you can retrieve (read) data in a formatted method and also send data back to the server in similar format in order to set (write) changes to the system.

Operations on resources are implemented with the standard primitives (request methods) of HTTP:

- GET – get information about list of resources or existing resource
- POST – create new resource
- PUT –update existing resource
- DELETE – delete existing resource

JSON Encoding: REST doesn't define which data encoding and formatting is to be used. The most commonly used methods are XML and JSON (JavaScript Object Notation).

Security: basic authentication will be used for each Rest request.

Version Control: REST API should be versioned (v1, v2 …) so client-server communication and API methods can be verified as the same version. Backward compatibility between versions should be kept.

This page is intentionally left blank.

2 ARM server REST API

The following shows the prefix URL for all rest requests:

```
<ARM_Configurator_IP>/ARM/v1
```

Where <ARM_Configurator_IP is the IP address of the ARM Configurator.

This page is intentionally left blank.

3 Topology Rest API

This chapter describes the REST API resources for the Topology server. The following APIs are described:

- Create Group
- Update Group
- Get API
- Create Topology API

3.1.1 Create Group

The `<ARM_Configurator_IP>/ARM/v1/topology/group` URL when used with the POST method, provides the ability for the ARM Manager to send a request to the ARM Configurator to create a new group.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/topology/group
```

HTTP Method

`POST`

Supported Content-Type

`application/json`

Attributes

Attribute	Type	Description
name	String	Group name: unique
elements	[Integer]	-
type	TopologyGroupType	TopologyGroupType: ▪ NODE ▪ PEER_CONNECTION ▪ VOIP_PEER

HTTP Response

- `200 OK`
- Topology Update API

3.1.2 Update Group

The `<ARM_Configurator_IP>/ARM/v1/topology/group/{id}` URL when used with the PUT method provides the ability for the ARM Manager to send a request to the ARM Configurator to update a specific Group interface.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/topology/group/{id}
```

HTTP Method

`PUT`

Supported Content-Type

`application/json`

Attributes

It MUST contain the following arguments:

Attribute	Type	Description
name	String	Group name: unique
elements	[Integer]	-

3.1.3 Get API

The `<ARM_Configurator_IP>/ARM/v1/topology/<ARM_attribute>` URL when used with the GET method provides the ability for the ARM Manager to send a GET Topology request to the ARM Configurator.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/topology/<ARM_attribute>
<ARM_Configurator_IP>/ARM/v1/topology/<ARM_attribute>/{id}
```

Where:

- <ARM_attribute> requests all of the ids of the specified attribute (listed in the table below).
- {id} requests a specific id of a specific attribute (listed in the table below).

HTTP Method

`GET`

Supported Content-Type

`application/json`

Attributes

The ARM Configurator must respond to the GET request with JSON data that includes the request URI attribute. If the id is not specified in the GET request, the URI JSON data MUST contain an array of all specified elements defined in the database for the attribute.

Attribute	Type	Description

node	ARRAY OF:
	[
	{
	id: integer,
	operState: {
	description:String
	status:integer
	}
	adminState: string
	name: string,
	ipAddr: string,
	hostname: String
	serialNum: string,
	deviceType: string
	url: string
	transportType:String
	isIpProfileAvailable: Boolean
	routingServerPolicy:String
	productType:String
	routingServers:[
	{
	id:Integer,
	priority:Integer,
	status:String
	}
	]
	operatorId: Integer
	credentialsId: Integer
	routingServerGroupId: Integer
	supportedFeatures: ENUM
	}
	]
		id- Node id in the database
		
		operState:
		• 0 - login
		• 4 - logout
		• 5 - unroutable
		• 6 - invalid_configuration
		
		adminState:
		• 0- unlock
		• 1- lock
		
		name- Name configured by ARM operator
		
		deviceType:
		▪ 1=MEDIANT_1000
		▪ 2=MEDIANT_3000
		▪ 3=M3K_8410
		▪ 4=MEDIANT_600
		▪ 5=MP500_MSBG
		▪ 6=MEDIANT_1000_MSBG
		▪ 7=MEDIANT_800_PRO
		▪ 8=MEDIANT_1000_PRO
		▪ 9=MEDIANT_4000_ESBC
		▪ 10=SW_SBC:int
		▪ 11=MEDIANT_500_MSBG
		▪ 12=MEDIANT_500_ESBC
		▪ 13=MEDIANT_2600_ESBC
		▪ 14=MEDIANT_500L_MSBR
		▪ 15=MEDIANT_500L_ESBC
		▪ 16=MEDIANTG_800B_MSBR
		▪ 17=MEDIANTG_800B_ESBC
		▪ 18=SW_SE_SBC
		▪ 19=SW_SE_H_SBC
		▪ 20=SW_VE_SBC
		▪ 21=SW_VE_H_SBC
		▪ 22=SW_9000_SBC
		▪ 23=ADC device- default device type,

Attribute	Type	Description
		URI – element rest path
		transportType: • HTTP • HTTPS
		routingServerPolicy: ▪ ROUND_ROBIN ▪ STICKY_PRIMARY ▪ STICKY_LAST_AVAILABLE
		productType : SBC, GW, HYBRID, UNDEFINED or THIRD_PARTY OperatorId – the id of the operator that will be used by the node to login into ARM CredentialsId – the id of the credentials that the ARM will use when authenticating to the node

peerCon	ARRAY OF:
	[
	{
	Id: integer
	nodeId: integer,
	nodeName: string,
	name: string,
	voipPeerId: integer
	operState: string
	adminState: string
	voipPeer: integer,
	routingInterface: integer,
	url: string
	
	}
	]
		▪ operState:
		✓ 0 - created
		✓ 1 - available,
		✓ 2 - not_available ,
		✓ 3 - predeleted,
		✓ 4- temp_not_available,
		✓ 5- not_synced
		▪ adminState:
		✓ 0- unlock
		✓ 1- lock
		✓ 2- locked Incoming
		✓ 3- locked outgoing
		▪ url **–** element rest path

voippeer	ARRAY OF: [{ Id: integer name: string, deviceType: string,	▪ deviceType: ✓ 0-NA ✓ 1-PSTN ✓ 2-SIP_Trunk ✓ 3-PBX

Attribute	Type	Description
	url: string }]	✓ 4-IP_PBX ✓ 5-IP_Phones ▪ url – element rest path

connection	ARRAY OF:
	[
	{
	id: integer
	weight: integer
	name: string,
	URI: string,
	adminState: string
	connections:
	[
	{
	srcNodeId: integer
	destNodeId: integer
	srcRoutingInterface: integer
	destRoutingInterface: integer
	operState: string
	quality: {
	“MOS”: {
	measurement: Integer
	isUsed:Boolean
	},
	“ASR”: {
	measurement: Integer
	isUsed:Boolean
	}
	}
	}
	result: String(enum)
	
	}
	]
	}
		▪ By default each connection is two-way. Each direction has the following statuses:
		✓ Direction 1 –from Node1 to Node2
		✓ Direction2- from Node2 to Node1
		▪ weight [1-100] Default =50
		▪ operState:
		✓ 1- available,
		✓ 2- not_available ,
		✓ 3- predeleted,
		✓ 4- temp_not_available ,
		✓ 5- not_synced
		• adminState:
		✓ 0- unlock
		✓ 1- lock
		▪ url – element rest path
		▪ measurment – the reading that was received from the node
		▪ isUsed: indicated whether the QoS attribute is taken into consideration when calculating the quality result the result of the quality calculation - enum BAD, FAIR, GOOD or UNKNOWN
		This value is sent for both connection nodes and the connection itself.
		▪ type: REGUALR, LOOP_BACK or NODE_TO_THIRD_PARTY

Attribute	Type	Description

	quality: {
	“MOS”: {
	isUsed:Boolean
	},
	“ASR”: {
	isUsed:Boolean
	}
	}
	}
	result: String(enum)
	type:String(enum)
	]	

routingInterface	ARRAY OF:
	[
	{
	Id: integer,
	nodeId: integer
	udpPort: integer,
	tcpPort: integer,
	tlsPort: integer,
	name:string,
	controlIpAddr: string,
	transportType: integer,
	operState: string,
	srdId: string,
	url: string
	}
	]
		• operState:
		✓ 1- available,
		✓ 2- not_available ,
		✓ 3- predeleted,
		✓ 4- temp_not_available ,
		✓ 5- not_synced
		▪ transportType:
		✓ UDP
		✓ TCP
		✓ TLS
		▪ url – element rest path

nodeArchive	ARRAY OF: [{ Id: integer, serialId : String userName: String password: String region: String url: string	

Attribute	Type	Description
	}]	

cluster	{
	id: String
	x: Integer
	y: Integer
	nodes : [Strings]
	name: String
	}	

group	{
	id: Integer
	name: String
	elements: [Integer]
	type: TopologyGroupType
	}
		TopologyGroupType:
		• NODE
		• PEER_CONNECTION
		• VOIP_PEER

registeredUsers	{
	user: String
	host: String
	pConIds: [Integer]
	nodeId: Integer
	}	

HTTP Response

- 200 OK
- 409 Conflict
- 500 Internal error

3.1.4 Filter

You can use a filter for specific requests. The filter should be added to the suffix of the URL, for example:

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/topology/peerCon?filter=(nodeId=100)
```

HTTP Method

GET

Supported Content-Type

application/json

Supported Filter

Filter	Available for these Requests	Description
nodeId	peerCon routingInterface	Filter results only for the node id specified.
name	node	Contains for partial search Equals for regular search
serial	node	Filter by serial number
ipAddrr	node	Filter nodes by ip address

HTTP Response

- 200 OK
- 409 Conflict
- 500 Internal error

3.2 Create Topology API

This section describes the API for creating Topology.

3.2.1 Create Connection

The `<ARM_Configurator_IP>/ARM/v1/topology/connection` URL when used with the POST method, provides the ability for the ARM Manager to send a Create Connection request to the ARM Configurator.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/topology/connection
```

HTTP Method

`POST`

Supported Content-Type

`application/json`

Attributes

Attribute	Type	Description
name	String	Unique name of the connection.
weight	integer	The weight of the connection.

connections	Array of two entries, one for each direction:
	[
	{
	srcNodeId: integer
	destNodeId: integer
	srcRoutingInterface: integer
	destRoutingInterface: integer
	}
	]	

type	String	REGULAR or NODE_TO_THIRD_PARTY

HTTP Response

- `201 Created`
- `409 Conflict`
- `500 Internal Error`

3.2.2 Create Node Archive

The `<ARM_Configurator_IP>/ARM/v1/topology/nodeArchive` URL when used with the POST method, provides the ability for the ARM Manager to send a Create Node Archive request to the ARM Configurator.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/topology/nodeArchive
```

HTTP Method

`POST`

Supported Content-Type

`application/json`

Attributes

Attribute	Type	Description

nodeArchive	Array of
	[
	{
	serialId : String
	userName: String
	password: String
	region: String
	}
	]
		The node details.

HTTP Response

- `201 Created`
- `409 Conflict`
- `500 Internal Error`

3.2.3 Create Virtual Peer Connection

The `<ARM_Configurator_IP>/ARM/v1/topology/peerCon` URL when used with the POST method, provides the ability for the ARM Manager to send a Create Virtual Peer Connection request to the ARM Configurator.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/topology/peerCon
```

HTTP Method

`POST`

Supported Content-Type

`application/json`

Attributes

It MUST contain the following arguments:

Attribute	Type	Description
name	String	The name of the virtual peer connection.
nodeId	Integer	The id of the Virtual Peer node
voipPeerId	Integer	The id of the voipPeerId node (optional)
weight	integer	The weight of the virtual peer node.

HTTP Response

- `201 Created`
- `409 Conflict`
- `500 Internal Error`

3.2.1 Create Routing Interface

The `<ARM_Configurator_IP>/ARM/v1/topology/routingInterface` URL when used with the POST method, provides the ability for the ARM Manager to send a Create Routing Interface request to the ARM Configurator.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/topology/routingInterface
```

HTTP Method

`POST`

Supported Content-Type

`application/json`

Attributes

It MUST contain the following arguments:

Attribute	Type	Description
name	String	The name of the routing interface.
controlIpAddr	String	The IP address of the routing interface (SIP interface In the SBC).

Attribute	Type	Description
tcpPort	Integer	The TCP port number (0 for none)
udpPort	integer	The UDP port number (0 for none)
tlsPort	integer	The TLS port number (0 for none)

The combination of IP address and the 3 ports must be unique.

HTTP Response

- `201 Created`
- `409 Conflict`
- `500 Internal Error`

3.2.2 Create VOIP Peer

The `<ARM_Configurator_IP>/ARM/v1/topology/voippeer` URL when used with the POST method, provides the ability for the ARM Manager to send a Create VOIP Peer request to the ARM Configurator.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/topology/voippeer
```

HTTP Method

`POST`

Supported Content-Type

`application/json`

Attributes

Request must contain the following data elements:

Attribute	Type	Description
deviceType	string	0=NA 1= PSTN 2= SIP_trunk 3= PBX 4= IP_PBX 5= IP_PHONES
name	string	Name of the VOIP peer

HTTP Response:

- 201 Created
- 409 Conflict
- 500 Internal Error

3.2.3 Add Node to ARM

The <ARM_Configurator_IP>/ARM/v1/topology/node URL when used with the POST method, provides the ability for the ARM Manager to send a request to the ARM Configurator to add a node.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/topology/node
```

HTTP Method

POST

Supported Content-Type

application/json

Attributes

Attribute	Type	Description
ipAddr	String	The IP address / hostname of the node
userName	String	Optional
operatorId	Integer	Credentials that the node uses to communicate with ARM
credentialsId	Integer	Credentials used to communicate with the device.
name	String	Unique node name
transportType	String	HTTP or HTTPS
type	String	If Third-party then a 3rd party node will be created.
routingInterfaces	Routing interface Json	Optional
x	Integer	The value of the x coordinate.
y	Integer	The value of the y coordinate.

HTTP Response:

- 201 Created

- 409 Conflict
- 500 Internal Error

3.2.4 Create Group

The <ARM_Configurator_IP>/ARM/v1/topology/group URL when used with the POST method, provides the ability for the ARM Manager to send a request to the ARM Configurator to create a new group.

REST Resource

<ARM_Configurator_IP>/ARM/v1/topology/group

HTTP Method

POST

Supported Content-Type

application/json

Attributes

Attribute	Type	Description
name	String	Group name: unique
elements	[Integer]	
type	TopologyGroupType	TopogyGroupType: • NODE • PEER_CONNECTION • VOIP_PEER

HTTP Response

- 200 OK

3.3 Topology Update API

This section describes the Topology Update API.

3.3.1 Update Connection

The <ARM_Configurator_IP>/ARM/v1/topology/connection/{id} URL when used with the PUT method, provides the ability for the ARM Manager to send a request to the ARM Configurator to update a specific connection.

REST Resource

<ARM_Configurator_IP>/ARM/v1/topology/connection/{id}

Where {id} is the id of the connection.

HTTP Method

`PUT`

Supported Content-Type

`application/json`

Attributes

Attribute	Type	Description
name	string	Name of the connection
weight	integer	Connection weight. 1-100 Default - 50

HTTP Response

- `200 OK`
- `409 Conflict`
- `500 Internal Error`

3.3.2 Update VoIP Peer

The `<ARM_Configurator_IP>/ARM/v1/topology/voippeer/{id}` URL when used with the PUT method, provides the ability for the ARM Manager to send a request to the ARM Configurator to update a specific VoIP Peer.

REST Resource

```
PUT <ARM_Configurator_IP>/ARM/v1/topology/voippeer/{id}
```

Where {id} is the id of the VoIP Peer.

HTTP Method

`PUT`

Supported Content-Type

`application/json`

Attributes

Attribute	Type	Description
deviceType	string	• 0=NA • 1= PSTN • 2= SIP trunk • 3= PBX • 4= IP_PBX • 5= IP_PHONES

Attribute	Type	Description
name	string	Name of the VoIP Peer

HTTP Response

- 200 OK
- 409 Conflict
- 500 Internal Error

3.3.3 Update VoIP Peers Map Locations

The `<ARM_Configurator_IP>/ARM/v1/topology/voippeer/location` URL when used with the PUT method, provides the ability for the ARM Manager to send a request to the ARM Configurator to save the location of the VoIP peers elements.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/topology/voippeer/location
```

HTTP Method

PUT

Supported Content-Type

application/json

Attributes

Attribute	Type	Description
id	Integer	-
x	Integer	X coordinate value
y	Integer	Y coordinate value

HTTP Response

- 200 OK
- 409 Conflict
- 500 Internal Error

3.3.4 Update Peer Connection

The `<ARM_Configurator_IP>/ARM/v1/topology/peerCon/{id}` URL when used with the PUT method, provides the ability for the ARM Manager to send a request to the ARM Configurator to update a peer connection.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/topology/peerCon/{id}
```

Where {id} is the id of the peer connection.

HTTP Method

`PUT`

Supported Content-Type

`application/json`

Attributes

Attribute	Type	Description
name	string	Name of the peer connection.
voipPeerId	Integer	Id of the peer connection.

quality: {
“MOS”: {
 	
isUsed:Boolean
},
“ASR”: {
 	
isUsed:Boolean
}
}
		isUsed: indicated whether the QoS parameter will be taken into consideration when calculating the quality, if this is NULL for all the quality parameters the default global settings will be used, otherwise local will be used

HTTP Response

- `200 OK`
- `409 Conflict`
- `500 Internal Error`

3.3.5 Update Node

The `<ARM_Configurator_IP>/ARM/v1/topology/node/{id}` URL when used with the PUT method, provides the ability for the ARM Manager to send a request to the ARM Configurator to update a specific node.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/topology/node/{id}
```

HTTP Method

`PUT`

Supported Content-Type

`application/json`

Attributes

Attribute	Type	Description
name	string	Name of the node.
transportType	string	Either HTTP or HTTPS
routingServerPolicy	String	ROUND_ROBIN, STICKY_PRIMARY, STICKY_LAST_AVALIABLE
useGeneralCredentials	Boolean	Either node should use general credentials
userName	String	userName for connecting to Node

routingServers:[
{
id:Integer,
priority:Integer,
status:String
}
]
	-	▪ Id – the id of the routing server
		▪ Priority – the order in which the routering server are written to the node (asc order)
		▪ Status – AVAILABLE, UNAVAILABLE

HTTP Response

- `200 OK`
- `409 Conflict`
- `500 Internal Error`

3.3.6 Update Nodes Map Locations

The `<ARM_Configurator_IP>/ARM/v1/topology/node/location` URL when used with the PUT method, provides the ability for the ARM Manager to send a request to the ARM Configurator to save the location of the node elements. In this event, the ARM Web interface will send a JSON object of all the nodes in the screen with their updated x and y coordinates.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/topology/node/location
```

HTTP Method

`PUT`

Supported Content-Type

`application/json`

Attributes

Attribute	Type	Description
id	Integer	Id of the map locations.
x	Integer	X coordinate
y	Integer	Y coordinate

HTTP Response

- `200 OK`
- `409 Conflict`
- `500 Internal Error`

3.3.7 Update Routing Interface

The `<ARM_Configurator_IP>/ARM/v1/topology/routingInterface/{id}` URL when used with the PUT method provides the ability for the ARM Manager to send a request to the ARM Configurator to update a routing interface.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/topology/routingInterface/{id}
```

HTTP Method

`PUT`

Supported Content-Type

`application/json`

Attributes

It MUST contain the following arguments:

Attribute	Type	Description
name	String	Name of the routing interface.
controlIpAddr	String	The IP address of the routing interface (SIP interface on the SBC).
tcpPort	Integer	TCP port name (0 for none)
udpPort	integer	UDP port name (0 for none)
tlsPort	integer	TLS port name (0 for none)

The combination of IP address and the 3 ports must be unique.

HTTP Response

- `200 OK`
- `409 Conflict`
- `500 Internal Error`

3.3.8 Update Clusters

The `<ARM_Configurator_IP>/ARM/v1/topology/cluster` URL when used with the PUT method provides the ability for the ARM Manager to send a request to the ARM Configurator to update a Clusters interface.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/topology/cluster
```

HTTP Method

`PUT`

Supported Content-Type

`application/json`

Attributes

It MUST contain the following arguments:

Attribute	Type	Description
id	String	Internal Id
x	Integer	X coordinate of the cluster
y	Integer	y coordinate of the cluster
nodes	[String]	List of UI internal node ids

Attribute	Type	Description
name	String	The name of the cluster

HTTP Response

- `200 OK`
- `409 Conflict`
- `500 Internal Error`

3.3.9 Update Group

The `<ARM_Configurator_IP>/ARM/v1/topology/group/{id}` URL when used with the PUT method provides the ability for the ARM Manager to send a request to the ARM Configurator to update a specific Group interface.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/topology/group/{id}
```

HTTP Method

`PUT`

Supported Content-Type

`application/json`

Attributes

It MUST contain the following arguments:

Attribute	Type	Description
name	String	Group name: unique
elements	[Integer]	

3.4 Topology Delete API

Note the following:

- The delete peer connection is only available when it is in locked status.
- The Peer connection can get to lock status after it is been deleted from the node itself.
- Delete node is only available if the node is locked.
- For multiple delete operations, multi response JSON is returned with the appropriate HTTP response code for each element.

URL

```
<ARM_Configurator_IP>/ARM/v1/topology/peerCon/{id}
<ARM_Configurator_IP>/ARM/v1/topology/peerCon?filter=(id=id1;id2…)
<ARM_Configurator_IP>/ARM/v1/topology/connection/{id}
<ARM_Configurator_IP>/ARM/v1/topology/connection?filter=(id=id1;id
2…)
<ARM_Configurator_IP>/ARM/v1/topology/voippeer/{id}
<ARM_Configurator_IP>/ARM/v1/topology/voippeer?filter=(id=id1;id2…
)
<ARM_Configurator_IP>/ARM/v1/topology/node/{id}
<ARM_Configurator_IP>/ARM/v1/topology/node?filter=(id=id1;id2…)
<ARM_Configurator_IP>/ARM/v1/topology/routingInterface/{id}
<ARM_Configurator_IP>/ARM/v1/topology/group/{id}
```

- Note that delete peer connection is only available when it is in locked status. Peer connection can enter lock status after it has been deleted from the node itself. Peer connection can enter lock status after it has been deleted from the node itself. The delete action will only delete it from the database.
- Delete virtual peer connection is available even if it is not in locked status.
- Delete node is only available if node is locked
- For multiple delete operation Multi response JSON is returned with the appropriate HTTP response code for each element.
- Deleting routing interface will only work on routing interfaces associated with 3rd party devices

Where:

For the suffix of each URL for update API, there is a {id} sign which represents the id of the element in the database.

HTTP Method

```
DELETE
```

Supported Content-Type

```
application/json
```

HTTP Response

- 200 OK
- 409 Conflict
- 500 Internal Error

3.5 Topology Actions API

3.5.1 Sync all Nodes Action

The <ARM_Configurator_IP>/ARM/v1/topology/sync URL when used with the GET method, provides the ability for the ARM Manager to send a request to the ARM Configurator to Sync all Nodes. This action forces the ARM server to read all configuration from all nodes in the system and update these values to the database. During this process, the server will push (if necessary) information such as connection details and servers details to a node.

Note the following:

- If a filter is used only nodes with thee specified ids are synchronized.
- This action is an unsynchronized action which returns a unique job id which the client may later ask for the action status with the assistance of the 7action status API (see Chapter 7).

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/topology/sync
<ARM_Configurator_IP>/ARM/v1/topology/sync?filter=(id=id1;id2…)
```

HTTP Method

GET

Supported Content-Type

application/json

HTTP Response

- 202- If action has been accepted.
- 409- In case a previous Sync All Nodes action is in progress on the server; the server will not start a new action if a previous action is in progress.

3.5.2 Sync Node Action

The `<ARM_Configurator_IP>/ARM/v1/topology/node/{id}/sync` when used with the GET method, provides the ability for the ARM Manager to send a request to the ARM Configurator to synchronize a specific node.

REST Resource

`<ARM_Configurator_IP>/ARM/v1/topology/node/{id}/sync`

Where {id} is the database id of the node. In this event, the ARM Configurator will perform synchronization between the current state of device and the node data in the database.

HTTP Method

`GET`

Supported Content-Type

`application/json`

HTTP Response

`200 OK`

3.5.3 Lock Node Action

The `<ARM_Configurator_IP>/ARM/v1/topology/node/{id}/lock` URL when used with the PUT method, provides the ability for the ARM Manager to send a request to the ARM Configurator to lock a node.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/topology/node/{id}/lock
<ARM_Configurator_IP>/ARM/v1/topology/node/lock?filter=(id=id1;id2
)
```

Where {id} is the database id of the node.

For multiple lock operation Multi response, JSON is returned with the appropriate HTTP response code for each element.

HTTP Method

`PUT`

Supported Content-Type

`application/json`

Attributes

Lock node request API MUST contain JSON data with the following argument:

Attribute	Type	Description
lock	string	1=lock 0=unlock

HTTP Response

- `200 OK`

For multiple lock operation Multi response, the following additional codes may be returned:

- `409 Conflict` (in case the operation is not allowed)
- `500 Internal Error`

3.5.4 Lock Peer Connection Action

The `ARM_Configurator_IP>/ARM/v1/topology/peerCon/{id}/lock` URL when used with the PUT method, provides the ability for the ARM Manager to send a request to the ARM Configurator to lock a Peer Connection.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/topology/peerCon/{id}/lock
<ARM_Configurator_IP>/ARM/v1/topology/peerCon/lock?filter=(id=id1;
id2…)
```

{id} – database id of the peer connection

For multiple lock operation Multi response, JSON is returned with the appropriate HTTP response code for each element.

HTTP Method

`PUT`

Supported Content-Type

`application/json`

Attributes

Lock node request API MUST contain JSON data with the following argument:

Attribute	Type	Description
lock	string	• 3=locked outgoing • 2=locked incoming • 1=lock • 0=unlock

HTTP Response

- `200 OK`
- `409 Conflict`
- `500 Internal Error`

3.5.5 Lock Router Action

The `<ARM_Configurator_IP>/ARM/v1/topology/router/{id}/lock` URL when used with the PUT method provides the ability for the ARM Manager to send a request to the ARM Configurator to lock a router.

