



aXAPI Reference

AX Series Advanced Traffic Manager

Document No.: D-030-01-00-0010

aXAPI Ver. 2.0, AX Release 2.7.0 10/10/2012

© A10 Networks, Inc. 10/10/12 - All Rights Reserved

Information in this document is subject to change without notice.

Trademarks

A10 Networks, the A10 logo, aACI, aCloud, ACOS, aDCS, aDNS, aELB, aFlex, aFlow, aGalaxy, aPlatform, aUSG, aVCS, aWAF, aXAPI, IDAccess, IDSENTRIE, IP to ID, SmartFlow, SoftAX, Unified Service Gateway, Virtual Chassis, VirtualADC, and VirtualN are trademarks or registered trademarks of A10 Networks, Inc. All other trademarks are property of their respective owners.

Patents Protection

A10 Networks products including all AX Series products are protected by one or more of the following US patents and patents pending: 8291487, 8266235, 8151322, 8079077, 7979585, 7716378, 7675854, 7647635, 7552126, 20120216266, 20120204236, 20120179770, 20120144015, 20120084419, 20110239289, 20110093522, 20100235880, 20100217819, 20090049537, 20080229418, 20080148357, 20080109887, 20080040789, 20070283429, 20070282855, 20070271598, 20070195792, 20070180101

Confidentiality

This document contains confidential materials proprietary to A10 Networks, Inc. This document and information and ideas herein may not be disclosed, copied, reproduced or distributed to anyone outside A10 Networks, Inc. without prior written consent of A10 Networks, Inc. This information may contain forward looking statements and therefore is subject to change.

A10 Networks Inc. Software License and End User Agreement

Software for all AX Series products contains trade secrets of A10 Networks and its subsidiaries and Customer agrees to treat Software as confidential information.

Anyone who uses the Software does so only in compliance with the terms of this Agreement. Customer shall not:

- 1) reverse engineer, reverse compile, reverse de-assemble or otherwise translate the Software by any means
- 2) sublicense, rent or lease the Software.

Disclaimer

The information presented in this document describes the specific products noted and does not imply nor grant a guarantee of any technical performance nor does it provide cause for any eventual claims resulting from the use or misuse of the products described herein or errors and/or omissions. A10 Networks, Inc. reserves the right to make technical and other changes to their products and documents at any time and without prior notification.

No warranty is expressed or implied; including and not limited to warranties of non-infringement, regarding programs, circuitry, descriptions and illustrations herein.

Environmental Considerations

Some electronic components may possibly contain dangerous substances. For information on specific component types, please contact the manufacturer of that component. Always consult local authorities for regulations regarding proper disposal of electronic components in your area.

Further Information

For additional information about A10 products, terms and conditions of delivery, and pricing, contact your nearest A10 Networks location, which can be found by visiting www.a10networks.com.

Contents

1	Introduction.....	15
1.1	Audience.....	15
1.2	aXAPI Architecture.....	15
1.2.1	Admin Authentication	15
1.2.2	Session ID.....	15
1.2.3	Encoding.....	16
1.3	Request Format.....	16
1.3.1	Request Header.....	16
1.3.2	Request Body.....	17
1.3.3	Request Delimiters	19
1.3.4	Syntax Rules.....	20
1.3.5	Example Session.....	21
1.4	Deleting Parameter Settings	23
1.5	Error Handling and Error Codes.....	23
1.6	HTTP Status Codes	23
1.7	Error Codes and Messages	24
1.8	Session Timeout	24
2	JSON formatted data structure.....	24
2.1	Introduction	24
2.2	"Read" APIs in JSON	25
2.3	"Write" APIs in JSON	26
2.4	The "update" method implementation.....	27
2.5	Sub-level object methods	27
2.6	Response Status in JSON	28
3	Role-Based Access Control	29
3.1	Admin Roles.....	29
3.1.1	Preconfigured Admin Roles	29
4	aXAPI Methods: Authentication and Session Close	34
4.1.1	"authenticate" Method.....	34
4.1.2	"session.close" Method	34
5	System Management APIs.....	36
5.1	System Hostname	36
5.1.1	"system.hostname.get" Method	36
5.1.2	"system.hostname.set" Method	37
5.2	Time.....	37
5.2.1	"system.time.get" Method	37
5.2.2	"system.time.set" Method	40
5.3	NTP Server	41
5.3.1	"system.ntp.get" Method	41
5.3.2	"system.ntp.add" Method	42
5.3.3	"system.ntp.update" Method	43
5.3.4	"system.ntp.delete" Method	44
5.4	System Operation	45
5.4.1	"system.action.reload" Method	45
5.4.2	"system.action.reboot" Method	46
5.4.3	"system.action.shutdown" Method.....	47
5.4.4	"system.action.write_memory" Method.....	47
5.4.5	"system.startup.link" Method.....	48
5.5	Maintainence	49