REST Resource

```
PUT <ARM_Configurator_IP>/ARM/v1/topology/router/{id}/lock
```

{id} – database id of the router

Attributes

The Lock router request API MUST contain JSON data of the following argument:

Attribute	Type	Description
lock	string	1=lock 0=unlock

HTTP Response

- `200 OK`
- `409 Conflict`
- `500 Internal Error`

3.5.6 Create Star Action

The `<ARM_Configurator_IP>/ARM/v1/topology/connection/createStar` URL when used with the POST method, provides the ability for the ARM Manager to send the Topology Builder to the ARM Configurator to build connections from the star central routing interface to all other routing interfaces of all nodes.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/topology/connection/createStar
```

HTTP Method

`POST`

Supported Content-Type

`application/json`

Attributes

Attribute	Type	Description
nodeId	integer	Id of the node of the center of the star.
routingInterface	Integer	Identification of Routing Interface dedicated to ARM network unique per Node.
nodeIds	Array	List of integers of ids of nodes to connect to the node.

This action is an unsynchronized action and will return a unique job Id which the client may later ask for the action status with the assistance of the action status API.

HTTP Response

- `202 Accepted`
- `409 Conflict`
- `500 Internal Error`

3.5.7 Create Mesh Action

The `<ARM_Configurator_IP>/ARM/v1/topology/connection/createMesh` URL when used with the POST method, provides the ability for the ARM Manager to send a request to the ARM Configurator to create a Mesh topology.

The ARM Configurator should build a connection between all Routing Interfaces in the ARM network. Each Routing interface should be connected to all other Routing Interfaces.

In case of mesh topology, the ARM Configurator builds topology where each routing interface will be connected to all Routing Interfaces in other nodes.

- The Mesh –automatic builder action should be enabled on the Node level and on the ARM level.
- The Node Level mesh builder allows connecting the selected Node's Routing Interface to mesh.

This action is an unsynchronized action which returns a unique job id which the client may late ask for the action status with the help of the Action Status API (see Chapter 7).

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/topology/connection/createMesh
```

HTTP Method

```
POST
```

Supported Content-Type

```
application/json
```

HTTP Response

- 202 Accepted
- 409 Conflict
- 500 Internal Error

3.5.8 Detach Peer Connection Action

The `<ARM_Configurator_IP>/ARM/v1/topology/peerCon/{id}/detach` URL when used with the PUT method, provides the ability for the ARM Manager to send a request to the ARM Configurator to detach a Peer Connection.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/topology/peerCon/{id}/detach
```

{id} – database id of the peer connection

HTTP Method

```
PUT
```

Attributes

Lock node request API MUST contain JSON data of the following argument:

Attribute	Type	Description
name	string	The name of the created VP, must be unique

HTTP Response

- 200 OK
- 409 Conflict
- 500 Internal Error

3.5.9 Configure Node

REST Resource

```
GET <ARM_Configurator_IP>/ARM/v1/topology/node/{id}/configure
```

{id} – database id of the node

If single sign on operation is supported returns url with token (logged in to node), otherwise returns the url of the node (node's login page).

HTTP Method

GET

HTTP Response

- 200 OK
- 409 Conflict
- 500 Internal Error

This page is intentionally left blank.

4 Routing API

This chapter describes the REST API resources for the ARM routing. The following APIs are described:

- Get API
- Routing Create API
- Routing Update API
- Routing Delete API
- Test Route API
- Time-based Routing (TBR) API
- Attribute Group API

4.1 Get API

The `<ARM_Configurator_IP>/ARM/v1/routing/<ARM_attribute> URL` when used with the GET method, provides the ability for the ARM Manager to send a request to the ARM Configurator to retrieve a route.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/routing/<ARM_attribute>
<ARM_Configurator_IP>/ARM/v1/routing/<ARM_attribute>/{id}
```

Where:

- {id} requests a specific id of a specific attribute (listed in the table below). In case of the Routing Rule attribute, requests for all Routing Rules will only be supported at the Group level.
- <ARM_attribute> requests all of the ids of the specified attribute (listed in the table below).

HTTP Method

`GET`

Supported Content-Type

`application/json`

Attributes

The ARM Configurator must respond to the Get request with JSON data that includes the request URI attribute. If the id is not specified in the GET request, URI JSON data MUST contain an array of all specified elements defined in the database for the attribute.

The following attributes MUST be included:

Attribute	Type	Description
routingRule	{ id: Integer name: String	Enums: RoutingRuleAttrType: PREFIX(0), • HOST_NAME(1)

Attribute	Type	Description

	adminState: adminState
	operState : rrOperState
	URL: String
	priority: Integer
	rrGroupId: Integer
	isFiltered: Boolean
	
	sourceNodes:
	[
	id : Integer
	]
	sourcePcons:
	[
	id: Integer
	]
	
	sourceRRAttr:
	{
	attributes: [
	{
	type: RoutingRuleAttrType (enum)
	value: String
	}
	],
	attributeGroups:
	[
	Integer
	]
	}
	destRRAttr:
	{
	attributes: [
	{
	type: RoutingRuleAttrType (enum)
	value: String
	}
	],
	attributeGroups:
	[
	Integer
	]
	}
	sourceUserGroups:
	[
		• adminState
		✓ LOCKED (1)
		✓ UNLOCKED (0)
		
		▪ rrOperState
		✓ UNDEFINED (0)
		✓ AVAILABLE (1)
		✓ UNAVAILABLE (2)
		• srcNormalizationGroupId / destNormalizationGroupId (Optional) – The normalization Group that will be performed on the action.
		▪ sipReason - Defines the SIP error code that will be presented in the SIP message when the message will be discarded.
		The value should be between 400-699 or null in case the SIP reason is taken from error code of the SIP message.
		▪ Discard - Defines whether this action should discard the call. If this field is set to true, this action must be with the lowest priority and there must not be another action with the same priority (load balancing).
		▪ isFilterd – indicates whether the group or one of its rules matches the search filter.

Attribute	Type	Description

	id: Integer
	]
	dstUserGroup:
	[
	id: Integer
	]
	actions:
	[
	{
	priority : Integer
	nodeId: Integer
	peerConnectionId: Integer
	voipPeerId: Integer
	srcNormalizationGroupId: Integer
	destNormalizationGroupId: Integer
	weight:Integer
	viaNodes:
	[
	Id: Integer
	]
	discard: Boolean
	sipReason: Integer
	
	}
	]
	}	

routingGroup	ARRAY OF:
	[
	{
	id: integer,
	URL: string
	name: String
	adminState: adminState
	priority : Integer
	isFiltered: Boolean
	routingRules:
	[
	{
	routingRule (in above structure)
	}
	]
	}
	]
		• adminState
		✓ LOCKED (1)
		✓ UNLOCKED (0)

sipResponseAction	ARRAY OF: [	

Attribute	Type	Description

	{
	id: int
	sipResponseNumber: int
	sipResponseDescription: String
	reroute: Boolean
	}
	]	

HTTP Response

- 200 OK
- 204 No Content Response

4.1.1 Get Filter

You can use a filter for specific requests. Filters should be added to the suffix of the URL, for example:

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/routing/routingRule?filter=(adminState=1)
<ARM_Configurator_IP>/ARM/v1/routing/routingGroup?filter=(quality=2)
```

HTTP Method

GET

Supported Content-Type

application/json

Attributes

Filter	Available for These Requests	Description
adminState	routingRule	Node id=<value>
testMode	routingRule	Connection id=<value>
quality	routingRule	User Group ID=<value>
isDiscard	routingRule	Equals (=)
rrName	routingRule	Contains (~)
rrGroupName	routingRule	Contains (~)
isRegisteredUsers	routingRule	Equals (=)
isContinueWithNodeTables	routingRule	Equals (=)
requestType	routingRule	Equals (=)

searchString	routingGroup	Contains (~) – searches in most of the fields of the group and the rules inside the group (node name, peer connection name, voippeer name, prefix, host etc.)

HTTP Response

- `200 OK` – if at least one resource is found.
- `204 No Content Response`

4.2 Routing Create API

This section describes the Routing Create API.

4.2.1 Routing Group Create API

The `ARM_Configurator_IP>/ARM/v1/routing/routingGroup` URL when used with the POST method, provides the ability for the ARM Manager to send a request to the ARM Configurator to create a Routing Group.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/routing/routingGroup
```

HTTP Method

`POST`

Supported Content-Type

`application/json`

Attributes

The request MUST consist of the following arguments:

Attribute	Type	Description
name	String	The name of the Routing Group (must be unique).
adminState	adminState	adminState: ▪ LOCKED (1) ▪ UNLOCKED (0)
prevRRGroupId	Integer	prevRRGroupId indicates the previous routing group prevRRGroupId = -1 if first in the list
nextRRGroupId	Integer	nextRRGroupId indicates the next routing group nextRRGroupId = -1 if last in the list

tbrConditionsIds	List of Integers	The ids of the condition to attach to the Routing Rule.

HTTP Response

- `201 Created` – The ARM Configurator responds to the request if completed successfully and the corresponding group is sent.

  In case there was recalculation of all priorities in the database (the ARM user has changed the order of the Routing Groups), the Created group is returned with -1 priority value.
- `409 Conflict` – in case the operation is not allowed.
- `500 Internal Error`

4.2.2 Routing Rule Create API

The `<ARM_Configurator_IP>/ARM/v1/routing/routingRule` URL when used with the POST method, provides the ability for the ARM Manager to send a request to the ARM Configurator to create a routing rule.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/routing/routingRule
```

HTTP Method

`POST`

Supported Content-Type

`application/json`

Attributes

The request MUST consist of the following arguments:

Attribute	Type	Description
name	String	The name of the Routing Rule (must be unique).
adminState	adminState	adminState LOCKED (1), UNLOCKED (0);
prevRRId	Integer	prevRRId stands for previous routing rule prevRRId = -1 if first in the list
nextRRId	Integer	nextRRId stands for next routing rule nextRRId = -1 if last in the list
rrGroupId	Integer	Routing rule group id
sourceNodes	[id : Integer]	Id of the source node (Optional)

Attribute	Type	Description
sourcePcons	[id: Integer]	Id of the source Pcons (Optional)

sourceRRAttr	{
	attributes: [
	{
	type: RoutingRuleAttrType (enum)
	value: String
	}
	],
	attributeGroups:
	[
	Integer
	]
	}
		Optional

destRRAttr	{
	attributes: [
	{
	type: RoutingRuleAttrType (enum)
	value: String
	}
	],
	attributeGroups:
	[
	Integer
	]
	}
		Optional

sourceUserGroups	[id: Integer]	Optional
dstUserGroup	[id: Integer]	Optional
actions	[{ nodeId: Integer	* priority will be used also for load balancing * At least one action must be present * peerConnectionId and viaNodes are Optional

Attribute	Type	Description
	priority: Integer peerConnectionId:Integer voipPeerId:Integer srcNormalizationGroupId: Integer destNormalizationGroupId: Integer weight:Integer isEquallyBalanced:Boolean routingAttempts:Integer sipReason:Integer discard: Boolean viaNodes: [Id: Integer] discard: Boolean sipReason: Integer }]	*must contain nodeId or voipPeerid * peer connection can appear only with node * weight – used when there are actions with the same priority, default 50 srcNormalizationGroupId / destNormalizationGroupId (Optional) – The normalization Group that will be done on the action isEquallyBalanced - whether all the actions in the same priority group should be load balanced, if should have the same value for all of the actions in the priority group. If its true the weight is ignored. • sipReason - Defines the SIP error code that will be presented in the SIP message when the message will be discarded The value should be between 400-699 Or null incase the sip reason is taken from error code of the SIP message • Discard - Defines whether this action should discard the call or not, if this field is set to true, the action must be with the lowest priority and there must not be another action with the same priority (load balancing).
sipReason	Integer	Defines the SIP error code that is displayed in the SIP message when a call is disconnected (when 'discard' attribute is true) • The value should be between 400-699. • 0 in case 'discard' attribute is disabled i.e. this attribute is only relevant when discard attribute is 'true'. When it is true then 'sipReason' is mandatory.
discard	Boolean	Defines whether the call should be disconnected.

Attribute	Type	Description
tbrConditionsIds	List of Integers	The ids of the condition to attach to the Routing Rule Group.
isEmergency	Boolean	Indicates if the rule is prioritized
isTrigger3xx	Boolean	-
isTriggerRefer	Boolean	-
isTriggerInitialOnly	Boolean	-
isTriggerBrokenConnection	Boolean	-
isTriggerFaxRerouting	Boolean	-
privacy	ENUM	Defines which privacy strategy rule to use: TRANSPARENT,TRANSPARENT_WITH_PRIVACY_ID, ANONYMOUS_CALLER, IDENTIFY_CALLER
eventOnMatch	Boolean	-

sipHeaders	{
	name:String
	value:String
	}
		Match specific headers, name is the SIP header, value is the value to match

sourceTopologyGroups	[Integer]	Indicates the source Topology Groups
isRegisteredUsers	Boolean	Enables the Registered Users feature for operators to route calls to SBC registered users
minSecureLogix	Integer	Used for Security Based Routing. Defines the minimum Security call score option for the 'SecureLogix' web service. The Routing Rule uses the score returned from the 'SecureLogix' web service as part of the match. Note: this service must be enabled in the ARM GUI “Security call score” for this setting to take effect. In addition, both this attribute and maxSecureLogix below must be configured, otherwise an error is returned.
maxSecureLogix	Integer	Used for Security Based Routing. Defines the minimum Security call score option for the 'SecureLogix' web service. The Routing Rule uses the score returned from the 'SecureLogix' web service as part of the match. Note: this service must be enabled in the ARM GUI “Security call score” for this setting to take effect. In addition, both this attribute and minSecureLogix above must be configured, otherwise an error is returned.

Attribute	Type	Description

tags	[RoutingRuleTag]	RoutingRuleTag:
		{
		id: Integer,
		tag: ActionCallSetupFields,
		value: String,
		rrId: Integer
		}
		
		ActionCallSetupFields:
		SOURCE_URI_USER(0),
		SOURCE_URI_HOST(1),
		DEST_URI_USER(2),
		DEST_URI_HOST(3),
		DEST_IP_ADDR(4),
		DEST_PORT(5),
		DEST_PROTOCOL(6),
		USER_CREDENTIALS_USER_NAME(7),
		USER_CREDENTIALS_PASSWORD(8),
		P_ASSERTED_IDENTITY_DISPLAY_NAME(9),
		P_ASSERTED_IDENTITY_USER(10),
		P_ASSERTED_IDENTITY_HOST(11),
		TAG_1(12),
		TAG_2(13),
		TAG_3(14),
		CREDENTIALS(15),
		DISCARD_CREDENTIALS(16)

requestType	ENUM	CALL (0), REGISTER (1)

HTTP Response

- `201 Created`
- `409 Conflict`
- `500 Internal Error`

4.2.3 SIP Response Action Create

The URL `<ARM_Configurator_IP>/ARM/v1/routing/sipResponseAction` when used with the POST method provides the ability for the ARM Manager to send a request to the ARM Configurator to create a SIP Response action.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/routing/sipResponseAction
```

HTTP Method

`POST`

Supported Content-Type

`application/json`

Attributes

The request MUST consist of the following arguments:

Attribute	Type	Description
sipResponseNumber	Int	The response number must be unique.
sipResponseDescription	String	The text description of the SIP response.
reroute	Boolean	Mandatory

HTTP Response

- `201 Created`
- `409 Conflict`
- `500 Internal Error`

4.3 Routing Update API

This section describes the Routing Update API.

4.3.1 Routing Rule Update API

The `<ARM_Configurator_IP>/ARM/v1/routing/routingRule/{id}` URL when used with the PUT method provides the ability for the ARM Manager to send a request to the ARM Configurator to update a routing rule.

REST Resource

`<ARM_Configurator_IP>/ARM/v1/routing/routingRule/{id}`

Where {id} is the id of the element in the database.

HTTP Method

`PUT`

Supported Content-Type

`application/json`

Attributes

The request should contain the following elements:

Attribute	Type	Description
name	String	The name of the Routing Rule (must be unique for the specific Routing Rule group).
adminState	adminState	adminState: ▪ LOCKED (1), ▪ UNLOCKED (0);
prevRRId	Integer	prevRRId is the previous routing rule prevRRId = -1 if first in the list. Optional – if not sent, nextRRId will also not be checked.
nextRRId	Integer	nextRRId is the next routing rule nextRRId = -1 if last in the list. Optional
sourceNodes	[id : Integer]	The id of the source node.
sourcePcons	[id: Integer]	The id of the source Pcons.

sourceRRAttr	{
	attributes: [
	{
	type: RoutingRuleAttrType (enum)
	value: String
	}
	],
	attributeGroups:
	[
	Integer
	]
	}
		The source Routing Rule attributes.

destRRAttr	{
	attributes: [
	{
	type: RoutingRuleAttrType (enum)
	value: String
	}
	],
	attributeGroups:
	[
	Integer
		The destination Routing Rule attributes.

Attribute	Type	Description
	] }	
sourceUserGroups	[id: Integer]	The id of the source User Groups.
dstUserGroups	[id: Integer]	The id of the destination User Group

actions	[
	{
	nodeId: Integer
	priority: Integer
	peerConnectionId: Integer
	voipPeerId: Integer
	srcNormalizationGroupId: Integer
	destNormalizationGroupId: Integer
	weight:Integer
	isEquallyBalanced:Boolean
	routingAttempts:Integer
	sipReason:Integer
	discard: Boolean
	viaNodes:
	[
	Id: Integer
	]
	}
	]
	topologyGroupId: Integer
	attemptsInTopologyGroup: Integer
	registeredUsers: Boolean
	continueWithNodeInternalTables: Boolean
	destTypeRequestUri: Boolean
		Mandatory (at least one)
		
		srcNormalizationGroupId / destNormalizationGroupId (Optional) – The normalization Group that will be applied to the action
		
		* must contain nodeId or voipPeerid
		* peer connection can appear only with node

Attribute	Type	Description
	srcNormalizationFromHeader: Boolean srcNormalizationPAIHeader: Boolean srcNormalizationPPIHeader: Boolean }]	
discard	Boolean	Defines whether the call should be disconnected.
sipReason	Integer ▪ The value should be between 400-699 • 0 in case discard is disabled	Defines the SIP error code that is displayed in the SIP message when a call is disconnected (when 'discard' attribute is true) • The value should be between 400-699. ▪ 0 in case 'discard' attribute is disabled i.e. this attribute is only relevant when discard attribute is 'true'. When it is true then 'sipReason' is mandatory.
tbrConditionsIds	List of Integers	The ids of the condition to attach to the Routing Rule Group.
quality	Enum	The quality of the route that will be chosen if quality based routing is enabled ▪ FAIR ▪ GOOD
isTrigger3xx	Boolean	-
isTriggerRefer	Boolean	-
isTriggerInitialOnly	Boolean	-
isTriggerBrokenConnection	Boolean	-
isTriggerFaxRerouting	Boolean	-
privacy	ENUM	Defines which privacy strategy should be used if the rule is matched: TRANSPARENT,TRANSPARENT_WITH_PRIVACY_ID, ANONYMOUS_CALLER, IDENTIFY_CALLER
eventOnMatch	Boolean	-

sipHeaders	{
	name:String
	value:String
	}
		Match specific headers, name is the SIP header, value is the value to match

sourceTopologyGroups	[Integer]	The Topology Group ID of the Routing rule.

Attribute	Type	Description
isRegisteredUsers	Boolean	Enables the Registered Users feature for operators to route calls to SBC registered users.
minSecureLogix	Integer	Used for Security Based Routing. Defines the minimum Security call score option for the 'SecureLogix' web service. The Routing Rule uses the score returned from the 'SecureLogix' web service as part of the match. Note: this service must be enabled in the ARM GUI “Security call score” for this setting to take effect. In addition, both this attribute and maxSecureLogix below must be configured, otherwise an error is returned.
maxSecureLogix	Integer	Used for Security Based Routing. Defines the maximum Security call score option only if the 'SecureLogix' web service is used. Once enabled, the Routing Rule will use the score returned from the 'SecureLogix' web service as part of the match. Note: this service must be enabled in the ARM GUI “Security call score” for this setting to take effect. In addition, both this attribute and minSecureLogix above must be configured, otherwise an error is returned.

tags	[RoutingRuleTag]	RoutingRuleTag:
		{
		id: Integer,
		tag: ActionCallSetupFields,
		value: String,
		rrId: Integer
		}

requestType	ENUM	CALL (0), REGISTER (1)

HTTP Response

- `200 OK`
- `409 Conflict` (in case the operation is not allowed)
- `500 Internal Server Error`

4.3.2 Routing Group Update API

The `<ARM_Configurator_IP>/ARM/v1/routing/routingGroup/{id}` URL when used with the PUT method provides the ability for the ARM Manager to send a request to the ARM Configurator to update a Routing Group.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/routing/routingGroup/{id}
```

Where {id} is the id of the element in the database.

HTTP Method

```
PUT
```

Supported Content-Type

```
application/json
```

Attributes

The request should contain the following elements:

Attribute	Type	Description
adminState	adminState	The adminState of the routing group: LOCKED (1), UNLOCKED (0);
name	String	The name of the routing group. This name must be unique.
prevRRId	Integer	prevRRId indicates the previous routing rule prevRRId = -1 if first Note: if not sent, this implies that the location of the rule has not changed and therefore nextRRId will also not be checked. If a new rule is added to table, -1
nextRRId	Integer	nextRRId indicates the next routing rule nextRRId = -1 if last
tbrConditionsIds	List of Integers	The ids of the condition to attach to the Routing Rule Group.

Response

Corresponding Routing Rule Group will be sent. In case there was recalculation of all priorities for the rules in the Database (the ARM User changed the order of the Routing Groups) the Created group will be returned with -1 priority value.

4.3.3 SIP Response Action Update API

The `<ARM_Configurator_IP>/ARM/v1/routing/sipResponseAction/{id}` URL when used with the PUT method, it provides the ability for the ARM Manager to send a request to the ARM Configurator to update a SIP response action.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/routing/sipResponseAction/{id}
```

Where {id} is the id of the element in the database.

HTTP Method

`PUT`

Supported Content-Type

`application/json`

Attributes

The request should contain the following elements:

Attribute	Type	Description
sipResponseNumber	Int	The SIP Response Number (must be unique).
sipResponseDescription	String	The description of the SIP response.
reroute	Boolean	Mandatory

HTTP Response

- `200 OK`
- `409 Conflict` – in case the operation was not allowed.
- `500 Internal Error`

4.4 Routing Delete API

You can send a request to the ARM Configurator to delete a routing rule, a routing group or a SIP Response action.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/routing/routingRule/{id}
<ARM_Configurator_IP>/ARM/v1/routing/routingGroup/{id}
<ARM_Configurator_IP>/ARM/v1/routing/sipResponseAction/{id}
```

Where {id} is the id of the element in the database.

HTTP Method

`DELETE`

Supported Content-Type

`application/json`

HTTP Response

- `200 OK`

- `409 Conflict` (in case the operation is not allowed)
- `500 Internal Server Error`

4.5 Test Route API

The `<ARM_Configurator_IP>/ARM/v1/routing/testRoute` URL when used with the POST method provides the ability for the ARM Manager to send a request to the ARM Configurator to update a test route.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/routing/testRoute
```

HTTP Method

`POST`

Supported Content-Type

`application/json`

Request Attributes

The request should contain the following elements:

Attribute	Type	Description
srcPeerConnection	Integer	The id of the peer connection.
srcNode	Integer	The id of the srcNode.
maxRoutes	Integer	The maximum number of routes to find. When this attribute is not contained in the request, the default value 6 applies.
maxRoutesPerDestination	Integer	The maximum number of routes per destination to find. When this attribute is not contained in the request, the default value 2 applies.
srcUser	string	The name of the user of the call source.
srcHost	string	The name of the source host (optional).
destUser	string	The name of the user of the call destination.
destHost	string	The name of the host of the call destination (optional).
routerId	Integer	(Optional) The id of the router that will respond to the route request
isTrigger3xx	Boolean	Mark the test route request as if it was a 3XX response
isTriggerRefer	Boolean	Mark the test route request as if it was a referred call.

Attribute	Type	Description
isTriggerInitialOnly	Boolean	-
isTriggerBrokenConnection	Boolean	-
isTriggerFaxRerouting	Boolean	-

headers	ARRAY OF:
	[
	{
	name: TestRouteRequestSipHeaderEnum,
	value: STRING
	}
	]
		• Name – for now only "Contact" will be supported. It should be a unique name in this array (optional). name is enum (current with only one value "Contact").
		• Value – as a real Contact header.
		For example : <sip:401@10.7.2.17> or <sip:401@10.7.2.17:5060> or <sip:401@moshebe.audiocodes.com> or <sip:mosheCell@moshebe.audiocodes.com:6061>
		• GUI validations :
		The header should be a unique name in this array. It can be added from IPGroup Pcon (not TrunkGroup). The Name and value are not null or empty string.

Response Attributes

The response should contain the following attributes:

Attribute	Type	Description
srcNode	Integer	The source node id.
srcPeerConnection	Integer	The peer connection id.
httpResponse	Integer	The HTTP response code.
description	String	The description of the routing manipulation list.

preRouteManipulationsList	ARRAY OF:[{
	originalValue:String
	manipulatedValue:String
	manipulatedEntity:String
	isChanged:Boolean
	manipulationsList:[{
	originalValue:String
	manipulatedValue:String
	manipulatingEntity:String
	manipulatingEntityUrl: String
	manipulatingEntityName:String
	usingManipulatingEntity:String
	usingManipulatingEntityUrl: String
	usingManipulatingEntityName:String
	description:String
	isChanged:Boolean
	}]
	}]
		• ManipulatingEntity (optional)- the entity from ARM data base/GUI perspective which tracked and might change the ManipulatedEntity.
		• ManipulatedEntity (optional)- the entity that is tracked and may change.
		• originalValue: the value of the manipulated entity prior to manipulation (in the case of user/password/destIP), these entities may be optional as well.
		• Note: When the ManipulatingEntity is a 'RoutingRuleAction', there is no URL.

paths	ARRAY OF:
	[{
	destNode:Integer
	destPeerConnection:Integer
	routingRuleAction:Integer
	routingRuleID:Integer
	
	edges:[{
	connSrcNode:Integer
	connDestNode:Integer
	connectionId:Integer
	}]
	pathRouteManipulationsList:[{
	originalValue:String
	manipulatedValue:String
	manipulatedEntity:String
		▪ ManipulatingEntity (optional)- the entity from ARM data base/GUI perspective which tracked and might change the ManipulatedEntity.
		▪ ManipulatedEntity (optional)- the entity that is tracked and might change.
		▪ originalValue: the value of the manipulated entity prior to manipulation (in the case of user/password/destIP), these entities may be optional as well.

	isChanged:Boolean manipulationsList:[{ originalValue:String manipulatedValue:String manipulatingEntity:String String manipulatingEntityName:String usingManipulatingEntity:String usingManipulatingEntityUrl: String usingManipulatingEntityName:String description:String isChanged:Boolean }] }]]	▪ Note: when the ManipulatingEntity is a RoutingRuleAction, it contains a RoutingRule URL. ▪ The RoutingRuleActionId is attached per path as well.
discardingByRoutingRule	Integer	The id of the routing rule which discards the call. -1 if there is no such rule.

HTTP Response

- `200 OK`
- `409 Conflict` (in case the operation is not allowed)
- `500 Internal Server Error`

Other Information

- preRouteManipulationsList and pathRouteManipulationsList:
  - ManipulatedEntity means which entity will be tracked and might change.
  - ManipulatedEntity values:
    - SOURCE_URI_USER
    - SOURCE_URI_HOST
    - DEST_URI_USER
    - DEST_URI_HOST
    - DEST_IP_ADDR
    - DEST_PORT
    - DEST_PROTOCOL
    - USER_CREDENTIALS_USER_NAME
    - USER_CREDENTIALS_PASSWORD
  - Only tracked will be added to routeManipulationsList.
  - isChanged will be set to true if one of the isChanged in manipulationsList is true.

- ManipulatingEntity means which entity from ARM data base/GUI perspective tracked and might change the ManipulatedEntity.

- ManipulatingEntity values:
  - ROUTING_INCOMING_MANIPULATION
  - PCON
  - CALL_SETUP_RULE
  - ROUTING_RULE_ACTION
  - NORMANIZATION_GROUP
- usingManipulatingEntity and usingManipulatingEntityId are Optional.

■ preRouteManipulationsList and pathRouteManipulationsList examples:

Example 1: if Pcon 1 with NormalizationGroup 2 and Call Setup Rule 3 and Routing Rule Action 4 with NormalizationGroup 5 changed sourceUriUser :

```
  preRouteManipulationsList:{
                   originalValue:12
                   manipulatedValue:56
                   manipulatedEntity:SOURCE_URI_USER
                   isChanged:true

                   manipulationsList:[{
                                                originalValue:12
                                                manipulatedValue:34
                                                manipulatingEntity:PCON
                                                manipulatingEntityId:1
                                                usingManipulatingEntity:N
ORMANIZATION_GROUP
                                                usingManipulatingEntityId
:2
                                                description:
                                                isChanged:true
                                                },
                                                {
                                                originalValue:34
                                                manipulatedValue:56
                                                manipulatingEntity:CALL_S
ETUP_RULE
                                                manipulatingEntityId:3
                                                description:
                                                isChanged:true
                                                }]

paths =[{
.
.
.               pathRouteManipulationsList:[{
                       afrerPreRouteValue:String
                                               originalValue:56
                                                manipulatedValue:78
                                                manipulatingEntity:ROUTIN
G_RULE_ACTION
                                                manipulatingEntityId:4
                                                usingManipulatingEntity:N
ORMANIZATION_GROUP
                                                usingManipulatingEntityId
:5
```

```
                                                    description:
                                                    isChanged:true
                                                    }
                                                    ]

                                    }]
                    }
```

Example 2 :if RoutingIncomingNormalization (it doesn't have really an index so we set it to "-1") with NormalizationGroup 2 and Call Setup Rule 3 and Routing Rule Action 4 with NormalizationGroup 5 changed sourceUriUser :

```
  preRouteManipulationsList:{
                    originalValue:12
                    manipulatedValue:56
                    manipulatedEntity:SOURCE_URI_USER
                    isChanged:true

                    manipulationsList:[{
                                                    originalValue:12
                                                    manipulatedValue:34
                                                    manipulatingEntity:
ROUTING_INCOMING_MANIPULATION
                                                    manipulatingEntityId:-1
                                                    usingManipulatingEntity:N
ORMANIZATION_GROUP
                                                    usingManipulatingEntityId
:2
                                                    description:
                                                    isChanged:true
                                                    },
                                                    {
                                                    originalValue:34
                                                    manipulatedValue:56
                                                    manipulatingEntity:CALL_S
ETUP_RULE
                                                    manipulatingEntityId:3
                                                    description:
                                                    isChanged:true
                                                   }]

paths =[{
.
.
.               pathRouteManipulationsList:[{
                                                  originalValue:56
                                                   manipulatedValue:78
                                                   manipulatingEntity:ROUTIN
G_RULE_ACTION
                                                   manipulatingEntityId:4
                                                   usingManipulatingEntity:N
ORMANIZATION_GROUP
                                                   usingManipulatingEntityId
:5
```

```
                                                    description:
                                                    isChanged:true
                                                    }
                                                    ]

                                    }]
                    }
```

4.6 Time-based Routing (TBR) API

This section describes the Time-based Routing API.

4.6.1 TBR Conditions Create API

The `<ARM_Configurator_IP>/ARM/v1/routing/condition/tbr` URL when used with the POST method, provides the ability for the ARM Manager to send a request to the ARM Configurator to create conditions for Time-based routing.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/routing/condition/tbr
```

HTTP Method

`POST`

Supported Content-Type

`application/json`

Attributes

The request should contain the following attributes:

Attribute	Type	Description

TimeCondition	{
	“id” : Integer,
	“name” String,
	“days” : [
	“day”: Day
	“startTime”: Time (String),
	“endTime”: Time (String),
	],
	“startDateTime”: DateTime (String),
	“endDateTime”: DateTime (String)
	“startDay”: Day,
	“endDay:” Day,
	“type”: Enum (Integer)
	}
		Enum DAYS:
		SUN, MON, TUE, WED, THU, FRI, SAT
		startTime and endTime are optional in case the type is WEEKLY or PERIOD.
		If they are not sent, the condition will be for a full day
		(not sent equals both of them being null),
		In type DAILY the days should have all the days of the week, and startTime and endTime are must, and must be the same time for all days.
		In type RANGE days should have only two days, the start day with startTime and end day with endTime.
		Example of the time format: 12:12:21.000
		startDateTime and endDateTime will be ignored unless the type is PERIOD

Attribute	Type	Description
		startDay and endDay will be ignored unless type is RANGE startDateTime and endDateTime should be in the same format as defined by Alex, for example: 2015-11-12T14:21:19.000Z Enum Type: DAILY, WEEKLY, PERIOD or RANGE

HTTP Response

- `200 OK` - if no error occurred.
- `409 Conflict` - if an error occurred.

4.6.2 TBR Condition Update API

The `<ARM_Configurator_IP>/ARM/v1/routing/condition/tbr/{id}` URL when used with the PUT method provides the ability for the ARM Manager to send a request to the ARM Configurator to update a TBR condition.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/routing/condition/tbr/{id}
```

HTTP Method

`PUT`

Supported Content-Type

`application/json`

Attributes

Attribute	Type	Description
TimeCondition	{ “id” : Integer, “name” String, “days” : [“day”: Day “startTime”: Time (String), “endTime”: Time (String),	Enum DAYS: SUN, MON, TUE, WED, THU, FRI, SAT startTime and endTime are optional in case the type is WEEKLY or PERIOD.

Attribute	Type	Description
	], “startDateTime”: DateTime (String), “endDateTime”: DateTime (String) “startDay”: Day, “endDay:” Day, “type”: Enum (Integer) }	If they are not sent, the condition will be for a full day (not being sent is equivelant to both of them being set to null). In type DAILY, the days should include all the days of the week, startTime and endTime are mandatory, and must be the same time for all days. In type RANGE, days should be set to only two days, the start day with startTime and end day with endTime. Example of the time format: 12:12:21.000 startDateTime and endDateTime will be ignored unless the type is PERIOD startDay and endDay will be ignored unless type is RANGE. startDateTime and endDateTime should be in the same format as defined by Alex, for example: 2015-11-12T14:21:19.000Z Enum Type: DAILY, WEEKLY, PERIOD or RANGE

HTTP Response

- `200 OK` – if no error occurred
- `409 Conflict` – if an error occurred

4.6.3 TBR Condition Delete API

The `<ARM_Configurator_IP>/ARM/v1/routing/condition/tbr/{id` URL provides the ability for the ARM Manager to send a request to the ARM Configurator to delete a TBR condition.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/routing/condition/tbr/{id}
```

Where id is the id of the element in the database.

HTTP Method

`DELETE`

Supported Content-Type

`application/json`

HTTP Response

- `200 OK` – if no error occurred.
- `409 Conflict` - if an error occurred.

Note: Conditions that are attached to Routing Rule Groups or Routing Rules cannot be deleted.

4.7 Attribute Group API

This section describes the URLs for Attribute Groups.

4.7.1 Attribute Group Get API

The `<ARM_Configurator_IP>/ARM/v1/routing/attributeGroup` URL when used with the GET method, provides the ability for the ARM Manager to send a request to the ARM Configurator to retrieve an Attribute Group.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/routing/attributeGroup
<ARM_Configurator_IP>/ARM/v1/routing/attributeGroup/{id}
```

HTTP Method

`GET`

Supported Content-Type

`application/json`

Attributes

Attribute	Type	Description

attrGroup	{
	“id” : Integer
	“name” String,
	“values” : [
	String
	],
	“type”: ENUM (String),
	}
		Name – must be unique
		type -
		PREFIX(0),
		HOST_NAME(1);

attrGroups	[
	{
	attrGroup
	}
	]
		Name – must be unique
		type -
		PREFIX(0),
		HOST_NAME(1);

HTTP Response

- `200 OK` – if no error occurred.
- `409 Conflict` – if an error occurred.

4.7.2 Attribute Group Create API

The `<ARM_Configurator_IP>/ARM/v1/routing/attributeGroup URL` when used with the PUT method provides the ability for the ARM Manager to send a request to the ARM Configurator to create an Attribute Group.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/routing/attributeGroup
```

HTTP Method

`PUT`

Supported Content-Type

`application/json`

Attributes

Attribute	Type	Description
attrGroup	{ “name” String, “values” : [String	Name – must be unique type - • PREFIX(0), • HOST_NAME(1);

Attribute	Type	Description
	], “type”: ENUM (String), }	

HTTP Response

- `200 OK` – if no error occurred.
- `409 Conflict` – if an error occurred.

4.7.3 Attribute Group Update API

The `<ARM_Configurator_IP>/ARM/v1/routing/attributeGroup/{id}` URL when used with the PUT method provides the ability for the ARM Manager to send a request to the ARM Configurator to update an Attribute Group.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/routing/attributeGroup/{id}
```

HTTP Method

`PUT`

Supported Content-Type

`application/json`

Attribute

Attribute	Type	Description

attrGroup	{
	“name” String,
	“values” : [
	String
	],
	“type”: ENUM (String),
	}
		Name – must be unique
		type -
		PREFIX(0),
		HOST_NAME(1);

HTTP Response

- `200 OK` – if no error occurred.
- `409 Conflict` – if an error occurred.

4.7.4 Attribute Group Delete API

The `<ARM_Configurator_IP>/ARM/v1/routing/attributeGroup/{id}` when used with the DELETE method provides the ability for the ARM Manager to send a request to the ARM Configurator to delete an Attribute Group.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/routing/attributeGroup/{id}
```

HTTP Method

`DELETE`

Supported Content-Type

`application/json`

HTTP Response

- `200 OK` – if no error occurred.
- `409 Conflict` – if an error occurred.

Note: An Attribute Group cannot be deleted when it contains conditions that are attached to Routing Rule Groups or Routing Rules.

4.7.5 Attribute Group Multiple Delete API

The `<ARM_Configurator_IP>/ARM/v1/routing/attributeGroup?filter=(id=1;2;3)` URL when used with the DELETE methods provides the ability for the ARM Manager to send a request to the ARM Configurator to delete multiple Attribute Groups.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/routing/attributeGroup?filter=(id=1;2;3)
```

HTTP Method

`DELETE`

Attributes

Attribute	Type	Description

multiStatusJsonList	[
	{
	status: Integer
	description: String
	id: Integer
	}
	]
		Status: the http response code for this specific id
		Description: the description
		Id: the id of the object for this specific response

HTTP Response

- `200 OK` – if no error occurred
- `409 Conflict`
- `500 Internal Error`

5 Security Rest API

This chapter describes the REST API resources for the ARM security. The following APIs are described:

- Get API
- Security Create API
- Security Update API
- Security Delete API
- Login API

5.1 Get API

The `<ARM_Configurator_IP>/ARM/v1/security/<ARM_attribute>` when used with the GET method, provides the ability for the ARM Manager to send a request to the ARM Configurator to retrieve security attributes.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/security/<ARM_attribute>
<ARM_Configurator_IP>/ARM/v1/security/<ARM_attribute>/{id}
```

Where:

- <ARM_attribute> requests all of the ids of the specified attribute (listed in the table below).
- {id} requests a specific id of a specific attribute (listed in the table below

HTTP Method

`GET`

Supported Content-Type

`application/json`

Attributes

The ARM Configurator must respond to the GET request with JSON data that includes the request URI attribute. If the id is not specified in the GET request, URI JSON data MUST contain an array of all specified elements defined in the database for the attribute.

Attribute	Type	Description

operator	ARRAY OF:
	[
	{
	id: integer,
	userName: string,
	role: OperatorRoleType
	url: string
	}
	]
		OperatorRoleType:
		▪ 1- ADMIN
		▪ 2- SECURITY_ADMIN
		▪ 3-DEVICE
		▪ 4-ROUTER
		
		url – element REST path

Attribute	Type	Description

credential	ARRAY OF:
	[
	{
	Id: Integer,
	username: String
	type: ENUM,
	description: String
	}
	]
		Type:
		• ADMIN (1)
		• SECURITY_ADMIN (2)
		• DEVICE (3)
		• ROUTER(4)
		• MONITOR(5)
		• NONE(100)

HTTP Response

- `200 OK`
- `409 Conflict`
- `500 Internal Error`

5.1.1 Filter

There is a possibility to use a filter for specific requests. Filter should be added to the suffix of the URL, for example:

```
<ARM_Configurator_IP>/ARM/v1/security/<ARM_element_path>?filter=(roles=ENUM)
```

Filter	Available for these Requests	Description
roles	Operator	Filter Operators according to the role
type	Credentials	Filter credentials according to the type

5.2 Security Create API

This section describes the Security Create API.

5.2.1 Create Operator

The `<ARM_Configurator_IP>/ARM/v1/security/operator` URL when used with the POST method, provides the ability for the ARM Manager to send a request to the ARM Configurator to create an operator.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/security/operator
<ARM_Configurator_IP>/ARM/v1/security/credentials
```

HTTP Method

`POST`

Supported Content-Type

`application/json`

Attributes

The request MUST contain JSON data that consists of the following attributes:

Attribute	Type	Description
userName	STRING	The requested user name
password	STRING	The password
role	role: string	role – 1= admin 2= security_admin The security level of the requested user security_admin - Security administrator – can add/delete/modify users, and do all the rest admin - Administrator – can NOT add/delete/modify users; but can do all the rest
description	String	Optional, only relevant for credentials creation

HTTP Response

- `200 OK`
- `201 Created` (successful user creation)
- `403 Forbidden` (incorrect credentials or with role of the user that does not match the security administrator)

5.3 Security Update API

This section describes the Security Update API.

5.3.1 Update Operator

The `<ARM_Configurator_IP>/ARM/v1/security/operator/{id}` URL when used with the PUT method, provides the ability for the ARM Manager to send a request to the ARM Configurator to update an operator.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/security/operator/{id}
<ARM_Configurator_IP>/ARM/v1/credentials/operator/{id}
```

Where {id} id is the id of the element in the database.

HTTP Method

`PUT`

Supported Content-Type

`application/json`

Attributes

Attribute	Type	Description
userName	String	The requested user name
password	String	The user password.
role	role: string	Role: • 1= admin • 2= security_admin The security level of the requested user: • ROLE_ SECURITY_ADMIN- Security administrator – can add/delete/modify users, and do all the rest • ROLE_ ADMIN - Administrator – can NOT add/delete/modify users; but can do all the rest.
description	String	Optional, Only relevant for credentials creation

HTTP Response

- `200 OK`
- `201 Created` (successful user creation)
- `403 Forbidden` (incorrect credentials or with role of the user that does not match the security administrator)

5.4 Security Delete API

This section describes the Security Delete API.

5.4.1 Delete Operator API

The `<ARM_Configurator_IP>/ARM/v1/security/operator/{id}` URL provides the ability for the ARM Manager to send a request to the ARM Configurator to delete an operator.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/security/operator/{id}
<ARM_Configurator_IP>/ARM/v1/security/credentials/{id}
```

Where {id} is the id of the element in the database.

HTTP Method

`DELETE`

HTTP Response

- `200 OK`

5.5 Login API

The `<ARM_Configurator_IP>/ARM/v1/login` URL when used with the POST method, provides the ability for the ARM Manager to send a request to login.to the ARM Configurator.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/login
```

HTTP Method

`POST`

Supported Content-Type

`application/json`

Attributes

The request MUST contain JSON data that consists of the following elements:

Attribute	Type	Description
userName	STRING	The requested user name
password	STRING	The login password

HTTP Response

- `200 OK`- Response will be sent if the user was successfully authenticated.

- `401` – Unauthorized in case of incorrect credentials or if the role of the user didn't match the correct privileges (security admin/admin).

Response Attributes

The response provided by the ARM Configurator must contain JSON data with the following attributes:

Attribute	Type	Description
userName	String	The requested user name (only in case of a successful login).
role	String	The role (permission) of the user (only in case of successful login). For the role the following enum is used: • SECURITY_ADMIN • ADMIN
sessionTimeout	Integer	Session timeout in hours (only in case of a successful login).
inactivityPeriod	Integer	Inactive period in minutes (only in case of successful login).
description	String	Description for a successful login and for failed login.

6 Server Rest API

This chapter describes the REST API resources for the ARM server. The following APIs are described:

- Get API
- Servers Create API
- Server Update API
- Server Delete API

6.1 Get API

The `<ARM_Configurator_IP>/ARM/v1/server` URL when used with the GET method, provides the ability for the ARM Manager to send a request to the ARM Configurator to retrieve an updated list of servers.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/server
<ARM_Configurator_IP>/ARM/v1/server/{id}
```

Where {id} requests a specific id of a specific attribute (listed in the table below).

HTTP Method

`GET`

Supported Content-Type

`application/json`

Attributes

The ARM Configurator must respond to the GET request with JSON data that includes the request URI attribute. If the id is not specified in the GET request, URI JSON data MUST contain an array of all specified elements defined in the database for the attribute.

Attribute	Type	Description
Server	ARRAY OF: [{ Id: integer address: String name: string, type: string, protocol: string, port: integer, url: string state:ServerState confRouterPort:Integer adminState:Integer memory:Integer	Protocol: 1–http 2-https type: ROUTING url – element rest path confRouterPort – conf <-> router ServerState: • AVAILABLE (0), • UNAVAILABLE (1), • INITIALIZING (2), • NOT_SYNC (3); adminState:

Attribute	Type	Description
	nodes:[Integer] }]	▪ LOCKED (1), ▪ UNLOCKED (0); Nodes – the ids of the nodes for which the router is configured.

HTTP Response

- `200 OK`
- `409 Conflict`
- `500 Internal error`

6.2 Servers Create API

This section describes the Servers Create API.

6.2.1 Create Server

The `<ARM_Configurator_IP>/ARM/v1/server` URL when used with the POST method, provides the ability for the ARM Manager to send a request to the ARM Configurator to create a new ARM Routing server.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/server
```

HTTP Method

`POST`

Supported Content-Type

`application/json`

Attributes

Attribute	Type	Description
name	String	The name of the server
address	String	The IP address of the server
type	String	type: ROUTING
protocol	String	The protocol that is used by the server.
port	Integer	The port that is used by the server.
confRouterPort	Integer	The port that is used by the Configurator to communicate with the routers.
nodes	ARRAY OF:	Ids of the router nodes.

Attribute	Type	Description
	[Integer]	

HTTP Response

- `201 Created`
- `409 Conflict`
- `500 Internal error`

6.3 Server Update API

This section describes the Server Update API.

6.3.1 Update Server

The `<ARM_Configurator_IP>/ARM/v1/server/{id}` URL when used with the PUT method, provides the ability for the ARM Manager to send a request to the ARM Configurator to update a server configuration.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/server/{id}
```

Where {id} is the id of the element in the database.

HTTP Method

`PUT`

Supported Content-Type

`application/json`

Attributes

Attribute	Type	Description
address	String	IP address of the server.
protocol	String	The protocol that is used by the server.
port	Integer	The port that is used by the server.
confRouterPort	Integer	The port that is used by the by the configurator to communicate with the routers.

HTTP Response

- `200 OK`
- `409 Conflict`
- `500 Internal error`

6.4 Server Delete API

This section describes the Server Delete API.

6.4.1 Delete Server API

The `<ARM_Configurator_IP>/ARM/v1/server/{id}`URL when used with the DELETE method, provides the ability for the ARM Manager to send a request to the ARM Configurator to delete a specific server.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/server/{id}
```

Where {id} is the id of the element in the database.

Supported Content-Type

```
application/json
```

HTTP Method

```
DELETE
```

HTTP Response

- 200 OK

6.5 Servers Group API

6.5.1 Get All Server Groups

The `<ARM_Configurator_IP>/ARM/v1/server/{id}`URL when used with the GET method, provides the ability for the ARM Manager to request to the ARM Configurator to retrieve a list of all server groups.

REST Resource

```
GET <ARM_Configurator_IP>/ARM/v1/server/group
```

Supported Content-Type

```
application/json
```

HTTP Method

```
GET
```

HTTP Response

- 200 OK
- 409 Conflict
- 500 Internal error

Attributes

Attribute	Type	Description
id	Integer	Server Group ID
name	String	Server Group Name
policyBetweenGroups	ENUM	STICKY_PRIMARY, STICKY_LAST_AVAILABLE
policyInsideGroup	ENUM	ROUND_ROBIN, STICKY_PRIMARY, STICKY_LAST_AVAILABLE

groups	[
	{
	id: Integer
	servers:{
	id: Integer
	priority: Integer
	}
	
	}
	]
		The outer id is the group id
		The id in the servers objects is the server id from the database
		Priority is between 1 - 10

6.5.2 Get Specific Server Group

The `<ARM_Configurator_IP>/ARM/v1/server/{id}` URL when used with the GET method, provides the ability for the ARM Manager to request to the ARM Configurator to retrieve a specific server group.

REST Resource

```
GET <ARM_Configurator_IP>/ARM/v1/server/group/{id}
```

Supported Content-Type

`application/json`

HTTP Method

`GET`

HTTP Response

- `200 OK`
- `409 Conflict`
- `500 Internal error`

Attributes

Attribute	Type	Description
id	Integer	Server Group ID
name	String	Server Group Name
policyBetweenGroups	ENUM	STICKY_PRIMARY, STICKY_LAST_AVAILABLE
policyInsideGroup	ENUM	ROUND_ROBIN, STICKY_PRIMARY, STICKY_LAST_AVAILABLE

groups	[
	{
	id: Integer
	servers:{
	id: Integer
	priority: Integer
	}
	
	}
	]
		The outer id is the group id
		The id in the servers objects is the server id from the database
		Priority is between 1 - 10

6.5.3 Create Server Group API

The `<ARM_Configurator_IP>/ARM/v1/server/group` URL when used with the POST method, provides the ability for the ARM Manager to request to the ARM Configurator to create a specific server group.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/server/group
```

Supported Content-Type

```
application/json
```

HTTP Method

```
POST
```

HTTP Response

- 201 Created
- 409 Conflict
- 500 Internal error

Attribute

Attribute	Type	Description
id	Integer	Server Group ID
name	String	Server Group Name
policyBetweenGroups	ENUM	STICKY_PRIMARY, STICKY_LAST_AVAILABLE
policyInsideGroup	ENUM	ROUND_ROBIN, STICKY_PRIMARY, STICKY_LAST_AVAILABLE

groups	[
	{
	id: Integer
	servers:{
	id: Integer
	priority: Integer
	}
	
	}
	]
		The outer id is the group id
		The id in the servers objects is the server id from the database
		Priority is between 1 - 10

6.5.4 Update Server Group API

The `<ARM_Configurator_IP>/ARM/v1/server/group` URL when used with the PUT method, provides the ability for the ARM Manager to request to the ARM Configurator to update a specific server group.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/server/group/{id}
```

HTTP Method

`PUT`

Attributes

Attribute	Type	Description
id	Integer	Server Group ID
name	String	Server Group Name
policyBetweenGroups	ENUM	STICKY_PRIMARY, STICKY_LAST_AVAILABLE
policyInsideGroup	ENUM	ROUND_ROBIN, STICKY_PRIMARY, STICKY_LAST_AVAILABLE

Attribute	Type	Description

groups	[
	{
	id: Integer
	servers:{
	id: Integer
	priority: Integer
	}
	
	}
	]
		The outer id is the group id
		The id in the servers objects is the server id from the database
		Priority is between 1 - 10

HTTP Response

- 200 OK
- 409 Conflict
- 500 Internal error

6.5.5 Delete Server Group API

The `<ARM_Configurator_IP>/ARM/v1/server/group` URL when used with the DELETE method, provides the ability for the ARM Manager to request to the ARM Configurator to delete a specific server group.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/server/group/{id}
```

HTTP Method

DELETE

Supported Content-Type

application/json

HTTP Response

- 200 OK
- 409 Conflict
- 500 Internal error

7 Action Status API

The `<ARM_Configurator_IP>/ARM/v1/action/status/{id}` URL when used with the GET method, provides the ability for the ARM Manager to send a request to the ARM Configurator to determine the status of an unsynchronized action.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/action/status/{id}
```

Where id is the unique identifier of the unsynchronized action.

HTTP Method

`GET`

Supported Content-Type

`application/json`

HTTP Response

- `202 Accepted` – in case the action is still in progress.
- `200 OK` – in case the action has been completed.

This page is intentionally left blank.

8 Network Planner API

This chapter describes the REST API resources for the Network Planner. The following APIs are described:

- Get API
- Network Planner Create API
- Network Planner Update API
- Network Planner Delete API
- Network Planner Actions API

8.1 Get API

The `<ARM_Configurator_IP>/ARM/v1/planning/<ARM_element_path>` URL when used with the GET method, provides the ability for the ARM Manager to request to the ARM Configurator to retrieve an ARM element.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/planning/<ARM_element_path>
<ARM_Configurator_IP>/ARM/v1/planning/<ARM_element_path>/{id}
```

{id} – id of the requested element

ARM Configurator must respond to the Get request with 200 respond and includes JSON data that depended from specified in request URI element and its id. If the id was not specified in get request URI JSON data must contain array of all specified elements defined in Topology data base.