5.5.1	“system.maintain.upgrade” Method	49
5.5.2	“system.backup” Method	50
5.5.3	“system.restore” Method	51
5.5.4	“system.banner.get” Method	52
5.5.5	“system.banner.set” Method.....	53
5.5.6	“system.boot_image.get” Method	54
5.5.7	“system.boot_image.set” Method.....	54
5.6	SNMP Configuration.....	55
5.6.1	“system.snmp.general.get” Method.....	55
5.6.2	“system.snmp.general.set” Method	56
5.6.3	“system.snmp.community.get” Method	57
5.6.4	“system.snmp.community.add” Method	58
5.6.5	“system.snmp.community.delete” Method.....	59
5.6.6	“system.snmp.trap.get” Method	60
5.6.7	“system.snmp.trap.add” Method	61
5.6.8	“system.snmp.trap.delete” Method.....	62
5.6.9	“system.snmp.trap_enable.get” Method.....	63
5.6.10	“system.snmp.trap_enable.set” Method	65
5.6.11	“system.snmp.mib.download” Method	67
5.7	Logging Configuration	67
5.7.1	“system.log.download” Method	67
5.7.2	“system.show_tech.download” Method	68
5.7.3	“system.log.level.get” Method	69
5.7.4	“system.log.level.set” Method.....	70
5.7.5	“system.log.server.get” Method	71
5.7.6	“system.log.server.set” Method.....	72
5.7.7	“system.log.buffer.get” Method	73
5.7.8	“system.log.buffer.set” Method.....	74
5.7.9	“system.log.smtp.get” Method	75
5.7.10	“system.log.smtp.set” Method.....	76
5.7.11	“system.log.audit.get” Method	77
5.7.12	“system.log.audit.set” Method.....	78
5.7.13	“system.log.clear” Method	79
5.8	System Infomation.....	80
5.8.1	“system.information.get” Method	80
5.8.2	“system.device_info.get” Method	81
5.9	System Performance	82
5.9.1	“system.performance.get” Method	82
5.10	System Resource Usage	84
5.10.1	“system.resource.get” Method.....	84
5.10.2	“system.resource.set” Method	85
5.10.3	“system.config_status.get” Method.....	87
5.10.4	“system.config_status.getAll” Method.....	88
6	Network Management APIs	90
6.1	Interface	90
6.1.1	“network.interface.get” Method	90
6.1.2	“network.interface.getAll” Method	92
6.1.3	“network.interface.set” Method.....	94
6.1.4	“network.interface.ipv4.add” Method	96
6.1.5	“network.interface.ipv4.delete” Method	97
6.1.6	“network.interface.ipv6.add” Method	97