HTTP Method

`GET`

Supported Content-Type

`application/json`

Attributes

The following attributes MUST be included:

Attribute	Type	Description

map	{
	nodes:
	{
	“id”: {node}
	}
	"routingInterfaces": {
	“id”:{routing interface}
	}
	voipPeers:{
		each object is a map of elements
		hashCode: the hash code of the full map object

Attribute	Type	Description

	“id”:{voipPeer}
	}
	“peerConnections” : {
	“id”:{peer connection}
	}
	"internalPeerConnections": {
	“id”:{peer connection}
	}
	"connections":{
	“id”:{connection}
	}
	hashCode: Integer
	}	

node	ARRAY OF:
	[
	{
	adminState: String
	deviceTypeI: Integer
	deviceTypeS: String
	id: Integer
	ipAddr: String
	isIpProfileAvailable: Boolean
	isOffline: Boolean
	name: String
	operState: String
	password: String
	prevSerialNum: String
	productType: String
	serialNum: String
	
	softwareVersion: String
	transportType: String
	useGeneralCredentials: Boolean
	username: String
	x: Interger
	y: Interger
	}
		id- Node id in the database
		operState -
		1- available
		2- not_available
		3- predeleted
		4- not_synced
		adminState –
		0- unlock
		1- lock
		name- Name configured by ARM operator

Attribute	Type	Description
	]	

peerCon	ARRAY OF:
	[
	{
	Id: integer
	nodeId: integer,
	nodeName: string,
	name: string,
	voipPeerId: integer
	operState: string
	adminState: string
	voipPeer: integer,
	routingInterface: integer,
	url: string
	proxySetWithIp: Boolean
	isOffline: Boolean
	internal: Boolean
	destNormalizationGroupId: integer
	srcNormalizationGroupId: integer
	}
	result: String(enum)
	]
		operState –
		1- available,
		2- not_available ,
		3- predeleted,
		4- temp_not_available ,
		5- not_synced
		
		adminState –
		0- unlock
		1- lock
		url – element rest path
		
		result the result of the quality calculation - enum BAD, FAIR, GOOD or UNKNOWN

voippeer	ARRAY OF:
	[
	{
	Id: integer
	name: string,
	deviceType: string,
	url: string
	isOffline: Boolean
	
	}
	]
		deviceType –
		0-NA
		1-PSTN
		2-SIP_Trunk
		3-PBX
		4-IP_PBX
		5-IP_Phones
		url – element rest path

connection	ARRAY OF: [{ id: integer	By default each connection is two-way Each direction has status Direction 1 –from Node1 to Node2

Attribute	Type	Description

	weight: integer
	name: string,
	URI: string,
	adminState: string
	isOffline: Boolean
	connections:
	[
	{
	offline: Boolean
	srcNodeId: integer
	destNodeId: integer
	srcRoutingInterface: integer
	destRoutingInterface: integer
	operState: string
	}
	result: String(enum)
	}
	]
	
	}
	result: String(enum)
	
	]
		Direction2- from Node2 to Node1
		weight [1-100]
		Default =50
		
		operState –
		1- available,
		2- not_available ,
		3- predeleted,
		4- temp_not_available ,
		5- not_synced
		
		adminState –
		0- unlock
		1- lock
		url – element rest path
		
		result the result of the quality calculation - enum BAD, FAIR, GOOD or UNKNOWN
		
		this will be send for both the of the edges and the connection itself

routingInterface	ARRAY OF: [{ Id: integer, nodeId: integer udpPort: integer, tcpPort: integer, tlsPort: integer, name:string,	operState – 1- available, 2- not_available , 3- predeleted, 4- temp_not_available , 5- not_synced transportType: UDP TCP TLS url – element rest path

Attribute	Type	Description
	controlIpAddr: string, transportType: integer, operState: string, srdId: string, url: string isOffline: Boolean }]	

8.1.1 Filter

There is a possibility to use a filter for specific requests. The filter should be added to the suffix of the URL, for example:

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/planning/peerCon?filter=(nodeId=100)
```

HTTP Method

GET

Supported Content-Type

application/json

Attributes

The following attributes MUST be included:

Filter	Available for these Requests	Description
isOffline	node peerCon voippeer connection	Filter results for elements that are only offline and that weren't imported in.
name	node peerCon voippeer connection	Contains for partial search Equals for regular search

HTTP Response

- `200 OK`
- `500 Internal error`
- `409 Conflict`

8.2 Network Planner Create API

8.2.1 Create Connection

The `<ARM_Configurator_IP>/ARM/v1/planning/connection` URL when used with the POST method, provides the ability for the ARM Manager to request to the ARM Configurator to create a network planner.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/planning/connection
```

HTTP Method

`POST`

Supported Content-Type

`application/json`

Attributes

The following attributes MUST be included:

Attribute	Type	Description
name	String	Unique name of the network planner.
weight	integer	Weight of the network planner

connections	Array of 2 entries, 1 for each direction
	[
	{
	srcNodeId: integer
	destNodeId: integer
	srcRoutingInterface: integer
	destRoutingInterface: integer
	}
	]	

HTTP Response

- `201 Created`

- 500 Internal error
- 409 Conflict

8.2.2 Create PeerConnection

The <ARM_Configurator_IP>/ARM/v1/planning/peerCon URL when used with the POST method, provides the ability for the ARM Manager to request to the ARM Configurator to create a peer connection.

REST Resource

<ARM_Configurator_IP>/ARM/v1/planning/peerCon

HTTP Method

POST

Supported Content-Type

application/json

Attributes

It MUST contain the following arguments:

Attribute	Type	Description
name	String	The name of the peer connection.
nodeId	Integer	The node id of the peer connection.
peerConnectionNodeType	PeerConnectionNodeType	IPGroup or TrunkGroup
voipPeerId	Integer	If null, ARM will create the voip peer
rilId	Integer	If null ARM will pick the SBC dummy routing interface
weight	Integer	The weight assigned to the peer connection.

HTTP Response

- 201 Created
- 500 Internal error
- 409 Conflict

8.2.3 Add Node to ARM

The `<ARM_Configurator_IP>/ARM/v1/planning/node` URL when used with the POST method, provides the ability for the ARM Manager to request to the ARM Configurator to create a planning node.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/planning/node
```

HTTP Method

`POST`

Supported Content-Type

`application/json`

Attributes

It MUST contain the following arguments:

Attribute	Type	Description
ipAddr	String	The IP address of the node
userName	String	Optional – otherwise Admin
password	String	Optional – otherwise Admin
productType	String	One of the following: • SBC • GW • HYBRID
x	Integer	X coordinate value
y	Integer	Y coordinate value

HTTP Response

- `201 Created`
- `500 Internal error`
- `409 Conflict`

8.2.4 Add VoIP Peer to ARM

The `<ARM_Configurator_IP>/ARM/v1/planning/voippeer` URL when used with the POST method, provides the ability for the ARM Manager to request to the ARM Configurator to create a VoIP peer.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/planning/voippeer
```

HTTP Method

`POST`

Supported Content-Type

`application/json`

Attributes

It MUST contain the following arguments:

Attribute	Type	Description
deviceType	String	IP_PHONES, IP_PBX, PSTN, NA, PBX, SIP_TRUNK
name	String	Name of the VP
x	Integer	Value of the x coordinate
y	Integer	Value of the y coordinate

HTTP Response

- `201 Created`
- `500 Internal error`
- `409 Conflict`

8.3 Network Planner Update API

At the suffix of each URL for update API there is a {id} sign which represents the id of the element in the database.

8.3.1 Update Connection

The `<ARM_Configurator_IP>/ARM/v1/planning/connection/{id}` URL when used with the PUT method, provides the ability for the ARM Manager to request to the ARM Configurator to update a specific VoIP connection.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/planning/connection/{id}
```

HTTP Method

`PUT`

Supported Content-Type

`application/json`

Attributes

It MUST contain the following arguments:

Attribute	Type	Description
name	string	The connection name.
weight	integer	The weight assigned to the connection. 1-100 Default - 50
result	String	BAD, FAIR, GOOD, UNKNOWN

HTTP Response

- `200 OK`
- `500 Internal error`
- `409 Conflict`

8.3.2 Update VOIP Peer

The `<ARM_Configurator_IP>/ARM/v1/planning/voippeer/{id}` URL when used with the PUT method, provides the ability for the ARM Manager to request to the ARM Configurator to update a VOIP Peer.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/planning/voippeer/{id}
```

HTTP Method

`PUT`

Supported Content-Type

`application/json`

Attributes

It MUST contain the following arguments:

Attribute	Type	Description
deviceType	string	0=NA 1= PSTN 2= SIP_trunk 3= PBX 4= IP_PBX 5= IP_PHONES
Name	string	Name of the VOIP Peer

HTTP Response

- `200 OK`
- `500 Internal error`
- `409 Conflict`

8.3.3 Update VoIP Peers Map Locations

The `<ARM_Configurator_IP>/ARM/v1/planning/voippeer/location` URL when used with the PUT method, provides the ability for the ARM Manager to request to the ARM Configurator to save the location of the VoIP peers elements.

REST Resource

```
PUT <ARM_Configurator_IP>/ARM/v1/planning/voippeer/location
```

HTTP Method

`PUT`

Supported Content-Type

`application/json`

Attributes

It MUST contain the following arguments:

Attribute	Type	Description
id	Integer	id of the VoIP Peers Map locations.
x	Integer	X coordinate value.
y	Integer	Y coordinate value.

HTTP Response

- `200 OK`
- `500 Internal error`

- 409 Conflict

8.3.4 Update Peer Connection

The <ARM_Configurator_IP>/ARM/v1/planning/peerCon/{id} URL when used with the PUT method, provides the ability for the ARM Manager to request to the ARM Configurator to update a peer connection.

REST Resource

<ARM_Configurator_IP>/ARM/v1/planning/peerCon/{id}

HTTP Method

PUT

Supported Content-Type

application/json

Attributes

The request MUST contain the following data elements::

Attribute	Type	Description
name	string	Name of the Peer connection.
voipPeerId	Integer	ID of the Peer connection.
result	String	BAD, FAIR, GOOD, UNKNOWN

HTTP Response

- 200 OK
- 500 Internal error
- 409 Conflict

8.3.5 Update Node

The `<ARM_Configurator_IP>/ARM/v1/planning/node/{id}` URL when used with the PUT method provides the ability for the ARM Manager to request to the ARM Configurator to update a specific node.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/planning/node/{id}
```

HTTP Method

`PUT`

Supported Content-Type

`application/json`

Attributes

The request MUST contain the following data elements:

Attribute	Type	Description
name	string	-
transportType	string	Either HTTP or HTTPS
useGeneralCredentials	Boolean	Either node should use general credentials
userName	String	userName for connecting to Node
password	String	-
ipAddress	String	-

HTTP Response

- `200 OK`
- `500 Internal error`
- `409 Conflict`

8.3.6 Update Nodes Map Locations

The `<ARM_Configurator_IP>/ARM/v1/planning/node/location` URL when used with the PUT method provides the ability for the ARM Manager to request to the ARM Configurator.to save the location of the node elements.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/planning/node/location
```

HTTP Method

`PUT`

Supported Content-Type

application/json

Attributes

The request MUST contain the following data elements:

Attribute	Type	Description
id	Integer	id of the node location
x	Integer	X coordinate
y	Integer	Y coordinate

HTTP Response

- 200 OK
- 500 Internal error
- 409 Conflict

8.4 Network Planner Delete API

When ARM Manager sends a Delete request to ARM Configurator, it includes a suffix of each URL for update API {id} sign which represents the id of the element in the database.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/planning/peerCon/{id}
<ARM_Configurator_IP>/ARM/v1/planning/connection/{id}
<ARM_Configurator_IP>/ARM/v1/planning/voippeer/{id}
<ARM_Configurator_IP>/ARM/v1/planning/node/{id}
<ARM_Configurator_IP>/ARM/v1/planning/map
```

HTTP Method

DELETE

Supported Content-Type

application/json

Note: The delete operation is only available for elements which are not part of a routing rules, or policy studio. Clearing the map will delete all the offline elements from the database and remove the imported elements.

HTTP Response

- 200 OK
- 500 Internal error
- 409 Conflict

8.5 Network Planner Actions API

8.5.1 Import Full Topology

The `<ARM_Configurator_IP>/ARM/v1/planning/import` URL when used with the POST method provides the ability for the ARM Manager to request to the ARM Configurator.to import all the elements from the regular map view into the offline planner.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/planning/import
```

HTTP Method

`POST`

Supported Content-Type

`application/json`

HTTP Response

- `200 OK`.
- `409` in case a previous sync all nodes action is in progress on the server. The server will not start a new action if a previous one is in progress.

8.5.2 Import Nodes

The `<ARM_Configurator_IP>/ARM/v1/planning/node/{id}/import` URL when used with the POST method provides the ability for the ARM Manager to request to the ARM Configurator.to import specific nodes from the regular map view into the offline planner.

REST Resource

```
POST <ARM_Configurator_IP>/ARM/v1/planning/node/{id}/import
POST <ARM_Configurator_IP>/ARM/v1/planning/node/import
```

{id} – database id of the node

HTTP Method

`POST`

Supported Content-Type

`application/json`

HTTP Response

- `201 Created`
- `500 Internal error`
- `409 Conflict`

8.5.3 Export Node

The `<ARM_Configurator_IP>/ARM/v1/planning/node/{id}/export` URL when used with the POST method provides the ability for the ARM Manager to request to the ARM Configurator to export specific nodes from the offline planner.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/planning/node/{id}/export
```

{id} – database id of the node

HTTP Method

`POST`

Supported Content-Type

`application/json`

HTTP Response

- `202 Accepted`
- `500 Internal error`
- `409 Conflict.`

8.5.4 Lock Node Action

The `<ARM_Configurator_IP>/ARM/v1/planning/node/{id}/lock` URL when used with the POST method provides the ability for the ARM Manager to request to the ARM Configurator to lock a node.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/planning/node/{id}/lock
```

{id} – database id of the node

HTTP Method

`POST`

Supported Content-Type

`application/json`

Note: For multiple lock operation Multi response JSON is returned with the appropriate HTTP response code for each element.

Attributes

The request MUST contain the Lock node request API MUST contain JSON data of the following argument:

Attribute	Type	Description
lock	string	1=lock 0=unlock

HTTP Response

- `201 Created`
- `500 Internal error`
- `409 Conflict`

8.5.5 Lock Peer Connection Action

The `<ARM_Configurator_IP>/ARM/v1/planning/node/{id}/lock` URL when used with the POST method provides the ability for the ARM Manager to request to the ARM Configurator to lock a specific peer connection.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/planning/peerCon/{id}/lock
```

{id} – database id of the peer connection

HTTP Method

`POST`

Supported Content-Type

`application/json`

Note: For multiple lock operation Multi response JSON is returned with the appropriate HTTP response code for each element.

Attributes

Lock node request API MUST contain JSON data of the following argument:

Attribute	Type	Description
lock	string	1=lock 0=unlock

HTTP Response

- `201 Created`
- `500 Internal error`
- `409 Conflict`

9 Identity

This chapter describes the REST API resources for the LDAP server. The following APIs are described:

- Get API
- Identity Create API
- Identity Update API
- Identity Delete API

9.1 Get API

The `<ARM_Configurator_IP>/ARM/v1/identity/<ARM_attribute>` URL when used with the GET method, provides the ability for the ARM Manager to send a request to the ARM Configurator to retrieve specific LDAP attributes for server, userGroup, users and user group.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/identity/<ARM_attribute>
<ARM_Configurator_IP>/ARM/v1/identity/<ARM_attribute>/{id}
```

Where:

- <ARM_attribute> requests all of the ids of the specified attribute (listed in the table below).
- {id} requests a specific id of a specific attribute (listed in the table below).

HTTP Method

`GET`

Supported Content-Type

`application/json`

Attributes

The ARM Configurator must respond to the Get request with JSON data that includes the request URI attribute. If the id is not specified in the GET request, URI JSON data MUST contain an array of all specified elements defined in the database for the attribute.

The following attributes MUST be included:

Attribute	Type	Description
server	[{ id: Integer name: String status: serverStatus host: String port : Integer dn: String	serverStatus • AVAILABLE (0), • UNAVAILABLE (1); syncIntervalMinutes (in minutes)

Attribute	Type	Description

	searchFilter: String
	baseObject: String
	numOfUsers: Integer
	sslEnabled: boolean
	certificate: String
	timeout: Integer
	syncIntervalMinutes: Integer
	LastSyncTimestamp: String
	fullSyncHour: Integer
	fullSyncMinute: Integer
	fullSyncIntervalInDays: Integer
	lastFullSyncTimestamp: String
	dicAttrMapping :
	[
	{
	dicAttrId: Integer
	dicAttrName: String
	aDAttrName: String
	normalizationGroup: Integer
	}
	]
	}
	]
		LastSyncTimeStamp and lastFullSyncTimestamp - UTC formatted string: "yyyy-mm-ddThh:mm:ss.mmmZ"

userGroup	[
	{
	id: Integer
	name: String
	associatedDialable: String
	conditions:
	[
	{
	dicAttrName: String
	operator: LogicalOperatorEnum
	values:
	[
	String
	]
	}
		LogicalOperatorEnum:
		▪ EQUALS (0),
		▪ NOT_EQUALS (1)
		▪ CONTAINS (2)
		▪ NOT_CONTAINS (3)
		▪ associatedDialable: name of dialable attribute

Attribute	Type	Description
	] }]	

user	{
	Id: long
	name: String
	origin: String
	groups:
	[
	groupId: String
	],
	dicAttributes:
	[
	{
	local: boolean
	dicAttrId: Integer
	value: String
	}
	]
	}
		▪ origin: “LOCAL” in case of local user or name of the AD in case of AD user
		▪ dicAttributes are dynamic, according to the dictionary in the system

user/count	Integer	▪ Total number of users
user/export	CSV file	▪ Export Users data to CSV file
Users list	{ cursor: { before: Integer, after: Integer }, users: [{ Id: long name: String origin: String groups: [groupId: String	• origin: “LOCAL” in case of local user or name of the AD in case of AD user. • dicAttributes are dynamic, according to the dictionary in the system cursor: • before: Returns the index of the previous page (where 0 represents page 1). • after: Returns the index of the next page (where 0 represents page 1).

Attribute	Type	Description

	],
	dicAttributes:
	[
	{
	local: boolean
	dicAttrId: Integer
	value: String
	}
	]
	}
	]
	}	

dicAttribute	[
	{
	Id: Integer
	name: String
	description: String
	primary: boolean
	defaultRemoteName: String
	local: boolean
	dialable: boolean
	}
	]
		• defaultRemoteName: Remote Attribute to map dicAttribute to by default
		• local: is attribute created to be “local” or already to be mapped to AD
		• defaultRemoteName: To display this option in the Web interface (not mapped internally)

fileRepository	[{ Id: Integer name: String status: FileRepositoryStatus delimiter: Character isFileHasHeaders: Boolean uniquePropertyId: Integer isNameUnique: Boolean nameFileMapping: String nameFileLocation: Integer numberOfUsers: Integer lastTimestamp: String description: String dicAttrMapping: [dicAttrId: Integer dicAttrName: String	FileRepositoryStatus: • INITIALIZING (0) • ACTIVE (1) • ERROR (2)

Attribute	Type	Description
	fileRepositoryAttrName: String fileRepositoryAttrLocation: Integer normalizationGroupId: Integer] }]	

HTTP Response

- `409 Conflict` – id does not exist.
- `200 OK` – if at least one resource is found.
- `204 No Content Response.`

9.1.1 Get Filter API

The `<ARM_Configurator_IP>/ARM/v1/identity/user?filter=(seachString~Alex,origin=1;2)URL` when used with the GET method provides the ability for the ARM Manager to send a request to the ARM Configurator to filter specific requests. The filter should be added to the suffix of the URL, for example:

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/identity/user?filter=(seachString~Alex,origin=1;2)
```

Supported Content-Type

`application/json`

HTTP Method

`GET`

Attributes

Filter	Available for these Requests	Description
name	user	Filters only the user name column (Contains ~)
limit	user	Limits the number of result items that can be retrieved. For example, 100 items.
before	user	Specifies the page index for which to retrieve items.

Filter	Available for these Requests	Description
after	user	Specifies the page index for which to retrieve items.
userGroup	user	Filters according to the user group id (Equals =)
searchString	user	Filters according to any attribute of the user (Contains ~)
origin	user	Filters according to the LDAP server id, -1 for local, can have multiple values (Equals =)
sort	user	Sorts the results according to the column name that was sent. If "-" was sent in the beginning, descending order will be used. For example &sort=name &sort=-Country Note: if "origin" is used, ARM will sort by the server id and not in alphabetical order.

HTTP Response

- `200 OK` – at least one resource is found.
- `204 No Content Response.`

9.2 Identity Create API

This section describes the Identity Create API for creating an LDAP server for managing LDAP server users.

9.2.1 Server Create API

The <ARM_Configurator_IP>/ARM/v1/identity/server URL when used with the POST method sends a request to the ARM Configurator to create an LDAP server.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/identity/server
```

Supported Content-Type

```
application/json
```

HTTP Method

```
POST
```

Attributes

Attribute	Type	Description
name	String	A unique name for the LDAP server (if the name is not unique, 409 Conflict is returned).
password	String	The password of the LDAP server.
host	String	The name of the LDAP server host.
port	Integer	The LDAP server port.
dn	String	The Distinguished Nam of the LDAP server.
searchFilter	String	The LDAP search filter.
baseObject	String	The LDAP base object.
sslEnabled	boolean	Determines whether SSL is enabled for the connection with the LDAP server.
certificate	String	The certificate that secures the connection with the LDAP server.

Attribute	Type	Description
syncIntervalMinutes	Integer	The interval time (in minutes) when partial synchronization is performed to retrieve users from the LDAP server. This operation retrieves users that have changed since the last partial sync was run.
fullSyncHour	Integer	The hour when full synchronization is performed to retrieve all users from the LDAP server.
fullSyncMinute	Integer	The minute in the hour when full synchronization is performed to retrieve all users from the LDAP server.
timeout	Integer	The timeout for connection with the LDAP server (in minutes).
fullSyncIntervalInDays	Integer	The interval time (in days) when full synchronization is performed to retrieve all users from the LDAP server.

dicAttrMapping	[
	{
	dicAttrId: Integer
	aDAttrName: String
	normalizationGroupId: Integer
	}
	]
		The dictionary attribute mapping with the LDAP server.

HTTP Response

- `201 Created`
- `409 Conflict`
- `500 Internal Error`

9.2.2 Server Test API

The <ARM_Configurator_IP>/ARM/v1/identity/server/test URL when used with the POST method, provides the ability for the ARM Manager to send a request to the ARM Configurator to test an LDAP server.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/identity/server/test
```

HTTP Method

POST

Supported Content-Type

application/json

Request Attributes

Attribute	Type	Description
name	String	A unique name for the LDAP server (if the name is not unique, 409 Conflict is returned).
password	String	The password of the LDAP server.
host	String	The name of the LDAP server host.
port	Integer	The LDAP server port
dn	String	The Distinguished Nam of the LDAP server.
searchFilter	String	The LDAP search filter.
baseObject	String	The LDAP base object.
sslEnabled	boolean	Determines whether SSL is enabled for the connection with the LDAP server.
certificate	String	The certificate that secures the connection with the LDAP server.

Attribute	Type	Description
syncIntervalMinutes	Integer	The interval time (in minutes) when partial synchronization is performed with the LDAP server. This operation retrieves users that have changed since the last partial sync was run.
fullSyncHour	Integer	The hour when full synchronization is performed to retrieve all users from the LDAP server.
fullSyncMinute	Integer	The minute in the hour upon when full synchronization is performed to retrieve all users from the LDAP server.
timeout	Integer	The timeout for connection with the LDAP server (in minutes).
fullSyncIntervalInDays	Integer	The interval time (in days) upon when full synchronization is performed to retrieve users from the LDAP server.

dicAttrMapping	[
	{
	dicAttrId: Integer
	aDAttrName: String
	normalizationGroup: Integer
	}
	]
		The dictionary attribute mapping with the LDAP server.

Response Attributes

The following attributes MUST be included in the response:

Attribute	Type	Description
connectionStatus	String	Status of the connection with the LDAP server.
availableAttributes	[{	Attributes available in this LDAP server that can be used as attribute mappings.

Attribute	Type	Description
	availableAttribute: String }]	

HTTP Response

- `201 Created`
- `409 Conflict`

9.2.3 User Group Create API

The `<ARM_Configurator_IP>/ARM/v1/identity/userGroup` URL when used with the POST method, provides the ability for the ARM Manager to send a request to the ARM Configurator to create an LDAP User Group.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/identity/userGroup
```

HTTP Method

`POST`

Supported Content-Type

`application/json`

Attributes

Attribute	Type	Description
name	String	The unique name of the LDAP server user group. If the name is not unique, 409 CONFLICT is returned.
associatedDialable	String	Dialable property associated with the user group. The User Group must have an associated dialable property, otherwise, a 409 CONFLICT is returned.

conditions	[
	{
	dicAttrName: String
	operator: LogicalOperatorEnum
	values:
	[
	String
	]
	}
	]
		LogicalOperatorEnum:
		▪ EQUALS (0)
		▪ NOTEQUALS (1)
		▪ CONTAINS (2
		▪ NOT_CONTAINS (3)

HTTP Response

- `201 Created`
- `409 Conflict`
- `500 Internal Server Error`

9.2.4 UserGroup Dry Run API

The `<ARM_Configurator_IP>/ARM/v1/identity/userGroup/dryRun` URL when used with the POST method, provides the ability for the ARM Manager to send a request to the ARM Configurator to verify which LDAP users are suitable for a specific LDAP User Group before the User Group is generated.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/identity/userGroup/dryRun
```

HTTP Method

`POST`

Supported Content-Type

`application/json`

Attributes

Attribute	Type	Description
name	String	The Unique name of the user group. If the name is not unique, the 409 CONFLICT is returned.
associatedDialable	String	Dialable property associated with the user group. The User Group must have an associated dialable property, otherwise, a 409 CONFLICT is returned.
conditions	[	LogicalOperatorEnum:

Attribute	Type	Description

	{
	dicAttrName: String
	operator: LogicalOperatorEnum
	values:
	[
	String
	]
	}
	]
		• EQUALS (0)
		• NOTEQUALS (1)
		• CONTAINS (2)
		• NOT_CONTAINS (3)

HTTP Response

- `201 Created`
- `409 Conflict`
- `500 Internal Server Error`

A filter should be added to the suffix of the URL, for example:

Filter Example

```
GET
<ARM_Configurator_IP>/ARM/v1/identity/userGroup/dryRun?filter=(seachString~A)
```

Filter	Available for these Requests	Description
searchString	user	Filters according to any attribute of the user (Contains ~).
limit	user	Limit the number of the results
before	user	Returns all the events prior to this id (larger than this ID).
after	user	Returns all the events after this ID (smaller than this ID).
sort	user	Sorts the result.

9.2.5 User Create API

The `<ARM_Configurator_IP>/ARM/v1/identity/user` URL when used with the POST method, provides the ability for the ARM Manager to send a request to the ARM Configurator to create an LDAP user.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/identity/user
```

HTTP Method

`POST`

Supported Content-Type

`application/json`

Attributes

Attribute	Type	Description
name	String	A unique name for local users. If the name is not unique, 409 CONFLICT message is returned.

dicAttributes	[
	{
	dicAttrId: Integer
	value: String
	}
	]
		Dynamic JSON constructed according to dicAttributes. If dicAttrId does not exist, 409 CONFLICT status is returned.

HTTP Response

- `201 Created`
- `409 Conflict`
- `500 Internal Error`

9.2.6 Dictionary Attribute Create API

The `<ARM_Configurator_IP>/ARM/v1/identity/dicAttribute` URL when used with the POST method, provides the ability for the ARM Manager to send a request to the ARM Configurator to create a Property Dictionary (for mapping with the AD objects) in the UI.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/identity/dicAttribute
```

HTTP Method

`POST`

Supported Content-Type

`application/json`

Attributes

Attribute	Type	Description

dicAttribute	{
	name: String
	description: String
	dialable: boolean
	primary: boolean
	defaultRemoteName: String
	}
		The unique name of the dictionary attribute. If the name is not unique, 409 CONFLICT status is returned.
		defaultRemoteName: The default mapping of this property in the LDAP server.

HTTP Response

- `201 Created`
- `409 Conflict`
- `500 Internal Error`

9.2.7 File Repository Create API

The `<ARM_Configurator_IP>/ARM/v1/identity/` fileRepository URL when used with the POST method, provides the ability for the ARM Manager to send a request to the ARM Configurator to create a File Repository (for mapping with the AD objects) in the ARM Web interface.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/identity/fileRepository
```

HTTP Method

`POST`

Supported Content-Type

application/json

Attributes

Attribute	Type	Description
name	String	Indicates the unique name for file repositories. If the name is not unique, 409 CONFLICTED status is returned.
status	FileRepositoryStatus: ▪ INITIALIZING (0) ▪ ACTIVE (1) ▪ ERROR (2)	Indicates the status of the File Repository as indicated in the adjacent column.
delimiter	Character	Indicates the character used to separate entries in the File repository. This can be one of the following: ‘,’, ‘;’ or ‘|’).
isFileHasHeaders	boolean	Indicates whether the file contains headers in the first line.
uniquePropertyId	Integer	When the unique property of the user repository is a user dictionary attribute, indicates the ID of this unique attribute. If 'Name' is set as the unique attribute for the user repository, then this attribute is Null.
isNameUnique	boolean	If the unique property is set to “Name”, this attribute is TRUE. If the unique property is taken from the property dictionary, this attribute is FALSE.
nameFileMapping	String	File mapping for the “name” property.
nameFileLocation	Integer	Indicates the column location e.g. 3 indicates that the “Name” property is the third column in the file.

dicAttrMapping	[
	{
	dicAttrId: Integer
	dicAttrName: String
	fileRepositoryAttrName: String
	fileRepositoryAttrLocation: Integer
	normalizationGroupId: Integer
	}
	]
		Mapping of the dictionary attributes specified in the adjacent array.

9.3 Identity Update API

This section describes the Identity Update API.

9.3.1 Server Update API

The `<ARM_Configurator_IP>/ARM/v1/identity/server/{id}` URL when used with the PUT method, provides the ability for the ARM Manager to send a request to the ARM Configurator to update an LDAP server.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/identity/server/{id}
```

Where {id} is the element in the database.

HTTP Method

`PUT`

Supported Content-Type

`application/json`

Attributes

Attribute	Type	Description
name	String	A unique name for the LDAP server (if the name is not unique, 409 Conflict is returned).
password	String	The password of the LDAP server.
host	String	The name of the LDAP server host.
port	Integer	The LDAP server port
dn	String	The Distinguished Nam of the LDAP server.
searchFilter	String	The LDAP search filter.
baseObject	String	The LDAP base object
sslEnabled	boolean	Determines whether SSL is enabled for the connection with the LDAP server.
certificate	String	The certificate that secures the connection with the LDAP server.
syncIntervalMinutes	Integer	The interval time (in minutes) when partial

Attribute	Type	Description
		synchronization is performed with the LDAP server. This operation retrieves users that have changed since the last partial sync was run.
fullSyncHour	Integer	The hour when full synchronization is performed to retrieve all users from the LDAP server.
fullSyncMinute	Integer	The minute in the hour when full synchronization is performed to retrieve all users from the LDAP server.
timeout	Integer	The timeout for connection with the LDAP server (in minutes).
fullSyncIntervalInDays	Integer	The interval time (in days) when full synchronization is performed to retrieve all users from the LDAP server.

dicAttrMapping	[
	{
	dicAttrId: Integer
	aDAttrName: String
	normalizationGroup: Integer
	}
	]
		The dictionary attribute mapping with the LDAP server.

HTTP Response

- `200 OK`
- `409 Conflict` status for non-existing id

9.3.2 User Group Update API

The `<ARM_Configurator_IP>/ARM/v1/identity/userGroup/{id}` URL when used with the PUT method, provides the ability for the ARM Manager to send a request to the ARM Configurator to update an Indentiy User Group.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/identity/userGroup/{id}
```

Where {id} is the element in the database.

HTTP Method

`PUT`

Supported Content-Type

`application/json`

Attributes

Attribute	Type	Description
name	String	The unique name of the LDAP user group. If the name is not unique, 409 CONFLICT is returned
associatedDialable	String	Dialable property associated with the user group. The User Group must have an associated dialable property, otherwise, a 409 CONFLICT is returned.

conditions	[
	{
	dicAttrName: String
	operator: LogicalOperatorEnum
	values:
	[
	value:String
	]
	}
	]
		LogicalOperatorEnum:
		• EQUALS (0)
		• NOTEQUALS (1)
		• CONTAINS (2)
		• NOT_CONTAINS (3)

HTTP Response

- `200 OK`
- `409 Conflict` for non-existing id.

9.3.3 User Update API

The `<ARM_Configurator_IP>/ARM/v1/identity/user/{id}` URL when used with the PUT method, provides the ability for the ARM Manager to send a request to the ARM Configurator to update an Identity user.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/identity/user/{id}
```

Where {id} is the element in the database.

HTTP Method

`PUT`

Supported Content-Type

`application/json`

Attributes

Attribute	Type	Description
name	String	The unique name for local users. If the name is not unique, 409 CONFLICT is returned.

dicAttributes	[
	{
	dicAttrId: Integer
	value : String
	}
	]
		Dynamic JSON constructed according to dictionary attributes.
		Validation: it is not possible to define attributes for a remote user which are part of the mapping between the dictionary and the user's AD.
		If this occurs, 409 CONFLICT is returned.

HTTP Response

- `200 OK`
- `409 Conflict` for non-existing id

9.3.4 Dictionary Attribute Update API

The `<ARM_Configurator_IP>/ARM/v1/identity/dicAttribute/{id}` URL when used with the PUT method, provides the ability for the ARM Manager to send a request to the ARM Configurator to update a Dictionary attribute.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/identity/dicAttribute/{id}
```

Where {id} is the element in the database.

HTTP Method

`PUT`

Supported Content-Type

`application/json`

Attributes

Attribute	Type	Description
name	String	The unique name for the dictionary attribute. If the name is not unique, 409 CONFLICT is returned.
description	String	The description of the dictionary attribute.
dialable	boolean	Indicates whether the LDAP AD property is 'dialable'.
primary	boolean	Indicates whether to display the 'primary' property in the user's screen.

HTTP Response

- `200 OK`
- `409 Conflict` for non-existing id

9.3.5 File Repository Update API

The **<ARM_Configurator_IP>/ARM/v1/identity/fileRepository/{id}** URL when used with the PUT method, provides the ability for the ARM Manager to send a request to the ARM Configurator to update a specific File Repository attribute.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/identity/ fileRepository/{id}
```

HTTP Method

```
PUT
```

Supported Content-Type

```
application/json
```

Attributes

Attribute	Type	Description
name	String	Indicates the unique name for the file repository. If not, 409 CONFLICTED status returned.
status	FileRepositoryStatus: • INITIALIZING (0) • ACTIVE (1) • ERROR (2)	Indicates the status of the file repository as specified in the adjacent column.
delimiter	Character	Indicates the character used to separate entries in the File repository. This can be one of the following: ‘,’, ‘;’ or ‘|’).
isFileHasHeaders	boolean	Indicates whether the file contains headers in the first line.
uniquePropertyId	Integer	If the unique property is set to “Name”, this attribute is TRUE. If the unique property is taken from the property dictionary, this attribute is FALSE.
isNameUnique	boolean	If the unique property is set to “Name”, this attribute is TRUE. If the unique property is taken from the property dictionary, this attribute is FALSE.
nameFileMapping	String	File mapping for the “name” property.
nameFileLocation	Integer	Indicates the column location e.g. 3 indicates that the “Name” property is the third column in the file.
dicAttrMapping	[{	Mapping of the dictionary attributes specified in the adjacent array

Attribute	Type	Description
	dicAttrId: Integer dicAttrName: String fileRepositoryAttrName: String fileRepositoryAttrLocation: Integer normalizationGroupId: Integer }]	

9.4 Identity Delete API

This section describes the Delete API.

9.4.1 Server Delete API

The `<ARM_Configurator_IP>/ARM/v1/identity/server/{id}` URL when used with the DELETE method, provides the ability for the ARM Manager to send a request to the ARM Configurator to delete an Identity server. When deleting the server, all its users' local attributes are also deleted.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/identity/server/{id}
```

Where {id} is the element in the database.

HTTP Method

`DELETE`

HTTP Response

`409 Conflict` for non-existing id.

9.4.2 User Group Delete API

The `<ARM_Configurator_IP>/ARM/v1/identity/userGroup/{id}` URL when used with the DELETE method, provides the ability for the ARM Manager to send a request to the ARM Configurator to delete a User Group.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/identity/userGroup/{id}
```

Where {id} is the element in the database.

HTTP Method

`DELETE`

Supported Content-Type

`application/json`

HTTP Response

`409 Conflict` status when trying to delete a User Group with associated Routing Rules.

9.4.3 User Delete API

The `<ARM_Configurator_IP>/ARM/v1/identity/user/{id}` URL when used with the DELETE method, provides the ability for the ARM Manager to send a request to the ARM Configurator to delete a user.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/identity/user/{id}
```

Where {id} is the element in the database.

For multiple delete:

```
<ARM_Configurator_IP>/ARM/v1/identity/user?filter=(id=id1;id2;…)
```

The response will be a multi-response JSON with the appropriate response code for each request id.

HTTP Method

`DELETE`

Supported Content-Type

`application/json`

HTTP Response

`409 Conflict` when trying to delete an AD user.

Attributes (Multi-delete)

Attribute	Type	Description

multiStatusJsonList	[
	{
	status: Integer
	description: String
	id: Integer
	}
	]
		• Status: the http response code for this specific id
		• Description: the description
		• Id: the id of the object for this specific response

9.4.4 Dictionary Attribute Delete API

The `<ARM_Configurator_IP>/ARM/v1/identity/dicAttribute/{id}` URL when used with the DELETE method, provides the ability for the ARM Manager to send a request to the ARM Configurator to delete a Dictionary attribute.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/identity/dicAttribute/{id}
```

Where {id} is the id of the element in the database.

For multiple delete actions:

```
<ARM_Configurator_IP>/ARM/v1/identity/dicAttribute?filter=(id=id1;id2;…)
```

HTTP Method

`DELETE`

Supported Content-Type

`application/json`

Attributes (Multi-delete)

Attribute	Type	Description

multiStatusJsonList	[
	{
	status: Integer
	description: String
	id: Integer
	}
	]
		Status: the http response code for this specific id
		Description: the description
		Id: the id of the object for this specific response

HTTP Response

The response will be a multi-response JSON with one of the following response codes for each id that is sent:

- `409 Conflict`
- `500 Internal Error`

9.4.5 File Repository Delete API

The `<ARM_Configurator_IP>/ARM/v1/identity/fileRepository/{id}` URL when used with the DELETE method, provides the ability for the ARM Manager to send a request to the ARM Configurator to delete a specific File Repository.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/identity/fileRepository/{id}
```

HTTP Method

`DELETE`

Supported Content-Type

`application/json`

10 Statistics REST API

This chapter describes the REST API resources for ARM statistics.

10.1 Get Report API

ARM provides an API for retrieving reports.

REST Resource

```
<ARM_Configurator_IP>/v1/statistics/report/<report_identifier>?filter=(dateType=RANGE,startDate=1542492000,endDate=1542578340,statistics=SOURCE_CALL_ATLTERNATIVE_ROUTING_ATTEMPTS )
```

HTTP Method

```
GET
```

Supported Content-Type

```
application/json
```

Attributes

filter	Type	Description
reportIdentifier <report_identifier> in the url	ReportIdentifier	ReportIdentifier: armOverTime, nodeOverTime, topNodes, topNodesOverTime, nodeStackedByPeerConnection, topNodesStckedByPeerConnection, peerConnectionOverTime, topPeerConnections, topPeerConnectionsOverTime, connectionOverTime, topConnections, topConnectionsOverTime, routerOverTime, topRouters, topRoutersOverTime, routingGroupOverTime, topRoutingGroups, topRoutingGroupsOverTime, routingGroupsStackedByRoutingRules, topRoutingGroupsStackedByRoutingRules, routingRuleOverTime, topRoutingRules, topRoutingRulesOverTime, routingRuleStackedByActions, topRoutingRulesStackedByActions;
dateType	DateType	DateType: RANGE,

filter	Type	Description
		RELATIVE; Must
startDate	Long	unix time in seconds Must when DateType is *RANGE*
endDate	Long	unix time in seconds Must when DateType is *RANGE*
relativeTimeUnit	TimeUnit	TimeUnit: MONTH, WEEK, DAY, HOUR, MINUTE; Must when DateType is *RELATIVE*
relativeTimeUnitValue	Integer	Must when DateType is *RELATIVE*
numberOfElements	Integer	
elements	[Integer]	IDs of selected elements.
statistics	Array of StatisticType	StatisticsType: SOURCE_CALL_1ST_ROUTING_ATTEMPTS, SOURCE_CALL_ATLTERNATIVE_ROUTING_ATTEMPTS, SOURCE_CALL_FAILED_ROUTING_ATTEMPTS, SOURCE_CALL_DISCARD_ROUTING_ATTEMPTS, DESTINATION_CALL, TRANSIENT_CALL,ss ROUTING_RULE_ACTION_1ST_MATCH, ROUTING_RULE_ACTION_2ND_MATCH, ROUTING_RULE_ACTION_3RD_MATCH, SOURCE_CALL_DROP_ROUTING_REQUEST, ROUTING_RULE_ACTION_TRY, ROUTING_RULE_ACTION_FAIL, NO_MATCH_RULE; Must
overTimeGraphType	OverTimeGraphType	OverTimeGraphType: MULTUPLE_STATISTICS, MULTUPLE_ELEMENTS; Must only in reports over time (except from ARM over time). Multiple statistics- means few statistics on one element.

filter	Type	Description
		Multiple elements – means few elements on one statistic.

Response:

Attribute	Type	Description

data	{
	"startDate": Long,
	"interval": Integer,
	"series": [
	{
	"elementName": String,
	"statisticName": String,
	"data": [
	{
	"name": String,
	"id": Integer,
	"y": Long
	}
	]
	}
	]
	}
		Interval should be in seconds.
		
		The order in series. Data list is important in case of over time graph.
		
		The first point in the data array representing the start date and each point after it is the previous date + interval.

title	String	
Subtitle	String	
graphType	GraphType	GraphViewType: COLUMN, LINE, STACKED_COLUMN;
preferences		This is in case we will want to add more information in the future.

HTTP Response

- `200 OK`
- `409 Conflict`
- `500 Internal Error`

10.1.1 Top Routes Statistics

The `<ARM_Configurator_IP>/ARM/v1/ performanceMonitoring/topRoutes?filter=(startTime={startTime},endTime={endTime})` URL when used with the GET method, provides the ability for the ARM Manager to send a request to the ARM Configurator to retrieve top routes. Top routes represent the busiest routes for a period of time.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/ performanceMonitoring/topRoutes?
filter=(startTime={startTime},endTime={endTime})
{ startTime } - unix time (seconds)
{endTime} - unix time (seconds)
```

HTTP Method

GET

Supported Content-Type

application/json

Response Attributes

Attribute	Type	Description

Top Routes	ARRAY OF:
	[
	{
	startTime: integer
	endTime: integer
	startTimeTs: String
	endTimeTs: String
	dataValid: boolean
	routes:
	[
	{
	route: String
	srcNode: integer
	srcPeerConnection: integer
	destNode: integer (optional)
	destPeerConnection: integer (optional)
	hits:integer
	
	edges:[(optional)
	{
	connSrcNode: integer
	connDestNode: integer
	connectionId: integer
	
	}
	
	]
	}
	]
	}
	]
		The query elements for the node upon which the data is collected.

10.1.1.1 Filter

The statistics API requires using a filter to control the size and content of the data. The filter should be added to the suffix of the URL, for example:

REST Resource

```
 <ARM_Configurator_IP>/ARM/v1/
performanceMonitoring/topRoutes?filter=(startTime={startTime},endT
ime={endTime}mmaxRoutes={max routes})
```

HTTP Method

`GET`

Supported Content-Type

`application/json`

Response Attributes

Filter	Available for these requests	Description
startTime	For all resources	Results data timestamp beginning from startTime. Where startTime is the unix time in seconds.
endTime	For all resources	Results data timestamp ending by endTime. Where endTime is the unix time in seconds.
maxRoutes	For all resources	The maximum number of routes to return.

HTTP Response

- `200 OK`
- `409 Conflict`
- `500 Internal Error`

This page is intentionally left blank.

11 Alarms History

This chapter describes the REST API resources for the Alarms History.

11.1 Get API

The `<ARM_Configurator_IP>/ARM/v1/alarms/history` URL when used with the GET method, provides the ability for the ARM Manager to send a request to the ARM Configurator to retrieve alarm events.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/alarms/history
<ARM_Configurator_IP>/ARM/v1/alarms/history/{id}
```

Where {id} is the id of a specific event.

HTTP Method

`GET`

Supported Response Attributes

The following attributes MUST be included:

Attribute	Type	Description
id	Integer	The internal id of the alarm in the database.
alarmDateAndTime	Date	The time that the ARM Configurator raised the alarm. In the following format: 2010-03-01T22:38:37.000Z (UTC time)
name	String	The name of the alarm For example: “Operation status changed”
alarmSource	String	The source of the alarm
alarmSeverity	Integer	The alarm severity: ▪ cleared(0) ▪ indeterminate(1) ▪ warning(2) ▪ minor(3) ▪ major(4) ▪ critical(5)
alarmType	Integer	The alarm type: • other(0) • communicationsAlarm(1) • qualityOfServiceAlarm(2) • processingErrorAlarm(3)

Attribute	Type	Description
		• equipmentAlarm(4) • environmentalAlarm(5) • integrityViolation(6) • operationalViolation(7) • physicalViolation(8) • securityServiceOrMechanismViolation(9) • timeDomainViolation(10)
alarmProbCause	Integer	The probable cause of the alarm.
alarmTextualDescription	String	The text description of the alarm.
alarmAddInfo1	String	The additional information for the alarm received in the AddInfo1 field.
alarmAddInfo2	String	The additional information for the alarm received in the AddInfo2 field.
alarmAddInfo3	String	The additional information for the alarm received in the AddInfo3 field.
nodeId	Integer	The id of the node where the alarm is raised.
routingInterfaceId	Integer	The routing interface id of the alarm.
alarmUniqId	Integer	The internal id of the alarm in the database.
isAlarm	Boolean	Indicates whether an alarm or an event.
clearedBy	Integer	Indicates which event cleared the alarm (Optional)

HTTP Response

- `409 Conflict`
- `200 OK` response if at least one resource is found.
- `204 No Content Response`

11.1.1 Get Filter API

There is a possibility to use filter for specific types of requests. The filter should be added to the suffix of the URL, for example:

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/alarms/history?filter=(severity=minor)
```

HTTP Method

`GET`

Supported Content-Type

`application/json`

Attributes

Filter	Operation	Description
searchString	contains	The search string for the filter query. Results are retrieved only if the provided string is part of the following columns values (or operation between columns): name, source, description
name	Equals or contains	The name of the alarm specified in the filter query. Results are retrieved only if the names specified in the filter is matched.
source	contains	The alarm source specified in the filter query. The result includes only alarms whose source contains the source string defined in the filter (the source in the filter if part of the source string in the alarm).
description	contains	The alarm description specified in the filter query. Results are retrieved only if the part of the description specified in the filter is matched.
severity	equals	The severity specified in the filter query. Results are retrieved only if the part of the severity specified in the filter is matched.
startDate	equals	The start date for the filter query. Results are retrieved only if the actual startDate is larger or equal to the startDate specified in the filter.
endDate	equals	The end date for the filter query. Results are retrieved only if the actual endDate is less or equal to the endDate specified in the filter.
isAlarm	equals	Whether the History view shows alarms or events. By default, both alarms and events are displayed.

11.1.2 Cursor API

Filter	Operation	Description
before	Return the latest alarms	Returns all the alarms before this id (larger than this id).
after	Return the oldest alarms	Returns all the alarms after this id (smaller than this id).
limit	Limits the number of retrieved alarms for analysis purposes	Limits the number of results.

HTTP Response

- `200 OK` response if at least one resource is found.
- `204 No Content Response`

11.1.3 Get Event Count API

The `<ARM_Configurator_IP>/ARM/v1/alarms/history/count` URL when used with the GET method, provides the ability for the ARM Manager to send a request to the ARM Configurator to retrieve the number of history alarms.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/alarms/history/count
```

HTTP Method

`GET`

Supported Content-Type

`application/json`

Attributes

The following attributes MUST be included:

Attribute	Type	Description

SeverityCount	ARRAY OF:
	[
	{
	type: AlarmSeverity,
	count: Long
	}
	]
		AlarmSeverity:
		▪ cleared
		▪ indeterminate
		▪ warning
		▪ minor
		▪ major
		▪ critical

Note: The absence of a count record for a specific severity, implies that there are no alarms for this type.

Cursor API

Filter	Operation	Description
before	Return the latest alarms	Returns all the events before this id (larger than this id).
after	Return the oldest alarms	Returns all the events after this id (smaller than this id).
limit	Limit the number of retrieved alarms for analysis purposes.	Limits the number of the results.

HTTP Response

- `200 OK` response if at least one resource is found
- `204 No Content Response`

This page is intentionally left blank.

12 Active Alarms

This chapter describes the REST API resources for the Active Alarms.