6.1.7	“network.interface.ipv6.delete” Method	98
6.1.8	“network.interface.global.get” Method	99
6.1.9	“network.interface.global.set” Method.....	100
6.2	Virtual Interface.....	101
6.2.1	“network.ve.getAll” Method	101
6.2.2	“network.ve.get” Method	104
6.2.3	“network.ve.set” Method.....	106
6.2.4	“network.ve.ipv4.add” Method	107
6.2.5	“network.ve.ipv4.delete” Method.....	108
6.2.6	“network.ve.ipv6.add” Method	109
6.2.7	“network.ve.ipv6.delete” Method.....	110
6.3	Management Interface	111
6.3.1	“network.mgmt_interface.get” Method.....	111
6.3.2	“network.mgmt_interface.set” Method	112
6.4	Gateway Configuration	114
6.4.1	“network.transparent_interface.get” Method	114
6.4.2	“network.transparent_interface.set” Method.....	115
6.5	Trunk	116
6.5.1	“network.trunk.getAll” Method.....	116
6.5.2	“network.trunk.search” Method	118
6.5.3	“network.trunk.create” Method.....	119
6.5.4	“network.trunk.update” Method.....	120
6.5.5	“network.trunk.delete” Method.....	121
6.6	VLAN.....	122
6.6.1	“network.vlan.getAll” Method	122
6.6.2	“network.vlan.search” Method	123
6.6.3	“network.vlan.create” Method.....	124
6.6.4	“network.vlan.update” Method.....	125
6.6.5	“network.vlan.delete” Method.....	126
6.7	Route	127
6.7.1	“network.route.ipv4static.getAll” Method	127
6.7.2	“network.route.ipv4static.create” Method.....	128
6.7.3	“network.route.ipv4static.update” Method.....	129
6.7.4	“network.route.ipv4static.delete” Method.....	130
6.7.5	“network.route.ipv6static.getAll” Method	131
6.7.6	“network.route.ipv6static.create” Method.....	132
6.7.7	“network.route.ipv6static.update” Method.....	133
6.7.8	“network.route.ipv6static.delete” Method.....	134
6.8	ARP	135
6.8.1	“network.arp.getAll” Method	135
6.8.2	“network.arp.search” Method.....	136
6.8.3	“network.arp.create” Method	137
6.8.4	“network.arp.update” Method	138
6.8.5	“network.arp.delete” Method	139
6.9	ACL.....	140
6.9.1	“network.acl.getAll” Method	140
6.9.2	“network.acl.search” Method	144
6.9.3	“network.acl.create” Method	147
6.9.4	“network.acl.update” Method.....	149
6.9.5	“network.acl.delete” Method.....	151
6.10	DNS	152

6.10.1	“network.dns.server.get” Method	152
6.10.2	“network.dns.server.set” Method	153
7	SLB Management APIs	155
7.1	Service Group.....	155
7.1.1	“slb.service_group.getAll” Method.....	155
7.1.2	“slb.service_group.search” Method.....	158
7.1.3	“slb.service_group.fetchAllStatistics” Method	160
7.1.4	“slb.service_group.fetchStatistics” Method	163
7.1.5	“slb.service_group.create” Method	166
7.1.6	“slb.service_group.update” Method	169
7.1.7	“slb.service_group.delete” Method	172
7.1.8	“slb.service_group.member.create” Method	174
7.1.9	“slb.service_group.member.update” Method	175
7.1.10	“slb.service_group.member.delete” Method	177
7.2	Server.....	179
7.2.1	“slb.server.getAll” Method.....	179
7.2.2	“slb.server.search” Method.....	185
7.2.3	“slb.server.fetchAllStatistics” Method	189
7.2.4	“slb.server.fetchStatistics” Method	195
7.2.5	“slb.server.create” Method	198
7.2.6	“slb.server.update” Method	201
7.2.7	“slb.server.delete” Method	204
7.2.8	“slb.server.port.create” Method.....	205
7.2.9	“slb.server.port.update” Method.....	207
7.2.10	“slb.server.port.delete” Method.....	209
7.3	Virtual Server	211
7.3.1	“slb.virtual_server.getAll” Method	211
7.3.2	“slb.virtual_server.search” Method	222
7.3.3	“slb.virtual_server.fetchAllStatistics” Method	235
7.3.4	“slb.virtual_server.fetchStatistics” Method.....	239
7.3.5	“slb.virtual_server.create” Method.....	244
7.3.6	“slb.virtual_server.update” Method.....	256
7.3.7	“slb.virtual_server.delete” Method.....	269
7.3.8	“slb.virtual_server.vport.create” Method	271
7.3.9	“slb.virtual_server.vport.update” Method	278
7.3.10	“slb.virtual_server.vport.delete” Method	286
7.4	Virtual Server Template	288
7.4.1	“slb.template.vip.getAll” Method	288
7.4.2	“slb.template.vip.search” Method	290
7.4.3	“slb.template.vip.create” Method	291
7.4.4	“slb.template.vip.update” Method	293
7.4.5	“slb.template.vip.delete” Method	294
7.5	Virtual Server Port Template.....	295
7.5.1	“slb.template.vport.getAll” Method	295
7.5.2	“slb.template.vport.search” Method	297
7.5.3	“slb.template.vport.create” Method	298
7.5.4	“slb.template.vport.update” Method	299
7.5.5	“slb.template.vport.delete” Method	300
7.6	Server Template.....	301
7.6.1	“slb.template.server.getAll” Method	301
7.6.2	“slb.template.server.search” Method	303