12.1 Get API

The `<ARM_Configurator_IP>/ARM/v1/alarms/active` URL when used with the GET method, provides the ability for the ARM Manager to send a request to the ARM Configurator to retrieve a list of active alarms or a specific alarm event.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/alarms/active
<ARM_Configurator_IP>/ARM/v1/alarms/active/{id}
```

Where {id} is the id of the requested event.

HTTP Method

`GET`

Supported Content-Type

`application/json`

Attributes

The following attributes MUST be included:

Attribute	Type	Description
id	Integer	The internal id of the alarm in the database.
alarmDateAndTime	Date	The time and date that the ARM Configurator raised the alarm. In the following format: 2010-03-01T22:38:37.000Z (UTC time)
name	String	The name of the alarm For example: “Operation status changed”
alarmSource	String	The source of the alarm
alarmSeverity	Integer	The alarm severity
alarmType	Integer	The alarm type

Attribute	Type	Description
alarmProbCause	Integer	The probable cause of the alarm.
alarmTextualDescription	String	The text description of the alarm.
alarmAddInfo1	String	The additional information for the alarm received in the AddInfo1 field.
alarmAddInfo2	String	The additional information for the alarm received in the AddInfo2 field
alarmAddInfo3	String	The additional information for the alarm received in the AddInfo3 field
nodeId	Integer	The id of the node where the alarm is raised.
routingInterfaceId	Integer	The routing interface id of the alarm.
alarmUniqId	Integer	The internal id of the alarm in the database.
isAlarm	Boolean	Indicates whether this is an alarm.
clearedBy	Integer	(Optional) Indicates which event was cleared by the alarm.

HTTP Response

- `409 Conflict`
- `200 OK` response if at least one resource is found
- `204 No Content Response`

12.1.1 Get Filter API

You can use a filter for specific requests. The filter should be added to the suffix of the URL, for example:

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/alarms/active?filter=(severity=minor)
```

HTTP Method

`GET`

Supported Content-Type

`application/json`

Attributes

Filter	Operation	Description
searchString	contains	The search string for the filter query. Results are retrieved only if the provided string is part of the following columns values (or operation between columns): name, source, description
name	equals or contains	The name of the alarm specified in the filter query. Results are retrieved only if the names specified in the filter is matched.
source	contains	The alarm source specified in the filter query. Results are retrieved only if the part of the alarm source specified in the filter is matched.
description	contains	The alarm description specified in the filter query. Results are retrieved only if the part of the description specified in the filter is matched.
severity	equals	The severity specified in the filter query. Results are retrieved only if the part of the severity specified in the filter is matched.
startDate	equals	The start date for the filter query. Results are retrieved only if the actual startDate is larger or equal to the startDate specified in the filter.
endDate	equals	The end date for the filter query. Results are retrieved only if the actual endDate is less or equal to the endDate specified in the filter.
isAcknowledged	equals	Return only if the alarm was acknowledged by the operator.

12.1.2 Cursor API

Filter	Operation	Description
before	Return the latest alarms	Returns all the events before this id (larger than this id).
after	Return the oldest alarms	Returns all the events after this id (smaller than this id).
limit	Limit the number of retrieved alarms for analysis purposes.	Limits the number of the results.

HTTP Response

- `200 OK` response if at least one resource is found
- `204 No Content Response`

12.1.3 Get Alarm Count API

The `<ARM_Configurator_IP>/ARM/v1/alarms/active/count` URL when used with the GET method, provides the ability for the ARM Manager to send a request to the ARM Configurator to retrieve the number of active alarms.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/alarms/active/count
```

HTTP Method

`GET`

Supported Content-Type

`application/json`

Attributes

The following attributes MUST be included:

Attribute	Type	Description

SeverityCount	ARRAY OF:
	[
	{
	type: AlarmSeverity,
	count: Long
	}
	]
		AlarmSeverity:
		• cleared
		• indeterminate
		• warning
		• minor
		• major
		• critical

HTTP Response

`200 OK` response if at least one resource is found

Note: The absence of a count record for a specific severity, implies that there are no alarms for this type.

12.2 Update API

The `<ARM_Configurator_IP>/ARM/v1/alarms/active/{id}` URL when used with the PUT method, provides the ability for the ARM Manager to send a request to the ARM Configurator to update the Active alarms.

REST Resource

```
PUT <ARM_Configurator_IP>/ARM/v1/alarms/active/{id}
```

Where {id} is the id of the requested event.

HTTP Method

`PUT`

Supported Content-Type

`application/json`

Attributes

The following attributes MUST be included:

Attribute	Type	Description
isAcknowledged	Boolean	Whether the alarm was acknowledged by the operator.

HTTP Response

- `409 Conflict` in the absence of an id
- `200 OK` response if at least one resource is found.

This page is intentionally left blank.

13 Alarms Additional Info

This section describes the URLs for supporting the customization of information provided in notifications for call matching a specific rule (Routing Rule Match), usually applied for emergency calls such as 911 calls e.g. user name, building, floor, country or office branch name. This information is not part of the SIP INVITE message but can be added to the ARM users database and used for additional information in notifications.

13.1 Get Alarms Additional Info

The `<ARM_Configurator_IP>/ARM/v1/alarms/additionalInfo` and `<ARM_Configurator_IP>/ARM/v1/alarms/additionalInfo/{id}` URLs when used with the GET method, provides the ability for the ARM Manager to send a request to the ARM Configurator to retrieve Additional Info for all alarms or for a specific alarm.

REST Resource

```
GET <ARM_Configurator_IP>/ARM/v1/alarms/additionalInfo
GET <ARM_Configurator_IP>/ARM/v1/alarms/additionalInfo/{id}
```

HTTP Method

`GET`

Supported Content-Type

`application/json`

Attributes

Attribute	Type	Description
id	Integer	The SNMP alarm OID
eventType	Integer	The event type
matchingProperty	MatchingProperty	MatchingProperty: • SRC_URI_NAME(0) • SRC_URI_USER(1) • SRC_URI_HOST(2) • DST_URI_NAME(3) • DST_URI_USER(4) • DST_URI_HOST(5) • SRC_IP_FROM_LYNC(6)

Attribute	Type	Description
matchingMethod	MatchingMethod	MatchingMethod: • FULL(0) • CONTAINS(1) • NETWORK_MASK(2)
matchedProperty	Integer	The property ID in the database.
template	String	The text contained in the additional info pattern.
isEnabled	Boolean	Indicates if this rule match\additional info is enabled
eventName	String	The event name

13.2 Test Alarms Additional Info

The `<ARM_Configurator_IP>/ARM/v1/alarms/additionalInfo/test` API when used with the PUT method, provides the ability for the ARM Manager to send a request to the ARM Configurator to test alarms Additional Info.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/alarms/additionalInfo/test
```

HTTP Method

```
PUT
```

Supported Content-Type

```
application/json
```

Attributes

Attribute	Type	Description
eventType	Integer (default 15)	The event type

Attribute	Type	Description
matchingProperty	MatchingProperty	MatchingProperty: • SRC_URI_NAME(0) • SRC_URI_USER(1) • SRC_URI_HOST(2) • DST_URI_NAME(3) • DST_URI_USER(4) • DST_URI_HOST(5) • SRC_IP_FROM_LYNC(6)
matchingMethod	MatchingMethod	MatchingMethod: • FULL(0) • CONTAINS(1) • NETWORK_MASK(2)
matchedProperty	Integer	The property ID in the data base.
template	String	Default value: “” (empty string)
eventName	String (default Routing Rule Match)	The event name
testValue	String	Test result for GET

This page is intentionally left blank.

14 Syslog

This chapter describes the REST API resources for the Syslog server. The following APIs are described:

- Get API
- Create API
- Update SNMP Target API
- Delete API

14.1 Get API

The `<ARM_Configurator_IP>/ARM/v1/externals/syslog` URL when used with the GET method, provides the ability for the ARM Manager to send a request to the ARM Configurator to retrieve the entire syslog or a specific syslog record.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/externals/syslog
<ARM_Configurator_IP>/ARM/v1/externals/syslog/{id}
```

Where:

{id} is the id of the requested element.

HTTP Method

`GET`

Supported Content-Type

`application/json`

Attributes

The following attributes MUST be included:

Attribute	Type	Description

syslog	ARRAY OF:
	[
	{
	id: Integer
	host: String
	protocol: String
	enabled: Boolean
	sourceType: String
	level:String
	port:Integer
	}
	]
	cdrFormat: CdrType
		▪ Protocol: UDP or TCP
		▪ Enabled : Whether the syslog is enabled
		• sourceType: the source of the syslog either TOPOLOGY(0) or ROUTER(1)
		• level: the detail level of the syslog, enum: TRACE,DEBUG,INFO,WARN,ERROR
		• CdrType enum: CDR_AS_CLEAR_TEXT, CDR_AS_JSON, BOTH

HTTP Response:

- `409 Conflict` status for a non-existent id.
- `200 OK` response if at least one resource is found.

14.1.1 Get Filter

You can use a filter for specific requests. The filter should be added to the suffix of the URL, for example:

REST Resource

```
GET <ARM_Configurator_IP>/ARM/
v1/externals/syslog?filter=(sourceType=ROUTER)
```

HTTP Method

`GET`

Supported Content-Type

`application/json`

Attributes

The following attributes MUST be included:

Filter	Available for these Requests	Description
sourceType	syslog	Returns the requested source type.

14.2 Create API

The `<ARM_Configurator_IP>/ARM/externals/syslog` URL when used with the POST method creates a syslog.

REST Resource

```
<ARM_Configurator_IP>/ARM/externals/syslog
```

HTTP Method

`POST`

Supported Content-Type

`application/json`

Attributes

The request MUST contain JSON data that consists of the following elements:

Attribute	Type	Description
host	String	The IP address of the remote
port	Integer	Valid port number
protocol	String	Either UDP or TCP
enabled	Boolean	Whether the syslog is enabled or not
sourceType	String	The source of the syslog either TOPOLOGY(0) or ROUTER(1)
level	String	level: the detail level of the syslog, enum: TRACE,DEBUG,INFO,WARN,ERROR

HTTP Response

- `201 Created`
- `409 Conflict` status for a non-existent id.
- `500 Internal error`

14.3 Update API

The `<ARM_Configurator_IP>/ARM/externals/syslog/` URL when used with the PUT method updates a Syslog record.

REST Resource

```
<ARM_Configurator_IP>/ARM/externals/syslog/{id}
```

HTTP Method

`PUT`

The request MUST contain JSON data that consists of the following elements:

Attribute	Type	Description
host	String	The IP address of the remote host
port	Integer	The valid port number.
protocol	String	The protocol: • UDP • TCP
enabled	Boolean	Whether the syslog is enabled
sourceType	String	The source of the syslog: • TOPOLOGY(0) • ROUTER(1)

Attribute	Type	Description
level	String	The detailed level of the syslog, enum: TRACE,DEBUG,INFO,WARN,ERROR

HTTP Response

- `200 OK`
- `409 Conflict` status for a non-existent id.
- `500 Internal error`

14.4 Delete API

The `<ARM_Configurator_IP>/ARM/v1/externals/syslog/{id}` URL enables you to delete a syslog.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/externals/syslog/{id}
```

HTTP Method

`DELETE`

Supported Content-Type

`application/json`

HTTP Response

`200 OK` response if at least one resource is found.

15 SNMP

This chapter describes the REST API resources for SNMP Managers. The following APIs are described:

- Get API
- Create SNMP Target API
- Update SNMP Target API
- HTTP Response
- 200 OK
- `409 Conflict` status for a non-existent id.
- `500 Internal error`
- Delete SNMP Target API

15.1 Get API

`The <ARM_Configurator_IP>/ARM/v1/externals/snmp/target/` URL when used with the GET method, URL provides the ability for the ARM Manager to send a request to the ARM Configurator to retrieve SNMP Managers.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/externals/snmp/target/
<ARM_Configurator_IP>/ARM/v1/externals/snmp/target/{id}
```

Where {id} is the id of the requested element.

HTTP Method

`GET`

Supported Content-Type

`application/json`

Attributes

Attribute	Type	Description

snmp	ARRAY OF:
	[
	{
	id: Integer
	host: String
	port: Integer
	community: String
	}
	]
		The SNMP Manager elements.

HTTP Response

- `409 Conflict` when an id is not included in the request.
- `200 OK` response if at least one resource is found.

15.2 Create SNMP Target API

The `<ARM_Configurator_IP>/ARM/v1/externals/snmp/target` URL when used with the POST method creates a target SNMP Manager.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/externals/snmp/target
```

HTTP Method

`POST`

Supported Content-Type

`application/json`

Attributes

The request MUST contain JSON data that consists of the following elements:

Attribute	Type	Description
host	STRING	Valid IP address of the SNMP Manager
port	INTEGER	Valid port of the SNMP Manager
community	STRING	The SNMPv2 community string “password”

HTTP Response

- `201 Created`
- `409 Conflict` status for a non-existent id.
- `500 Internal error`

15.3 Update SNMP Target API

The `<ARM_Configurator_IP>/ARM/v1/externals/snmp/target/{id}URL` when used with the PUT method updates the details of an SNMP Manager.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/externals/snmp/target/{id}
```

Where {id} is the id of the element in the database.

HTTP Method

`PUT`

Attributes

The request MUST contain JSON data that consists of the following elements:

Attribute	Type	Description
host	STRING	Valid IP address of the SNMP Manager
port	INTEGER	Valid port of the SNMP Manager
community	STRING	The SNMPv2 community string “password”.

HTTP Response

- `200 OK`
- `409 Conflict` status for a non-existent id.
- `500 Internal error`

15.4 Delete SNMP Target API

The `<ARM_Configurator_IP>/ARM/v1/externals/snmp/target/{id}` URL provides the ability for the ARM Manager to send a request to the ARM Configurator to delete a Target SNMP Manager.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/externals/snmp/target/{id}
```

Where {id} is the id of the element in the database.

HTTP Method

`DELETE`

Supported Content-Type

`application/json`

HTTP Response

- `200 OK`

This page is intentionally left blank.

16 External Alarms

This chapter describes the URLs for creating external alarms for third-party proprietary equipment such as routers and for third-party applications such as Web Services.

16.1 Create External Alarm

The `<ARM_Configurator_IP>/ARM/v1/externals/alarm` URL when used with the POST method, provides the ability for the ARM Manager to send a request to the ARM Configurator to create a single external alarm.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/externals/alarm
```

HTTP Method

`POST`

Supported Content-Type

`application/json`

Attributes

The request MUST contain JSON data that consists of the following elements:

Attribute	Type	Description
source	STRING	SNMP Alarm source
identifier	STRING	SNMP Alarm OID
description	STRING	Alarm description field
addInfo1		The additional information for the alarm received in the AddInfo1 field.
addInfo2		The additional information for the alarm received in the AddInfo2 field.
severity	AlarmSeverity	AlarmSeverity: ▪ cleared(0), ▪ indeterminate(1), ▪ warning(2), ▪ minor(3), ▪ major(4), ▪ critical(5);
alarmType	AlarmEventType	AlarmEventType: ▪ other(0), ▪ communicationsAlarm(1), ▪ qualityOfServiceAlarm(2), ▪ processingErrorAlarm(3), ▪ equipmentAlarm(4), ▪ environmentalAlarm(5),

Attribute	Type	Description
		▪ integrityViolation(6), ▪ operationalViolation(7), ▪ physicalViolation(8), ▪ securityServiceOrMechanismViolation(9), ▪ timeDomainViolation(10);
probabyCause	AlarmProbableCause	AlarmProbableCause: • other(0), • adapterError(1), • applicationSubsystemFailure(2), • bandwidthReduced(3), • callEstablishmentError(4), • communicationsProtocolError(5), • communicationsSubsystemFailure(6), • configurationOrCustomizationError(7), • congestion(8), • corruptData(9), • cpuCyclesLimitExceeded(10), • dataSetOrModemError(11), • degradedSignal(12), • dteDceInterfaceError(13), • enclosureDoorOpen(14), • equipmentMalfunction(15), • excessiveVibration(16), • fileError(17), • fireDetected(18), • floodDetected(19), • framingError(20), • heatingVentCoolingSystemProblem(21), • humidityUnacceptable(22), • inputOutputDeviceError(23), • inputDeviceError(24), • lanError(25), • leakDetected(26), • localNodeTransmissionError(27), • lossOfFrame(28), • lossOfSignal(29), • materialSupplyExhausted(30), • multiplexerProblem(31), • outOfMemory(32), • ouputDeviceError(33), • performanceDegraded(34), • powerProblem(35), • pressureUnacceptable(36), • processorProblem(37), • pumpFailure(38), • queueSizeExceeded(39), • receiveFailure(40), • receiverFailure(41), • remoteNodeTransmissionError(42), • resourceAtOrNearingCapacity(43), • responseTimeExecessive(44), • retransmissionRateExcessive(45),

Attribute	Type	Description
		softwareError(46), • softwareProgramAbnormallyTerminated(47), • softwareProgramError(48), • storageCapacityProblem(49), • temperatureUnacceptable(50), • thresholdCrossed(51), • timingProblem(52), • toxicLeakDetected(53), • transmitFailure(54), • transmitterFailure(55), • underlyingResourceUnavailable(56), • versionMismatch(57), • authenticationFailure(58), • breachOfConfidentiality(59), • cableTamper(60), • delayedInformation(61), • denialOfService(62), • duplicateInformation(63), • informationMissing(64), • informationModificationDetected(65), • informationOutOfSequence(66), • intrusionDetection(67), • keyExpired(68), • nonRepudiationFailure(69), • outOfHoursActivity(70), • outOfService(71), • proceduralError(72), • unauthorizedAccessAttempt(73), • unexpectedInformation(74);

16.2 Create Multiple External Alarms

The `<ARM_Configurator_IP>/ARM/v1/externals/alarms` URL when used with the POST method, provides the ability for the ARM Manager to send a request to the ARM Configurator to create multiple external alarms.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/externals/alarms
```

HTTP Method

`POST`

Supported Content-Type

`application/json`

Attributes

The request MUST contain JSON data that consists of the following elements:

Attribute	Type	Description

[
{
source: String
identifier: String
description: String
addInfo1: String
addInfo2: String
severity: AlarmSeverity
alarmType: AlarmEventType
probabyCause: AlarmProbableCause
}
]		

17 Normalization Group

This chapter describes the REST resources for the Normalization Group API (Attribute Manipulation Group). The following APIs are described:

- Get API
- Create Normalization Group API
- Update Normalization Group Target API
- Delete Normalization Group API
- Test Normalization Group
- Get Normalization Group Associations (Relationships)

17.1 Get API

The `<ARM_Configurator_IP>/ARM/v1/normalization/group` URL when used with the GET method provides the ability for the ARM Manager to send a request to the ARM Configurator to retrieve Normalization Groups.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/normalization/group
<ARM_Configurator_IP>/ ARM/v1/normalization/group/{id}
```

Where {id} is the id of the element in the database.

HTTP Method

`GET`

Supported Content-Type

`application/json`

Attributes

Attribute	Type	Description
normalizationGroup	ARRAY OF: [{ id: Integer groupName: String normalizationAttrs: [{ id: Integer priority: Integer regExp: String	• Priority – the priority of the attribute inside the group • regEx – regular expression that will be matched against the element • replacement – the value to replace with the matched expression

Attribute	Type	Description
	replacement: String url: String }] }]	

HTTP Response

- `409 Conflict` when an id is not contained in the request.
- `200 OK`

17.2 Create Normalization Group API

The `<ARM_Configurator_IP>/ARM/v1/normalization/group` URL when used with the POST method provides the ability for the ARM Manager to send a request to the ARM Configurator to create a Normalization Group.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/normalization/group
```

HTTP Method

`POST`

Supported Content-Type

`application/json`

Attributes

The request MUST contain JSON data that consists of the following elements:

Attribute	Type	Description
groupName	STRING	The name of the group must be unique.

normalizationAttrs
[
{
priority: Integer
regEx: String
replacement: String
}
]
		• priority – the priority of the attribute inside the group
		• regEx – regular expression that will be matched against the element
		• replacement – the value to replace with the matched expression.

HTTP Response

- 201 Created
- 409 Conflict
- 500 Internal Error

17.3 Update Normalization Group Target API

The `<ARM_Configurator_IP>/ARM/v1/normalization/group/{id}` URL when used with the PUT method, updates the details of a Normalization Group.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/normalization/group/{id}
```

Where {id} is the id of the element in the database.

HTTP Method

PUT

Supported Content-Type

application/json

Attributes

The request MUST contain JSON data that consists of the following elements:

Attribute	Type	Description
groupName	STRING	The name of the group, must be unique

normalizationAttrs
[
{
Id: Integer
priority: Integer
regEx: String
replacement: String
}
]
		• Id – the id of the attribute as received in the get request
		• regEx – regular expression that will be matched against the element
		• replacement – the value to replace with the matched expression

HTTP Response

- 200 OK
- 409 Conflict
- 500 Internal Error

Note: To delete the normalization attribute, the update request should be sent to the group, without the attribute you wish to delete.

17.4 Delete Normalization Group API

The `<ARM_Configurator_IP>/ARM/v1/normalization/group/{id}` URL provides the ability for the ARM Manager to send a request to the ARM Configurator to delete a Normalization Group.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/normalization/group/{id}
```

Where {id} is the id of the element in the database.

HTTP Method

`DELETE`

Supported Content-Type

`application/json`

HTTP Response

- `409 Conflict` if the Group is connected to other elements
- `200 OK`

17.5 Test Normalization Group

The `<ARM_Configurator_IP>/ARM/v1/normalization/group/test` URL when used with the POST method, tests a Normalization Group over random input (Rules Simulation for Attribute Manipulation Group).

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/normalization/group/test
```

HTTP Method

`POST`

Supported Content-Type

`application/json`

Request Attributes

The request MUST contain JSON data that consists of the following elements:

Attribute	Type	Description
testString	String	The string which the manipulation will be done on

normalizationAttrs
[
{
priority: Integer
regEx: String
replacement: String
}
]
		• regEx – regular expression that will be matched against the element
		• replacement – the value to replace with the matched expression .

Response Attributes

The response MUST contain JSON data that consists of the following elements:

Attribute	Type	Description
result	String	The result of the normalization on the "testString" field

results:[
{
regEx: String
result: String
}
]
		The intermediate results of each replacement done according to the regular expression

HTTP Response

- `409 Conflict` when there is an error regarding the syntax of the regular expressions (including error description)
- `200 OK`

17.5.1 Examples

The operator may wish to test an existing normalization group, where this may be performed by sending the following HTTP request:

```
POST <ARM_Configurator_IP>/ARM/v1/normalization/group/{id}/test
```

Where {id} is the id of the element in the database.

Request Attribute

Attribute	Type	Description
testString	String	The string upon which the manipulation will be performed.

Response Attributes

Attribute	Type	Description
result	String	The result of the normalization on the “testString” field

results:[
{
regEx: String
result: String
}
]
		The intermediate results of each replacement performed according to the regular expression.

HTTP Response

- `200 OK`

17.6 Get Normalization Group Associations (Relationships)

The `<ARM_Configurator_IP>/ARM/v1/normalization/group/{id}/associations` URL when used with the GET method, provides the ability for the ARM Manager to send a request to the ARM Configurator to retrieve the Normalization Group Associations (Relationships). This request provides details of the relationships of a normalization group to different elements in the system, such as routing rules and peer connections.

REST Resource

`<ARM_Configurator_IP>/ARM/v1/normalization/group/{id}/associations`

HTTP Method

`GET`

Supported Content-Type

`application/json`

Attributes

Attribute	Type	Description

{
routingRules: [
Integer
],
 	
}
		• routingRules – ids of the routing rules to which the normalization group is associated.

HTTP Response

- `200 OK`

This page is intentionally left blank.

18 NTP Server Rest API

This chapter describes the REST resources for the NTP server. The following APIs are described:

- Get API
- Create NTP Server
- Update NTP Server
- NTP Server Delete API

18.1 Get API

The `<ARM_Configurator_IP>/ARM/v1/ntpServer` when used with the GET method, provides the ability for the ARM Manager to send a request to the ARM Configurator to retrieve NTP servers.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/ntpServer
<ARM_Configurator_IP>/ARM/v1/ntpServer/{id}
```

Where {id} is the id of the element in the data.

HTTP Method

`GET`

Supported Content-Type

`application/json`

Attributes

The ARM Configurator must respond to the Get request with JSON data from the request URI element and its id. If the Id was not specified in the Get request URI, JSON data must contain an array of all specified elements defined in the database.

The following attributes MUST be included:

Attribute	Type	Description

Server	ARRAY OF:
	[
	{
	Id: integer
	address: String
	serverName: string,
	}
	]
		IP address or FDQN of the NTP server.

HTTP Response

- `200 OK`

18.2 Create NTP Server

The `<ARM_Configurator_IP>/ARM/v1/ntpServer` URL when used with the POST method, provides the ability for the ARM Manager to send a request to the ARM Configurator to create an NTP server.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/ntpServer
```

HTTP Method

`POST`

Supported Content-Type

`application/json`

Attributes

The request MUST contain JSON data that consists of the following elements:

Attribute	Type	Description
serverName	String	Name of the NTP server.
address	String	IP Address of the NTP server.

HTTP Response

- `201 Created`

18.3 Update NTP Server

The `<ARM_Configurator_IP>/ARM/v1/ntpServer/{id}` URL when used with the PUT method, provides the ability for the ARM Manager to send a request to the ARM Configurator to update an NTP server.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/ntpServer/{id}
```

Where {id} is the id of the element in the database.

HTTP Method

`PUT`

Supported Content-Type

`application/json`

Attributes

The request must contain the following data elements:

Attribute	Type	Description
serverName	String	Name of the NTP server.
address	String	IP address of the NTP server.

HTTP Response

- `200 OK`

18.4 NTP Server Delete API

The `<ARM_Configurator_IP>/ARM/v1/ntpServer/{id}` URL when used with the DELETE method provides the ability for the ARM Manager to send a request to the ARM Configurator to delete an NTP server.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/ntpServer/{id}
```

Where {id} is the id of the element in the database.

HTTP Method

`DELETE`

Supported Content-Type

`application/json`

HTTP Response

- `200 OK`

19 ARM License

This chapter describes the REST resources for the ARM License. The following APIs are described:

- Get License API
- Get ARM Machine ID
- Update License Key API

19.1 Get License API

The `<ARM_Configurator_IP>/ARM/v1/license/key` URL when used with the GET method, provides the ability for the ARM Manager to send a request to the ARM Configurator to retrieve an ARM License.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/license/key
```

HTTP Method

`GET`

Supported Content-Type

`application/json`

Response Attributes

The response should be as follows:

Attribute	Type	Description
url	String	The URL of the ARM license.
license	String	The ARM license string.
machineId	String	The machine id of the ARM Configurator.
limitedByDate	boolean	Whether the license expires according to a specific date.

licenseDate
{
day: String,
month: String,
year: String
}
		The license date.
		Can be null if limitedByDate is false.

Attribute	Type	Description

Features [
{
feature: String,
value: String
 	
}
]
		Array of license features

licenseExists	Boolean	Whether license exists.

HTTP Response

- 200 OK
- 409 Conflict
- 500 Internal Error

19.2 Get ARM Machine ID

The `<ARM_Configurator_IP>/ARM/v1/license/machineId` URL when used with the GET method, provides the ability for the ARM Manager to send a request to the ARM Configurator to retrieve the details of a ARM Machine ID.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/license/machineId
```

HTTP Method

GET

Supported Content-Type

application/json

Attributes

Attribute	Type	Description
machineId	String	The ARM machine Id.

HTTP Response

- 200 OK
- 409 Conflict
- 500 Internal error

19.3 Update License Key API

The `<ARM_Configurator_IP>/ARM/v1/license/key` URL when used with the PUT method provides the ability for the ARM Manager to send a request to the ARM Configurator to update a License Key.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/license/key
```

HTTP Method

`PUT`

Supported Content-Type

`application/json`

Attributes

The request MUST contain JSON data that consists of the following elements:

Attribute	Type	Description
license	String	The valid ARM license string.

HTTP Response

- `200 OK`
- `409 Conflict`
- `500 Internal error`

This page is intentionally left blank.

20 Configuration

This chapter describes additional configuration requests for the following APIs:

- Normalization API
- License Configuration
- Routing Rule Configuration
- Security Configuration
- Quality-based Routing Configuration
- LDAP Authentication
- RADIUS Authentication
- Calls Configuration
- Registered Users
- Analytics
- Registration

20.1 Normalization API

This section describes additional configuration requests for the Normalization API.

20.1.1 Update Routing Normalization API

The `<ARM_Configurator_IP>/ARM/v1/configuration/normalization` URL when used with the PUT method provides the ability for the ARM Manager to send a request to the ARM Configurator to update Routing Normalization API.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/configuration/normalization
```

HTTP Method

`PUT`

Supported Content-Type

`application/json`

Attributes

The request MUST contain JSON data that consists of the following elements:

Attribute	Type	Description
srcNormalization	Integer	Id of the normalization group
destNormalization	Integer	Id of the normalization group

HTTP Response

- `200 OK` response if at least one resource is found.
- `409 Conflict` in case of an incorrect request.

20.1.2 Get Routing Normalization API

The `<ARM_Configurator_IP>/ARM/v1/configuration/normalization` URL when used with the GET method, provides the ability for the ARM Manager to send a request to the ARM Configurator to retrieve a Normalization API.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/configuration/normalization
```

HTTP Method

`GET`

Supported Content-Type

`application/json`

Attributes

Attribute	Type	Description
srcNormalization	Integer	Id of the normalization group
destNormalization	Integer	Id of the normalization group

HTTP Response

- `200 OK` response if at least one resource is found.
- `409 Conflict` in case of an incorrect request.

20.2 License Configuration

This section describes additional configuration requests for the License API.

20.2.1 Get License Configuration

The `<ARM_Configurator_IP>/ARM/v1/configuration/license` URL when used with the GET method, provides the ability for the ARM Manager to send a request to the ARM Configurator to retrieve an ARM License.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/configuration/license
```

HTTP Method

`GET`

Supported Content-Type

`application/json`

Attributes

Attribute	Type	Description
numberOfSessionsAlarmWarning	Integer	The Alarm warning threshold to indicate when the number of valid licensed sessions is exceeded (percentage)
numberOfSessionsAlarmCritical	Integer	The Alarm critical threshold to indicate when the number of valid licensed sessions is exceeded.
currentNumberOfSessions	Integer	The current number of sessions in the system.

HTTP Response

- `200 OK` response if at least one resource is found.
- `409 Conflict` in case of an incorrect request.

20.2.2 Update License Configuration

The `<ARM_Configurator_IP>/ARM/v1/configuration/license` URL when used with the PUT method, provides the ability for the ARM Manager to send a request to the ARM Configurator to update an ARM License.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/configuration/license
```

HTTP Method

`PUT`

Supported Content-Type

`application/json`

The request MUST contain JSON data that consists of the following elements:

Attribute	Type	Description
numberOfSessionsAlarmWarning	Integer	The Alarm warning threshold to indicate when the number of valid licensed sessions is exceeded (percentage).

HTTP Response

- `201 Created`
- `409 Conflict` in case of an incorrect request.

20.3 Routing Rule Configuration

This section describes additional configuration requests for the Routing Rule API.

20.3.1 Routing Rule Configuration

The `<ARM_Configurator_IP>/ARM/v1/configuration/routingRule` URL when used with the PUT method, provides the ability for the ARM Manager to send a request to the ARM Configurator to configure a routing rule.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/configuration/routingRule
```

HTTP Method

`PUT`

Supported Content-Type

`application/json`

Attributes

Attribute	Type	Description
maxRoutes	Integer	The maximum number of routes Default value is 6
maxRoutesPerAction	Integer	The maximum number of routes per action. Default value is 2
maxOfPconsPerVoippeer	Integer	The maximum number of peer connections per VoipPeer Default value is 2

HTTP Response

- `201 Created`
- `409 Conflict` in case of an incorrect request.

20.3.2 Update Routing Rule Configuration

The `<ARM_Configurator_IP>/ARM/v1/configuration/routingRule` URL when used with the PUT method, provides the ability for the ARM Manager to send a request to the ARM Configurator to update a routing rule.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/configuration/routingRule
```

HTTP Method

`PUT`

Supported Content-Type

`application/json`

Attributes

Attribute	Type	Description
maxRoutes	Integer	The maximum number of routes. Default value is 6
maxRoutesPerAction	Integer	The maximum number of routes per action. Default value is 2
maxOfPconsPerVoippeer	Integer	The maximum number of peer connections per VoipPeer Default value is 2

HTTP Response

- `201 Created`
- `409 Conflict` in case of an incorrect request.

Note: All the values must be positive between 1 and 10, otherwise 409 conflict will be returned. Values that are not sent in the JSON will not be updated.

20.4 Security Configuration

This section describes the security configuration.

20.4.1 Get Security Configuration

The `<ARM_Configurator_IP>/ARM/v1/configuration/security` URL when used with the GET method, provides the ability for the ARM Manager to send a request to the ARM Configurator to retrieve the security credentials of ARM users.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/configuration/security
```

HTTP Method

```
GET
```

Supported Content-Type

```
application/json
```

Attributes

Attribute	Type	Description
userName	String	User name that ARM uses for HTTP authentication with the devices. Default value is Admin
Password	String	Password that ARM uses for HTTP authentication with the devices. Default value is Admin
sessionTimeout	Integer	Client session timeout (in hours). Default value is Default value is 5 hours
inactivityPeriod	Integer	Client inactivity period (in minutes). Default value is 5 minutes
httpHttpsEnabled	Boolean	Whether mixed mode is on or off (http+https)
configuratorIpAddress	String	The IP address of the configurator
configuratorHostname	String	The host name of the configurator
IsConfCommunicationIpBased	Boolean	Indicates whether ARM should publish the host name or the ip address
verifyCertificate	Boolean	Indicates whether the configurator should verify

Attribute	Type	Description
		the server certificate before initiating the TLS connection.
verifyCertificateSubjectName	Boolean	Indicates whether the configurator should verify the server's subject name before initiating the TLS connection.

HTTP Response

- `200 OK`
- `500 Internal error`

20.4.2 Update Security Configuration

The `<ARM_Configurator_IP>/ARM/v1/configuration/security` URL when used with the PUT method, provides the ability for the ARM Manager to send a request to the ARM Configurator to update the security credentials of an ARM user.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/configuration/security
```

HTTP Method

`PUT`

Supported Content-Type

`application/json`

Attributes

Attribute	Type	Description
userName	String	User name that ARM uses for HTTP authentication with the devices. Default value is 'Admin'
Password	String	Password that ARM uses for HTTP authentication with the devices. Default value is 'Admin'
sessionTimeout	Integer	Client session timeout (in seconds). Default value is Default value is 300
inactivityPeriod	Integer	Client inactivity period (in seconds). Default value is 300
httpHttpsEnabled	Boolean	Indicates whether mixed mode (HTTP+HTTPS) is enabled or disabled.
configuratorIpAddress	String	The IP address of the configurator
configuratorHostname	String	The host name of the configurator
IsConfCommunicationIpBased	Boolean	Indicates whether ARM should publish the host name or the IP address.
verifyCertificate	Boolean	Indicates whether the configurator should verify the server certificate before initiating the TLS connection

Attribute	Type	Description
verifyCertificateSubjectName	Boolean	Indicates whether the configurator should verify the server's subject name before initiating the TLS connection.

Note: Security configuration is only available for the Security administrator (Get/ Update).

HTTP Response

- `200 OK`
- `409 Conflict`
- `500 Internal error`

20.5 Quality-based Routing Configuration

20.5.1 Quality-based Routing Configuration

The `<ARM_Configurator_IP>/ARM/v1/configuration/quality` URL when used with the PUT method, provides the ability for the ARM Manager to send a request to the ARM Configurator to update the Quality-based routing configuration.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/configuration/quality
```

HTTP Method

`PUT`

Supported Content-Type

`application/json`

Attributes

Attribute	Type	Description
samplingTime	Integer	The rate in seconds in which the nodes send quality status updates This attribute also controls the rate in which the topology updates the routers regarding the quality changes Default 60 seconds
isQualityEnabled	Boolean	Is the feature enabled or not

Attribute	Type	Description

“ASR”:{
lowerThreshold: Float
upperThreshold: Float
isUsed:Boolean
}
	-	ASR value:
		• Lower Threshold
		• Upper Threshold
		• If Used

“MOS”:{
lowerThreshold: Float
upperThreshold: Float
isUsed:Boolean
}
		Mos value:
		• Lower Threshold
		• Upper Threshold
		• If Used

HTTP Response

- `200 OK`
- `409 Conflict`
- `500 Internal error`

20.5.2 Quality-based Routing Update Configuration

The `<ARM_Configurator_IP>/ARM/v1/configuration/quality` URL when used with the GET method, provides the ability for the ARM Manager to send a request to the ARM Configurator to retrieve the Quality-based routing configuration.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/configuration/quality
```

HTTP Method

`GET`

Supported Content-Type

`application/json`

Attributes

Attribute		Type	Description
samplingTime		Integer	The rate in seconds in which the nodes send quality status updates This attribute also controls the rate in which the topology updates the routers regarding the quality changes

Attribute		Type	Description
			Default 60 seconds
isQualityEnabled		Boolean	Is the feature enabled or not

“ASR”:{
lowerThreshold: Float
upperThreshold: Float
isUsed:Boolean
}
			ASR value:
			• Lower Threshold
			• Upper Threshold
			• If Used

“MOS”:{
lowerThreshold: Float
upperThreshold: Float
isUsed:Boolean
}
			Mos value:
			• Lower Threshold
			• Upper Threshold
			• If Used

HTTP Response

- `200 OK`
- `409 Conflict`
- `500 Internal error`

20.6 LDAP Authentication

20.6.1 Get LDAP Authentication Server

The `<ARM_Configurator_IP>/ARM/v1/configuration/security/ldap` URL when used with the GET method, provides the ability for the ARM Manager to send a request to the ARM Configurator to retrieve the LDAP Authentication Server attributes.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/configuration/security/ldap
```

HTTP Method

```
GET
```

Supported Content-Type

```
application/json
```

Attributes

Attribute	Type	Description
baseObject	String	The base object from which the search will be conducted
dn	String	The user DN to do the initial binding to the server
isEnabled	Boolean	Whether the LDAP authentication feature is activated
isSSL	Boolean	Should LDAPS be used
port	Integer	LDAP server port number
host	String	LDAP server host name

“parsedCertificate”:{
issuer: String
notAfter: String
notBefore: String
subject:String
validity:String
}
		The parsed certificate if such certificate exists

Attribute	Type	Description

“mappings”:{
adminMapping: String
permissionsAttribute: String
securityAdminMapping: String
userNameAttribute:String
}
		The mapping for each permission lever (Security admin and Admin).
		The field that the permission will be checked against and the field for the username.

HTTP Response

- `200 OK`
- `409 Conflict`
- `500 Internal error`

20.6.2 Update/Add LDAP Authentication Server

The `<ARM_Configurator_IP>/ARM/v1/configuration/security/ldap` URL when used with the PUT method, provides the ability for the ARM Manager to send a request to the ARM Configurator to Update/Add the LDAP Authentication Server attributes.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/configuration/security/ldap
```

HTTP Method

`PUT`

Supported Content-Type

`application/json`

Attributes

Attribute	Type	Description
baseObject	String	The base object from which the search will be conducted
dn	String	The user DN to do the initial binding to the server
isEnabled	Boolean	Whether the LDAP authentication feature is activated
isSSL	Boolean	Should LDAPS be used
port	Integer	LDAP server port number

Attribute	Type	Description
host	String	LDAP server host name
certificate		Certificate if LDAPS is sued

“mappings”:{
adminMapping: String
permissionsAttribute: String
securityAdminMapping: String
userNameAttribute:String
}
		The mapping for each permission lever (Security admin and Admin)
		The field that the permission will be checked against and the field for the username

password	String	The password for the initial binding

HTTP Response

- `200 OK`
- `409 Conflict`
- `500 Internal error`

20.6.3 Test LDAP Authentication Server

The `<ARM_Configurator_IP>/ARM/v1/configuration/security/ldap` URL when used with the PUT method, provides the ability for the ARM Manager to send a request to the ARM Configurator to test the LDAP Authentication Server.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/configuration/security/ldap/test
```

HTTP Method

`PUT`

Supported Content-Type

`application/json`

Attributes

Attribute	Type	Description
baseObject	String	The base object from which the search will be conducted
dn	String	The user DN to do the initial binding to the server
isEnabled	Boolean	Whether the LDAP authentication feature is activated
isSSL	Boolean	Should LDAPS be used
port	Integer	LDAP server port number
host	String	LDAP server host name
certificate		Certificate if LDAPS is sued

“mappings”:{
adminMapping: String
permissionsAttribute: String
securityAdminMapping: String
userNameAttribute:String
}
		The mapping for each permission lever (Security admin and Admin)
		The field that the permission will be checked against and the field for the username

passsword	String	The password for the initial binding

Attribute	Type	Description

testConnectivity:{
name:String
password:String
}
		The user name and the password to check against the LDAP server

HTTP Response

- 200 OK
- 409 Conflict
- 500 Internal error

20.7 RADIUS Authentication

20.7.1 Get RADIUS Authentication Server

The <ARM_Configurator_IP>/ARM/v1/configuration/security/ldap URL when used with the GET method, provides the ability for the ARM Manager to send a request to the ARM Configurator to retrieve the RADIUS Authentication Server attributes.

REST Resource

<ARM_Configurator_IP>/ARM/v1/configuration/security/radius

HTTP Method

GET

Supported Content-Type

application/json

Attributes

Attribute	Type	Description
defaultRole	ENUM	SECURITY_ADMIN,ADMIN,MONITOR,REJECT The default authentication level if the RADIUS doesn't have audiocodes dictionary but the user was found in the RADIUS
isEnabled	Boolean	Whether the RADIUS authentication feature is activated
port	Integer	The pot number
host	String	The host / IP address of the RADIUS server

Attribute	Type	Description
retryCount	Integer	The number of retries in case of the failure
sharedSecret	String	The password of the RADIUS server
timeout	Integer	The connection timeout

HTTP Response

- `200 OK`
- `409 Conflict`
- `500 Internal error`

20.7.2 Update/Add RADIUS Authentication Server

The `<ARM_Configurator_IP>/ARM/v1/configuration/security/radius` URL when used with the PUT method, provides the ability for the ARM Manager to send a request to the ARM Configurator to update or add the RADIUS Authentication Server attributes.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/configuration/security/radius
```

HTTP Method

`PUT`

Supported Content-Type

`application/json`

Attributes

Attribute	Type	Description
defaultRole	ENUM	SECURITY_ADMIN,ADMIN,MONITOR,REJECT The default authentication level if the RADIUS doesn't have audiocodes dictionary but the user was found in the RADIUS
isEnabled	Boolean	Whether the RADIUS authentication feature is activated
port	Integer	The pot number
host	String	The host / IP address of the RADIUS server
retryCount	Integer	The number of retries in case of the failure

Attribute	Type	Description
sharedSecret	String	The password of the RADIUS server
timeout	Integer	The connection timeout

HTTP Response

- `200 OK`
- `409 Conflict`
- `500 Internal error`

20.7.3 Test RADIUS Authentication Server

The `<ARM_Configurator_IP>/ARM/v1/configuration/security/radius` URL when used with the PUT method, provides the ability for the ARM Manager to send a request to the ARM Configurator to test the RADIUS Authentication Server.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/configuration/security/radius/test
```

HTTP Method

`PUT`

Supported Content-Type

`application/json`

Attributes

Attribute	Type	Description
defaultRole	ENUM	SECURITY_ADMIN,ADMIN,MONITOR,REJECT The default authentication level if the RADIUS doesn't have an AudioCodes dictionary; however, the user was found on the RADIUS server.
isEnabled	Boolean	Whether the RADIUS authentication feature is activated
port	Integer	The port number
host	String	The host / IP address of the RADIUS server
retryCount	Integer	The number of retries in case of the failure

Attribute	Type	Description
sharedSecret	String	The password of the RADIUS server
timeout	Integer	The connection timeout

testConnectivity:{
name:String
password:String
}
		The user name and the password to check against the RADIUS server

HTTP Response

- `200 OK`
- `409 Conflict`
- `500 Internal error`

20.8 Calls Configuration

20.8.1 Get Calls Configuration

The `<ARM_Configurator_IP>/ARM/v1/configuration/calls` URL when used with the GET method, provides the ability for the ARM Manager to send a request to the ARM Configurator to retrieve the Calls configuration.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/configuration/calls
```

HTTP Method

`GET`

Supported Content-Type

`application/json`

Attributes

Attribute	Type	Description
limit	Integer	The maximum number of calls that will be saved in the database. Maximum value is 10,000,000

Attribute	Type	Description
enabled	Boolean	Indicates whether calls will be sent by the routers and whether they will be saved in database
threshold	Integer	Indicates the time after which an open call is considered closed
includeCdrInSuccessfulCalls	Boolean	Should the entire CDR chain be saved for successful calls
includeCdrInPartialCalls	Boolean	Should the entire CDR chain be saved for partial calls

HTTP Response

- `200 OK`
- `409 Conflict`
- `500 Internal error`

20.8.2 Update Calls Configuration

The `<ARM_Configurator_IP>/ARM/v1/configuration/calls` URL when used with the PUT method, provides the ability for the ARM Manager to send a request to the ARM Configurator to update the Calls configuration.

REST Resource

```
<ARM_Configurator_IP>/ARM/ v1/configuration/calls
```

HTTP Method

`PUT`

Supported Content-Type

`application/json`

Attributes

Attribute	Type	Description
limit	Integer	The maximum number of calls that will be saved in the database. Maximum value is 10,000,000

Attribute	Type	Description
enabled	Boolean	Indicates whether calls will be sent by the routers and whether they will be saved in database
threshold	Integer	Indicates the time after which an open call is considered closed
includeCdrInSuccessfulCalls	Boolean	Should the entire CDR chain be saved for successful calls
includeCdrInPartialCalls	Boolean	Should the entire CDR chain be saved for partial calls

HTTP Response

- `200 OK`
- `409 Conflict`
- `500 Internal error`

20.9 Registered Users

20.9.1 Get Registered Users Configuration

The `<ARM_Configurator_IP>/ARM/v1/configuration/registeredUsers` URL when used with the GET method, provides the ability for the ARM Manager to send a request to the ARM Configurator to retrieve the registered users configuration.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/configuration/registeredUsers
```

HTTP Method

`GET`

Supported Content-Type

`application/json`

Attributes

Attribute	Type	Description
enabled	Boolean	Enable registered users feature

20.9.2 Update Registered Users Configuration

The `<ARM_Configurator_IP>/ARM/v1/configuration/registeredUsers` URL when used with the PUT method, provides the ability for the ARM Manager to send a request to the ARM Configurator to update the registered users configuration.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/configuration/registeredUsers
```

HTTP Method

`PUT`

Supported Content-Type

`application/json`

Attributes

Attribute	Type	Description
enabled	Boolean	Enable registered users feature

20.10 Analytics

20.10.1 Get Analytics Configuration

The `<ARM_Configurator_IP>/ARM/v1/configuration/analytics` URL when used with the GET method, provides the ability for the ARM Manager to send a request to the ARM Configurator to retrieve the Analytics configuration.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/configuration/analytics
```

HTTP Method

`GET`

Supported Content-Type

`application/json`

Attributes

Attribute	Type	Description
password	String	Analytics user password. This password can be any string which doesn't contain the following characters: , ; \ '
host1	String	IP / hostname 1 for connecting to the Analytics database using default user 'analytics'. Default='%' you can connect using the analytics user from any IP / hostname. For example, " 'analytics'@'%' "; " 'analytics'@'ABC' "
host2	String	IP / hostname 2 for connecting to the Analytics database using default user 'analytics'. Default='%'
host3	String	IP / hostname 3 for connecting to the Analytics database using default user 'analytics'.. Default='%'

20.10.2 Update Analytics Configuration

The `<ARM_Configurator_IP>/ARM/v1/configuration/analytics` URL when used with the PUT method, provides the ability for the ARM Manager to send a request to the ARM Configurator to update the Analytics configuration.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/configuration/analytics
```

HTTP Method

```
PUT
```

Supported Content-Type

```
application/json
```

Attributes

Attribute	Type	Description
password	String	Analytics user password.

Attribute	Type	Description
		This password can be any string which doesn't contain the following characters: , ; \ '
host1	String	IP / hostname 1 for connecting to the Analytics database using default user 'analytics'. Default='%' you can connect using the analytics user from any IP / hostname. For example, " 'analytics'@'%' "; " 'analytics'@'ABC' "
host2	String	IP / hostname 2 for connecting to the Analytics database using default user 'analytics'. Default='%'
host3	String	IP / hostname 3 for connecting to the Analytics database using default user 'analytics'. Default='%'

20.10.3 Restore Analytics to Default

The `<ARM_Configurator_IP>/ARM/v1/configuration/analytics/default` URL when used with the PUT method, provides the ability for the ARM Manager to send a request to the ARM Configurator to restore the Analytics configuration to default.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/configuration/analytics/default
```

HTTP Method

```
PUT
```

Supported Content-Type

```
application/json
```

20.10.4 Get Analytics Users

The `<ARM_Configurator_IP>/ARM/v1/configuration/analytics/users` URL when used with the GET method, provides the ability for the ARM Manager to send a request to the ARM Configurator to retrieve a list of Analytics users.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/configuration/analytics/users
```

HTTP Method

GET

Supported Content-Type

application/json

Attributes

Attribute	Type	Description
[String]	["'analytics'@'%'", "'analytics'@'ABCCorp'", "'analytics'@'VoIP'"]	User-defined analytics users

20.11 Registration

20.11.1 Get Registration

The <ARM_Configurator_IP>/ARM/v1/configuration/registration URL when used with the GET method, provides the ability for the ARM Manager to send a request to the ARM Configurator to retrieve the registration configuration.

REST Resource

<ARM_Configurator_IP>/ARM/v1/configuration/registration

HTTP Method

GET

Supported Content-Type

application/json

Attributes

Attribute	Type	Description
registrationPropertyId	Integer	The ID of the dictionary property which should be mapped to one of the following: ▪ **True:** ARM Routers route Register SIP messages for the registered user. ▪ **False:** User is not eligible for the Register SIP message routing.

Attribute	Type	Description
dailablePropertyIds	[Integer]	List of dictionary property ids which are mapped to the dialable / address of the user. Used as the match for the user in the “To” SIP Header of the SIP Register message.
Enabled	Boolean	Indicates whether the routing of registration messages should be enabled.

20.11.2 Update Registration

The `<ARM_Configurator_IP>/ARM/v1/configuration/registration` URL when used with the PUT method, provides the ability for the ARM Manager to send a request to the ARM Configurator to update the registration configuration.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/configuration/registration
```

HTTP Method

```
PUT
```

Supported Content-Type

```
application/json
```

Attributes

Attribute	Type	Description
registrationPropertyId	Integer	The ID of the dictionary property “registration” which should be mapped to one of the following: • **True:** ARM Routers route Register SIP messages for the registered user. • **False:** User is not eligible for the Register SIP message routing.
dailablePropertyIds	[Integer]	List of dictionary property IDs which are mapped to the dialable / address of the user. Used as the match for the user in the “To” SIP Header of the SIP Register message.

Attribute	Type	Description
Enabled	Boolean	Indicates whether the routing of registration messages should be enabled for the user. Equivalent to “Destination is a registered user in ARM” in the routing rule in the ARM Web interface.

20.11.3 Get Registration Users

The `<ARM_Configurator_IP>/ARM/v1/configuration/registration` URL when used with the GET method, provides the ability for the ARM Manager to send a request to the ARM Configurator to retrieve the list of registered users.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/configuration/registration
```

HTTP Method

```
GET
```

Supported Content-Type

```
application/json
```

Attributes

Attribute	Type	Description
[String]		Name of registered user.

This page is intentionally left blank.