7.6.3	“slb.template.server.create” Method	305
7.6.4	“slb.template.server.update” Method	306
7.6.5	“slb.template.server.delete” Method	308
7.7	Server Port Template	309
7.7.1	“slb.template.rport.getAll” Method	309
7.7.2	“slb.template.rport.search” Method	311
7.7.3	“slb.template.rport.create” Method	313
7.7.4	“slb.template.rport.update” Method	315
7.7.5	“slb.template.rport.delete” Method	316
7.8	HTTP Template	317
7.8.1	“slb.template.http.getAll” Method	317
7.8.2	“slb.template.http.search” Method	322
7.8.3	“slb.template.http.create” Method	324
7.8.4	“slb.template.http.update” Method	326
7.8.5	“slb.template.http.delete” Method	329
7.9	Cache Template	330
7.9.1	“slb.template.cache.getAll” Method	330
7.9.2	“slb.template.cache.search” Method	331
7.9.3	“slb.template.cache.create” Method	333
7.9.4	“slb.template.cache.update” Method	334
7.9.5	“slb.template.cache.delete” Method	335
7.10	PBSLB Template	336
7.10.1	“slb.template.pbslb.getAll” Method	336
7.10.2	“slb.template.pbslb.search” Method	341
7.10.3	“slb.template.pbslb.create” Method	343
7.10.4	“slb.template.pbslb.update” Method	345
7.10.5	“slb.template.pbslb.delete” Method	347
7.11	SMTP Template	348
7.11.1	“slb.template.smtp.getAll” Method	348
7.11.2	“slb.template.smtp.search” Method	349
7.11.3	“slb.template.smtp.create” Method	351
7.11.4	“slb.template.smtp.update” Method	352
7.11.5	“slb.template.smtp.delete” Method	353
7.12	SIP Template	354
7.12.1	“slb.template.sip.getAll” Method	354
7.12.2	“slb.template.sip.search” Method	356
7.12.3	“slb.template.sip.create” Method	358
7.12.4	“slb.template.sip.update” Method	359
7.12.5	“slb.template.sip.delete” Method	361
7.13	RTSP Template	362
7.13.1	“slb.template.rtsp.getAll” Method	362
7.13.2	“slb.template.rtsp.search” Method	363
7.13.3	“slb.template.rtsp.create” Method	364
7.13.4	“slb.template.rtsp.update” Method	365
7.13.5	“slb.template.rtsp.delete” Method	366
7.14	Connection-Reuse Template	367
7.14.1	“slb.template.conn_reuse.getAll” Methods	367
7.14.2	“slb.template.conn_reuse.search” Method	368
7.14.3	“slb.template.conn_reuse.create” Method	369
7.14.4	“slb.template.conn_reuse.update” Method	370
7.14.5	“slb.template.conn_reuse.delete” Method	371

7.15	TCP Template.....	372
7.15.1	“slb.template.tcp.getAll” Method.....	372
7.15.2	“slb.template.tcp.search” Method.....	373
7.15.3	“slb.template.tcp.create” Method	374
7.15.4	“slb.template.tcp.update” Method	375
7.15.5	“slb.template.tcp.delete” Method	376
7.16	UDP Template	377
7.16.1	“slb.template.udp.getAll” Method.....	377
7.16.2	“slb.template.udp.search” Method.....	378
7.16.3	“slb.template.udp.create” Method	379
7.16.4	“slb.template.udp.update” Method	380
7.16.5	“slb.template.udp.delete” Method	381
7.17	Cookie Persistence Template	382
7.17.1	“slb.template.cookie_persistence.getAll” Method	382
7.17.2	“slb.template.cookie_persistence.search” Method	383
7.17.3	“slb.template.cookie_persistence.create” Method	384
7.17.4	“slb.template.cookie_persistence.update” Method.....	385
7.17.5	“slb.template.cookie_persistence.delete” Method.....	386
7.18	Source-IP Persistence Template	387
7.18.1	“slb.template.src_ip_persistence.getAll” Method	387
7.18.2	“slb.template.src_ip_persistence.search” Method	389
7.18.3	“slb.template.src_ip_persistence.create” Method.....	390
7.18.4	“slb.template.src_ip_persistence.update” Method.....	391
7.18.5	“slb.template.src_ip_persistence.delete” Method.....	392
7.19	Destination-IP Template.....	393
7.19.1	“slb.template.dst_ip_persistence.getAll” Method	393
7.19.2	“slb.template.dst_ip_persistence.search” Method	394
7.19.3	“slb.template.dst_ip_persistence.create” Method.....	395
7.19.4	“slb.template.dst_ip_persistence.update” Method.....	396
7.19.5	“slb.template.dst_ip_persistence.delete” Method.....	397
7.20	SSL Persistence Template	398
7.20.1	“slb.template.ssl_sid_persistence.getAll” Method	398
7.20.2	“slb.template.ssl_sid_persistence.search” Method	399
7.20.3	“slb.template.ssl_sid_persistence.create” Method.....	400
7.20.4	“slb.template.ssl_sid_persistence.update” Method.....	401
7.20.5	“slb.template.ssl_sid_persistence.delete” Method.....	402
7.21	Client-SSL Template	403
7.21.1	“slb.template.client_ssl.getAll” Method.....	403
7.21.2	“slb.template.client_ssl.search” Method	405
7.21.3	“slb.template.client_ssl.create” Method	407
7.21.4	“slb.template.client_ssl.update” Method	408
7.21.5	“slb.template.client_ssl.delete” Method	410
7.22	Server-SSL Template	410
7.22.1	“slb.template.server_ssl.getAll” Method.....	410
7.22.2	“slb.template.server_ssl.search” Method	412
7.22.3	“slb.template.server_ssl.create” Method	413
7.22.4	“slb.template.server_ssl.update” Method	415
7.22.5	“slb.template.server_ssl.delete” Method	416
7.23	TCP-Proxy Template.....	417
7.23.1	“slb.template.tcp_proxy.getAll” Method.....	417
7.23.2	“slb.template.tcp_proxy.search” Method	418

7.23.3	“slb.template.tcp_proxy.create” Method	420
7.23.4	“slb.template.tcp_proxy.update” Method	421
7.23.5	“slb.template.tcp_proxy.delete” Method	422
7.24	DNS Template	423
7.24.1	“slb.template.dns.getAll” Method	423
7.24.2	“slb.template.dns.search” Method	425
7.24.3	“slb.template.dns.create” Method	427
7.24.4	“slb.template.dns.update” Method	429
7.24.5	“slb.template.dns.delete” Method	430
7.25	Diameter Template	431
7.25.1	“slb.template.diameter.getAll” Method	431
7.25.2	“slb.template.diameter.search” Method	433
7.25.3	“slb.template.diameter.create” Method	435
7.25.4	“slb.template.diameter.update” Method	436
7.25.5	“slb.template.diameter.delete” Method	438
7.26	Health Monitor	439
7.26.1	“slb.hm.getAll” Method	439
7.26.2	“slb.hm.search” Method	442
7.26.3	“slb.hm.create” Method	446
7.26.4	“slb.hm.update” Method	448
7.26.5	“slb.hm.delete” Method	451
7.26.6	“slb.hm.external.getAll” Method	451
7.26.7	“slb.hm.external.search” Method	452
7.26.8	“slb.hm.external.create” Method	453
7.26.9	“slb.hm.external.update” Method	454
7.26.10	“slb.hm.external.delete” Method	455
7.26.11	“slb.hm.http_post_file.getAll” Method	456
7.26.12	“slb.hm.http_post_file.delete” Method	457
7.26.13	“slb.hm.http_post_file.download” Method	457
7.26.14	“slb.hm.http_post_file.upload” Method	458
7.26.15	“slb.hm.global.get” Method	458
7.26.16	“slb.hm.global.set” Method	459
7.27	aFleX	460
7.27.1	“slb.aflex.getAll” Method	460
7.27.2	“slb.aflex.search” Method	461
7.27.3	“slb.aflex.upload” Method	462
7.27.4	“slb.aflex.download” Method	462
7.27.5	“slb.aflex.update” Method	463
7.27.6	“slb.aflex.delete” Method	463
7.27.7	“slb.aflex.fetchAllStatistics” Method	464
7.27.8	“slb.aflex.fetchStatistics” Method	465
7.27.9	“slb.aflex.clearAllStatistics” Method	465
7.27.10	“slb.aflex.clearStatistics” Method	466
7.27.11	“slb.aflex.clearAllEvents” Method	467
7.27.12	“slb.aflex.clearEvents” Method	467
7.28	Class List	469
7.28.1	“slb.class_list.getAll” Method	469
7.28.2	“slb.class_list.search” Method	471
7.28.3	“slb.class_list.upload” Method	474
7.28.4	“slb.class_list.download” Method	474
7.28.5	“slb.class_list.create” Method	475