21 Calls

21.1 Get Calls Summary

The `<ARM_Configurator_IP>/ARM/v1/configuration/calls` URL when used with the GET method, provides the ability for the ARM Manager to send a request to the ARM Configurator to retrieve a summary of the Calls configuration.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/calls
```

HTTP Method

`GET`

Supported Content-Type

`application/json`

Attributes

Attribute	Type	Description
cursor	Integer	The next page or -1 if it is the last page

calls	{
	sessionKey: String
	sessionId:String
	armSetupTime: Date
	srcUriBeforeManipulation:String
	dstUriBeforeManipulation:String
	incomingNodeId:Integer
	incomingPconId:Integer
	outgoingNodeId:Integer
	outgoingPconId:Integer
	routingRuleId:Integer
	sipTerminationReason:String
	}
		Call details

HTTP Response

- `200 OK`
- `409 Conflict`
- `500 Internal error`

21.1.1 Filter

The Statistics API requires using a filter to control the size and content of the data. The filter should be added to the suffix of the URL as is shown in the example below:

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/calls?
filter=(incomingNodeId={id},outgoingNodeId={id})
```

HTTP Method

GET

Supported Content-Type

application/json

Attributes

Filter	Available for these Requests	Description
incomingNodeId	calls	Equals (=)
outgoingNodeId	calls	Equals (=)
incommingPconId	calls	Equals (=)
outgoingPconId	calls	Equals (=)
startTime	calls	Equals (=)
endTime	calls	Equals (=)
dstUriBeforeManipulation	calls	Contains (~)
srcUriBeforeManipulation	calls	Contains (~)
sessionId	calls	Equals (=)
routingRuleId	calls	Equals (=)
sipTerminationReason	calls	Equals (=)

HTTP Response

- 200 OK
- 409 Conflict
- 500 Internal error

21.2 Get Specific Call

The `<ARM_Configurator_IP>/ARM/v1/calls/{sessionKey}` URL when used with the GET method, provides the ability for the ARM Manager to send a request to the ARM Configurator to retrieve the details of a specific call.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/calls/{sessionKey}
```

HTTP Method

`GET`

Supported Content-Type

`application/json`

Attributes

Attribute	Type	Description
sessionKey	String	Unique combination of node id and sessionId
sessionId	String	The session Id generated by the Node
armSetupTime	Date	UTC time
armReleaseTime	Date	UTC time
routerIp	String	The IP address of the initial router that calculated the path.
srcUriBeforeManipulation	String	The source URI as it was prior to manipulation.
dstUriBeforeManipulation	String	The destination URI as it was prior to manipulation.
discardingByRoutingRule	Integer	The routing rule id of the routing rule which case the call to be discarded.
routingRuleId	Integer	The routing rule id of the successful path, or the last path if the call failed.
partial	Boolean	Indicates whether the call was ended with all the CDR information, or if some of the data was missing.
description	String	Description regarding the call (if any).
incomingNodeId	Integer	The id of the incoming node.

Attribute	Type	Description
incommingPconId	Integer	The id of the incoming peer connection.
incommingConnectionId	Integer	The id of the incoming connection (only relevant to 3rd party nodes).
outgoingNodeId	Integer	The id of the last node in the paths chain.
outgoingPconId	Integer	The id of the last peer connection in the paths chain.
sipTerminationReason	String	The SIP response code.
voiceDuration	Integer	Duration of the voice call in milliseconds.

preRouteManipulationsList	{
	originalValue:String
	manipulatedValue:String
	manipulatedEntity:ENUM
	manipulationsList:{
	originalValue: String
	manipulatedValue: String
	manipulatingEntity:ENUM
	manipulatingEntityName: String
	isChanged: Boolean
	description: String
	}
	}
		• ManipulatingEntity (optional)- the entity from ARM data base/GUI perspective which tracked and might change the ManipulatedEntity.
		• ManipulatedEntity (optional)- the entity that is tracked and may change.
		• originalValue: the value of the manipulated entity prior to manipulation (in the case of user/password/destIP), these entities may be optional as well.
		Note: When the ManipulatingEntity is a 'RoutingRuleAction', there is no URL.

Attribute	Type	Description

cdrSessionPathList	{
	sipTerminationReason: String
	sipTerminationReasonDesc: String
	routeSeq: Integer
	routerIp: String
	srcUriAfterManipulation: String
	dstUriAfterManipulation: String
	armSetupTime: Date
	armReleaseTime: Date
	sbcSetupTime: Date
	sbcConnectTime: Date
	sbcReleaseTime: Date
	sbcAlertTime: Date
	alertDuration: Integer
	voiceDuration: Integer
	completeDuration: Integer
	}
		Represents a path of a call. Iif the call failed without starting this object, this field is Null.
		Durations are -1 if the path fails

HTTP Response

- `200 OK`
- `409 Conflict`
- `500 Internal error`

This page is intentionally left blank.

22 WEB Services

This chapter describes the REST API resources for Web Services. The following APIs are described:

- Get Web Services Templates
- Get Web Services Implementation
- Create Web Service Implementation
- Update Web Service Implementation
- Delete Web Service Implementation

22.1 Get Web Services Templates

`The <ARM_Configurator_IP>/ARM/v1/numberPortability/template`

URL when used with the GET method, provides the ability for the ARM Manager to send a request to the ARM Configurator to retrieve a Web Services template.

REST Resource

`<ARM_Configurator_IP>/ARM/v1/numberPortability/template`

The Template defines all the attributes that the operator will be able to define for a specific web service.

HTTP Method

`GET`

Supported Content-Type

`application/json`

Attributes

Attribute	Type	Description
id	Integer	Id of the Web services template
name	String	Name of the Web services template.
numberPortabilityAgentType	Enum	The unique id of the template type.

params	[
	{
	name: String
	uiName: String
	value: String
	type:enum
	}
	]
		Name – the database of the paramters in the database
		uiName- the name of the attribute in the UI
		value – the default value
		type - stringType(0), numberType(1), booleanType(2);

HTTP Response

- 200 OK
- 409 Conflict
- 500 Internal error

22.2 Get Web Services Implementation

The <ARM_Configurator_IP>/ARM/v1/numberPortability/implementation URL when used with the GET method, provides the ability for the ARM Manager to send a request to the ARM Configurator to retrieve a specific Web Services Implementation.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/numberPortability/implementation
<ARM_Configurator_IP>/ARM/v1/numberPortability/implementation/{id}
```

HTTP Method

GET

Supported Content-Type

application/json

Attributes

Implementation of one of the templates

Attribute	Type	Description
id	Integer	Id of the Web services template
name	String	The name of the Web Service Implementation.
numberPortabilityAgentType	Enum	The unique id of the template type.

params	[
	{
	name: String
	uiName: String
	value: String
	type:enum
	}
	]
		• Name – the database of the attributes in the database
		• uiName- the name of the attribute in the UI
		• value – the default value
		• type
		✓ stringType(0),
		✓ numberType(1),
		✓ booleanType(2)

HTTP Response

- 200 OK
- 409 Conflict
- 500 Internal error

22.3 Create Web Service Implementation

The ARM_Configurator_IP>/ARM/v1/numberPortability/implementation URL when used with the POST method, provides the ability for the ARM Manager to send a request to the ARM Configurator to add a Web Service Implementation.

REST Resource

<ARM_Configurator_IP>/ARM/v1/numberPortability/implementation

HTTP Method

POST

Supported Content-Type

application/json

Attributes

Attribute	Type	Description
id	Integer	The id of the Web Service Implementation
name	String	The name of the Web Service Implementation.
numberPortabilityAgentType	Enum	The unique id of the template type.

params	[
	{
	name: String
	uiName: String
	value: String
	type:enum
	}
	]
		▪ Name – the database of the attributes in the database
		▪ uiName- the name of the attribute in the UI
		▪ value – the default value
		▪ type - stringType(0), numberType(1), booleanType(2);

HTTP Response

- 201 Created
- 409 Conflict
- 500 Internal error

22.4 Update Web Service Implementation

The `<ARM_Configurator_IP>/ARM/v1/numberPortability/implementation/{id}` URL when used with the POST method, provides the ability for the ARM Manager to send a request to the ARM Configurator to update a Web Service Implementation.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/numberPortability/implementation/{id}
```

HTTP Method

```
PUT
```

Supported Content-Type

```
application/json
```

Attributes

Attribute	Type	Description
id	Integer	The id of the Web Service Implementation.
name	String	The name of the Web Service Implementation.
numberPortabilityAgentType	Enum	The unique id of the template type.

params	[
	{
	name: String
	uiName: String
	value: String
	type:enum
	}
	]
		▪ Name – the database of the paramtere in the databse
		▪ uiName- the name of the attribute in the UI
		▪ value – the default value
		▪ type - stringType(0),
		✓ numberType(1)
		✓ booleanType(2);

HTTP Response

- `200 OK`
- `409 Conflict`
- `500 Internal error`

22.5 Delete Web Service Implementation

The `<ARM_Configurator_IP>/ARM/v1/numberPortability/implementation/{id}` URL when used with the DELETE method, provides the ability for the ARM Manager to send a request to the ARM Configurator to delete a Web Service Implementation.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/numberPortability/implementation/{id}
```

HTTP Method

`DELETE`

Supported Content-Type

`application/json`

HTTP Response

- `200 OK`
- `409 Conflict`
- `500 Internal error`

This page is intentionally left blank.

23 Certificates

The URLs describes in this section can be implemented to manage the SSL connection for the ARM SSL client for authenticating the server certificate of a remote server (one-way authentication).

23.1 Get Certificate API

The `<ARM_Configurator_IP>/ARM/v1/certificate` URL when used with the GET method, provides the ability for the ARM Manager to send a request to the ARM Configurator to retrieve a remote server certificate or root certificate from the Trusted Root Store.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/certificate
```

HTTP Method

`GET`

Supported Content-Type

`application/json`

Attribute	Type	Description
alias	String	The name of the alias of the certificate
value	String	Base64 representation of the certificate

HTTP Response

- `200 OK`
- `409 Conflict`
- `500 Internal error`

23.2 Add Certificate API

The `<ARM_Configurator_IP>/ARM/v1/certificate` URL when used with the POST method, provides the ability for the ARM Manager to send a request to the ARM Configurator to upload a remote server certificate or root certificate to the Trusted Root Store.

REST Resource

```
POST <ARM_Configurator_IP>/ARM/v1/certificate
```

HTTP Method

`POST`

Supported Content-Type

`application/json`

Attribute	Type	Description
alias	String	The name of the alias of the certificate
value	String	Base64 representation of the certificate

HTTP Response

- `201 Created`
- `409 Conflict`
- `500 Internal error`

23.3 Delete Certificate API

The `<ARM_Configurator_IP>/ARM/v1/certificate/{alias}` URL when used with the DELETE method, provides the ability for the ARM Manager to send a request to the ARM Configurator to delete a remote server certificate or root certificate from the Trusted Root Store.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/certificate/{alias}
```

HTTP Method

`DELETE`

Supported Content-Type

`application/json`

HTTP Response

- `200 OK`
- `409 Conflict`
- `500 Internal error`

24 Policy Studio

This chapter describes the REST API resources for the Policy Studio. The following APIs are described:

- Get Policy Studio API
- Add Policy Studio API
- Update Policy Studio API
- Delete Policy Studio API

24.1 Get Policy Studio API

The `<ARM_Configurator_IP>/ARM/v1/callSetupRule` URL when used with the GET method, provides the ability for the ARM Manager to send a request to the ARM Configurator to retrieve a call setup rule.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/callSetupRule
<ARM_Configurator_IP>/ARM/v1/callSetupRule/{id}
```

HTTP Method

`GET`

Supported Content-Type

`application/json`

Attribute	Type	Description
id	Integer	Database ID
name	String	Name of the policy studio rule.

searchFields	{
	field: enum
	propertyDicEntryIds :[Integer]
	
	}
		Possible values for the field:
		SOURCE_URI_USER, SOURCE_URI_HOST, DEST_URI_USER, DEST_URI_HOST, CONTACT_URI_USER, CONTACT_URI_HOST, CONTACT_URI_PORT, P_ASSERTED_IDENTITY_DISPLAY_NAME, P_ASSERTED_IDENTITY_USER P_ASSERTED_IDENTITY_HOST
		Each represents SIP header to be matched

actionFields	{
	field: enum
	propertyDicEntryIds :[Integer]
	}
		Possible values for the field:
		
		SOURCE_URI_USER, SOURCE_URI_HOST, DEST_URI_USER, DEST_URI_HOST, DEST_IP_ADDR, DEST_PORT, DEST_PROTOCOL, USER_CREDENTIALS_USER_NAME, USER_CREDENTIALS_PASSWORD, P_ASSERTED_IDENTITY_DISPLAY_NAM

Attribute	Type	Description
		E, P_ASSERTED_IDENTITY_USER, P_ASSERTED_IDENTITY_HOST Each represents SIP header to be changed
nodeIds	[Integer]	Nodes to be matched
pConIds	[Integer]	Peer connections to be matched
topologyGroupIds	[Integer]	Topology Groups to be matched
priority	Integer	The priority of the policy studio rule
adminState	Enum	LOCKED, UNLOCKED

destAttr	{
	attributes: {
	value: String
	}
	attributeGroups:[Integer]
	}
		Destination attribute to match (destination number), can be either prefix, or prefix group id

type	enum	USER, NUMBER_PORTABILITY User is the regular policy studio Number portability is the web services
numberPortabilityId	Integer	If the type is NUMBER_PORTABILITY,ID of the web service which will be done in case of a match is provided.
isRegisteredUsers	Boolean	Enables the Registered Users feature for operators to route calls to SBC registered users.
destUserGroupsIds	[Integer]	Destination User Group IDs
srcUserGroupsIds	[Integer]	Source User Group IDs
requestType	Enum	CALL (0) REGISTER (1)
didMatchType	Enum	SRC_AND_DEST, DEST, SOURCE This attribute is only used for DID masking web services.
poolId	Integer	The id of the “pool of numbers” prefix group, • If defined: ARM Router chooses a free random number and manipulates the source number, • If not defined: ARM saves the source URI and destination URI in the database without manipulating (masking) the source number. This attribute is only used for DID masking web services.
callExpirationTime	Integer	Timeout before the call is unavailable for callback (in seconds). This attribute is only used for DID masking web services.

Attribute	Type	Description
didXHeader	String	If defined, the ARM router does not manipulate the destination number for the incoming call, instead it adds it as the defined X Header; the header must start with the X-ARM- prefix. This attribute is only used for DID masking web services.
srcNormalizationGroupId	Integer	ID of the normalization group. This normalization is performed on the source URI prior to the database lookup / persistence. This manipulation does not affect the routing. This attribute is only used for DID masking web services.
setDestNormalizationGroupId	Integer	ID of the normalization group. This normalization is performed on the destination URI prior to the database lookup / persistence. This manipulation does not affect the routing. This attribute is only used for DID masking web services.
direction	Enum	Call direction: • OUTGOING • INCOMING This attribute is only used for DID masking web services.
flow	Enum	STOP, CONTINUE, CONTINUE_TO_RULE
continueToRuleId	Integer	ID of the policy studio. This attribute is only relevant when the flow attribute is set to CONTINUE_TO_RULE.

HTTP Response

- `200 OK`
- `409 Conflict`
- `500 Internal error`

24.2 Add Policy Studio API

The `<ARM_Configurator_IP>/ARM/v1/callSetupRule` URL when used with the POST method, provides the ability for the ARM Manager to send a request to the ARM Configurator to add a Policy Studio generated call setup rule.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/callSetupRule
```

HTTP Method

POST

Supported Content-Type

application/json

Attribute	Type	Description
id	Integer	Database ID
name	String	Name of the policy studio rule

searchFields	{
	field: enum
	propertyDicEntryIds :[Integer]
	
	}
		Possible values for the field:
		SOURCE_URI_USER, SOURCE_URI_HOST, DEST_URI_USER, DEST_URI_HOST, CONTACT_URI_USER, CONTACT_URI_HOST, CONTACT_URI_PORT, P_ASSERTED_IDENTITY_DISPLAY_NAME, P_ASSERTED_IDENTITY_USER P_ASSERTED_IDENTITY_HOST
		Each represents SIP header to be matched

actionFields	{
	field: enum
	propertyDicEntryIds :[Integer]
	}
		Possible values for the field:
		
		SOURCE_URI_USER, SOURCE_URI_HOST, DEST_URI_USER, DEST_URI_HOST, DEST_IP_ADDR, DEST_PORT, DEST_PROTOCOL, USER_CREDENTIALS_USER_NAME, USER_CREDENTIALS_PASSWORD, P_ASSERTED_IDENTITY_DISPLAY_NAME, P_ASSERTED_IDENTITY_USER, P_ASSERTED_IDENTITY_HOST
		Each represents SIP header to be changed

nodeIds	[Integer]	Nodes to be matched
pConIds	[Integer]	Peer connections to be matched
priority	Integer	The priority of the policy studio rule
prevId	Integer	The id of the previous policy studio rule, -1 if first
nextId	Integer	The id of the next policy studio rule, -1 if last
adminState	Enum	LOCKED, UNLOCKED

destAttr	{
	attributes: {
	value: String
	}
	attributeGroups:[Integer]
	}
		Destination attribute to match (destination number), can be either prefix, or prefix group id

Attribute	Type	Description
type	enum	USER, NUMBER_PORTABILITY User is the regular policy studio Number portability is the web services
numberPortabilityId	Integer	If the type is NUMBER_PORTABILITY, ID of the Web service which will be done in case of a match must be provided
isRegisteredUsers	Boolean	Enables the Registered Users feature for operators to route calls to SBC registered users (“Destination is a registered user in ARM” check box in ARM GUI).
destUserGroupsIds	[Integer]	The list of Destination User Group IDs
srcUserGroupsIds	[Integer]	The list of Source Group IDs
requestType	Enum	CALL (0) REGISTER (1)
didMatchType	Enum	SRC_AND_DEST, DEST, SOURCE This attribute is only used for DID masking web services.
poolId	Integer	This attribute is only used for DID masking web services.
callExpirationTime	Integer	This attribute is only used for DID masking web services.
didXHeader	Integer	This attribute is only used for DID masking web services.
srcNormalizationGroupId	Integer	This attribute is only used for DID masking web services.
setDestNormalizationGroupId	Integer	This attribute is only used for DID masking web services.
direction	Enum	This attribute is only used for DID masking web services. OUTGOING, INCOMING
flow	Enum	This attribute is only used for DID masking web services. STOP, CONTINUE, CONTINUE_TO_RULE
continueToRuleId	Integer	This attribute is only used for DID masking web services.

HTTP Response

- `201 Created`
- `409 Conflict`
- `500 Internal error`

24.3 Update Policy Studio API

The `<ARM_Configurator_IP>/ARM/v1/callSetupRule` URL when used with the PUT method, provides the ability for the ARM Manager to send a request to the ARM Configurator to update a Policy Studio generated call setup rule.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/callSetupRule/{id}
```

HTTP Method

`PUT`

Supported Content-Type

`application/json`

Attributes

Attribute	Type	Description
id	Integer	Database ID
name	String	Name of Policy Studio rule.

searchFields	{
	field: enum
	propertyDicEntryIds :[Integer]
	}
	}
		Possible values for the field:
		SOURCE_URI_USER, SOURCE_URI_HOST, DEST_URI_USER, DEST_URI_HOST, CONTACT_URI_USER, CONTACT_URI_HOST, CONTACT_URI_PORT, P_ASSERTED_IDENTITY_DISPLAY_NAME, P_ASSERTED_IDENTITY_USER P_ASSERTED_IDENTITY_HOST
		Each represents SIP header to be matched

actionFields	{
	field: enum
	propertyDicEntryIds :[Integer]
	}
		Possible values for the field:
		SOURCE_URI_USER, SOURCE_URI_HOST, DEST_URI_USER, DEST_URI_HOST, DEST_IP_ADDR, DEST_PORT, DEST_PROTOCOL, USER_CREDENTIALS_USER_NAME, USER_CREDENTIALS_PASSWORD, P_ASSERTED_IDENTITY_DISPLAY_NAME, P_ASSERTED_IDENTITY_USER, P_ASSERTED_IDENTITY_HOST
		Each represents SIP header to be changed

nodeIds	[Integer]	Nodes to be matched
pConIds	[Integer]	Peer connections to be matched
priority	Integer	The priority of the policy studio rule

Attribute	Type	Description
prevId	Integer	The id of the previous policy studio rule, -1 if first
nextId	Integer	The id of the next policy studio rule, -1 if last
adminState	Enum	LOCKED, UNLOCKED

destAttr	{
	attributes: {
	value: String
	}
	attributeGroups:[Integer]
	}
		Destination attribute to match (destination number), can be either prefix, or prefix group id

type	enum	USER, NUMBER_PORTABILITY User is the regular policy studio Number portability is the web services
numberPortabilityId	Integer	If the type is NUMBER_PORTABILITY, Id of th web service which will be done in case of a match must be provided
isRegisteredUsers	Boolean	Indicates whether the user is registered.
destUserGroupsIds	[Integer]	The list of Destination User Group IDs
srcUserGroupsIds	[Integer]	The list of Source User Group IDs
requestType	Enum	CALL (0) REGISTER (1)
didMatchType	Enum	SRC_AND_DEST, DEST, SOURCE This attribute is only used for DID masking web services.
poolId	Integer	Only for DID masking web services
callExpirationTime	Integer	Only for DID masking web services
didXHeader	Integer	This attribute is only used for DID masking web services.
srcNormalizationGroupId	Integer	This attribute is only used for DID masking web services.
setDestNormalizationGroupId	Integer	This attribute is only used for DID masking web services.
direction	Enum	This attribute is only for DID masking web services. OUTGOING, INCOMING
flow	Enum	This attribute is only used for DID masking web services. STOP, CONTINUE, CONTINUE_TO_RULE

Attribute	Type	Description
continueToRuleId	Integer	This attribute is only used for DID masking web services.

HTTP Response

- 200 OK
- 409 Conflict
- 500 Internal error

24.4 Delete Policy Studio API

The `<ARM_Configurator_IP>/ARM/v1/callSetupRule` URL when used with the DELETE method, provides the ability for the ARM Manager to send a request to the ARM Configurator to delete a Policy Studio generated call setup rule.

REST Resource

```
DELETE <ARM_Configurator_IP>/ARM/v1/callSetupRule/{id}
```

HTTP Response

- 200 OK
- 409 Conflict
- 500 Internal error

25 Journal

25.1 Get Journal

The `<ARM_Configurator_IP>/ARM/v1/journal` URL when used with the GET method, provides the ability for the ARM Manager to send a request to the ARM Configurator to retrieve pages of journal entries.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/journal
```

HTTP Method

`GET`

Supported Content-Type

`application/json`

Attributes

Attribute	Type	Description
totalPages	Integer	Total pages according to the limit selected
totalElements	Integer	Total number of journal entries
numberOfElementsInPage	Integer	Total of journal entries in the current page
pageNumber	Integer	Current page

elements	{
	Id: Integer
	source: String
	createDate: Date
	description: String
	value: String
	action: enum
	actionDescription: String
	}
		value – the JSON oject that was used to do the action, or null if if didn't have a body
		
		action – enum of the action
		
		actionDescription – textual description of the action

25.1.1 Filter

The Statistics API requires using a filter to control the size and content of the data. Filter should be added to the suffix of the URL, for example:

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/journal?
filter=(searchString={string},description={String})&limit=25
```

HTTP Method

GET

Supported Content-Type

application/json

Attributes

Filter	Available for these requests	Description
searchString	calls	Contains (~)
value	calls	Equals (=)
description	calls	Contains (~)
operator	calls	Equals (=) or Contains (~)
source	calls	Equals (=)
action	calls	Contains (~)
startDate	calls	Equals (=)
endDate	calls	Equals (=)
limit	calls	Equals (=)
page	calls	Equals (=)
sort	calls	DESC or ASC

HTTP Response

- 200 OK
- 409 Conflict
- 500 Internal error

25.1.2 Get Specific Journal Entry

The `<ARM_Configurator_IP>/ARM/v1/journal` URL when used with the GET method, provides the ability for the ARM Manager to send a request to the ARM Configurator to retrieve specific journal entries.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/journal/{id}
```

HTTP Method

`GET`

Supported Content-Type

`application/json`

Attributes

Attribute	Type	Description
Id	Integer	Database id
source	String	The source of the journal, generally ARM
createDate	Date	The creation date of the journal entry in UTC
description	String	Description of the action
Value	String	The JSON value if any of the body request
Action	ENUM	An action enumerator
actionDescription	String	Textual description of the action enumerator

HTTP Response

- `200 OK`
- `409 Conflict`
- `500 Internal error`

This page is intentionally left blank.

26 Logging

This chapter describes the REST resources for logging.

26.1 Client Logging into the Server

The `<ARM_Configurator_IP>/ARM/v1/logging/client` when used with the POST method, provides the ability for the ARM Manager to send a request to the ARM Configurator to send an error log message to a dedicated log file on the server.

REST Resource

```
<ARM_Configurator_IP>/ARM/v1/logging/client
```

HTTP Method

`POST`

Supported Content-Type

`application/json`

Attributes

The request MUST contain JSON data that consists of the following elements:

Attribute	Type	Description
message	String	The message to be logged to the log file

HTTP Response

- `201 Created` – The log message was created successfully to the log file.
- `409 Conflict`
- `500 Internal error`

This page is intentionally left blank.

27 REST Responses Codes

The following is a summary of the status codes for responses that will be used in the API:

- **200 OK** – Indicates that a successful request has been completed.
- **201 Created** – Indicates the successful creation of a new resource and corresponding JSON is sent.
- **202 Accepted** – Indicates that the request has been accepted for processing, however the processing has not yet been completed.
- **204 No Content Response** – a successful request has been completed without a response body (the result was empty).
- **401 Unauthorized** – Indicates that that the username or password of an existing WEB user is not correct.
- **403 Forbidden** – Indicates that the existing WEB user does not have privileges to execute this request (incorrect credentials are sent or if the role of the user didn't match the security administrator's credentials).
- **409 Conflict** – Indicates that there is a conflict in the request.
- **500 Internal Error** – the server encountered an internal error

International Headquarters
1 Hayarden Street,
Airport City
Lod 7019900, Israel
Tel: +972-3-976-4000
Fax: +972-3-976-4040

AudioCodes Inc.
200 Cottontail Lane
Suite A101E
Somerset NJ 08873
Tel: +1-732-469-0880
Fax: +1-732-469-2298

Contact us: https://www.audiocodes.com/corporate/offices-worldwide
Website: https://www.audiocodes.com/

Document #: LTRT-41964