7.28.6	“slb.class_list.update” Method	477
7.28.7	“slb.class_list.delete” Method	479
7.28.8	“slb.class_list.entry.create” Method	480
7.28.9	“slb.class_list.entry.update” Method	481
7.28.10	“slb.class_list.entry.delete” Method	483
7.28.11	“slb.class_list.dns.create” Method.....	484
7.28.12	“slb.class_list.dns.update” Method.....	486
7.28.13	“slb.class_list.dns.delete” Method.....	488
7.29	GLID List	489
7.29.1	“slb.glid.getAll” Method	489
7.29.2	“slb.class_list.search” Method.....	491
7.29.3	“slb.glid.create” Method.....	492
7.29.4	“slb.glid.update” Method.....	493
7.29.5	“slb.glid.delete” Method.....	493
7.30	Global	494
7.30.1	“slb.global.settings.get” Method	494
7.30.2	“slb.global.settings.set” Method.....	496
7.30.3	“slb.global.ddos_protection.get” Method.....	497
7.30.4	“slb.global.ddos_protection.set” Method	498
7.30.5	“slb.global.log_rate_limiting.get” Method	499
7.30.6	“slb.global.log_rate_limiting.set” Method	500
7.31	SSL Management	500
7.31.1	“slb.ssl.getAll” Method	500
7.31.2	“slb.ssl.search” Method.....	502
7.31.3	“slb.ssl.create” Method.....	503
7.31.4	“slb.ssl.upload” Method	504
7.31.5	“slb.ssl.download” Method	505
7.32	RAM Cache	506
7.32.1	“slb.cache.entry.clear” Method	506
7.32.2	“slb.cache.stats.clear” Method.....	507
7.33	PBSLB	507
7.33.1	“slb.pbslb.getAll” Method	507
7.33.2	“slb.pbslb.search” Method.....	509
7.33.3	“slb.pbslb.create” Method	510
7.33.4	“slb.pbslb.update” Method	510
7.33.5	“slb.pbslb.delete” Method	511
7.33.6	“slb.pbslb.import” Method	512
8	HA Management APIs.....	513
8.1	Global	513
8.1.1	“ha.global.get” Method	513
8.1.2	“ha.global.set” Method.....	514
8.1.3	“ha.sync_config” Method.....	515
8.1.4	“ha.force_self_standby” Method	516
8.1.5	“ha.status.get” Method	516
8.2	Group	518
8.2.1	“ha.group.getAll” Method	518
8.2.2	“ha.group.fetchStatistics” Method.....	519
8.2.3	“ha.group.create” Method.....	520
8.2.4	“ha.group.delete” Method	521
8.3	Status Check	522
8.3.1	“ha.status_check.gw.getAll” Method	522

8.3.2	“ha.status_check.gw.create” Method.....	523
8.3.3	“ha.status_check.gw.delete” Method.....	524
8.3.4	“ha.status_check.vlan.getAll” Method	524
8.3.5	“ha.status_check.vlan.create” Method.....	526
8.3.6	“ha.status_check.vlan.delete” Method.....	527
8.4	Floating IP	528
8.4.1	“ha.floating_ip.getAll” Method.....	528
8.4.2	“ha.floating_ip.create” Method	529
8.4.3	“ha.floating_ip.delete” Method	530
8.5	Inline Mode	531
8.5.1	“ha.inline_mode.get” Method.....	531
8.5.2	“ha.inline_mode.set” Method.....	532
8.6	Interface.....	533
8.6.1	“ha.interface.getAll” Method	533
8.6.2	“ha.interface.get” Method.....	534
8.6.3	“ha.interface.set” Method	535
9	GSLB Management APIs	537
9.1	DNS Proxy	537
9.1.1	“gslb.dns_proxy.getAll” Method.....	537
9.1.2	“gslb.dns_proxy.search” Method	540
9.1.3	“gslb.dns_proxy.create” Method	543
9.1.4	“gslb.dns_proxy.update” Method	545
9.1.5	“gslb.dns_proxy.delete” Method	548
9.1.6	“gslb.dns_proxy.vport.create” Method.....	549
9.1.7	“gslb.dns_proxy.vport.update” Method.....	551
9.1.8	“gslb.dns_proxy.vport.delete” Method.....	553
9.2	Policy.....	555
9.2.1	“gslb.policy.getAll” Method	555
9.2.2	“gslb.policy.search” Method	564
9.2.3	“gslb.policy.create” Method	570
9.2.4	“gslb.policy.update” Method	575
9.2.5	“gslb.policy.delete” Method	580
9.3	Service IP	581
9.3.1	“gslb.service_ip.getAll” Method	581
9.3.2	“gslb.service_ip.search” Method	584
9.3.3	“gslb.service_ip.create” Method	586
9.3.4	“gslb.service_ip.update” Method	588
9.3.5	“gslb.service_ip.delete” Method	590
9.3.6	“gslb.service_ip.port.create” Method	591
9.3.7	“gslb.service_ip.port.update” Method	592
9.3.8	“gslb.service_ip.port.delete” Method	594
9.4	Site.....	595
9.4.1	“gslb.site.getAll” Method	595
9.4.2	“gslb.site.search” Method	599
9.4.3	“gslb.site.create” Method	602
9.4.4	“gslb.site.update” Method	605
9.4.5	“gslb.site.delete” Method	608
9.4.6	“gslb.site.slb_device.create” Method	609
9.4.7	“gslb.site.slb_device.update” Method	611
9.4.8	“gslb.site.slb_device.delete” Method	612
9.4.9	“gslb.site.ip_server.create” Method.....	614

9.4.10 “gslb.site.ip_server.delete” Method.....	615
9.5 SNMP Template	618
9.5.1 “gslb.snmp_template.getAll” Method	618
9.5.2 “gslb.snmp_template.search” Method.....	620
9.5.3 “gslb.snmp_template.create” Method.....	622
9.5.4 “gslb.snmp_template.update” Method	624
9.5.5 “gslb.snmp_template.delete” Method.....	626
9.6 Zone.....	627
9.6.1 “gslb.zone.getAll” Method.....	627
9.6.2 “gslb.zone.search” Method.....	633
9.6.3 “gslb.zone.create” Method	637
9.6.4 “gslb.zone.update” Method	640
9.6.5 “gslb.zone.delete” Method	643
9.6.6 “gslb.zone.service.create” Method	644
9.6.7 “gslb.zone.service.update” Method	647
9.6.8 “gslb.zone.service.search” Method	650
9.6.9 “gslb.zone.service.delete” Method	652
9.6.10 “gslb.zone.service.dns_txt_record.create” Method	654
9.6.11 “gslb.zone.service.dns_txt_record.update” Method	655
9.6.12 “gslb.zone.service.dns_txt_record.delete” Method	656
9.6.13 “gslb.zone.service.dns_txt_record.getAll” Method.....	657
9.6.14 “gslb.zone.service.dns_txt_record.deleteAll” Method	658
9.6.15 “gslb.zone.service.dns_mx_record.create” Method	659
9.6.16 “gslb.zone.service.dns_mx_record.update” Method	660
9.6.17 “gslb.zone.service.dns_mx_record.delete” Method	661
9.6.18 “gslb.zone.service.dns_mx_record.getAll” Method.....	662
9.6.19 “gslb.zone.service.dns_mx_record.deleteAll” Method	663
9.6.20 “gslb.zone.service.dns_mx_record.search” Method.....	664
9.6.21 “gslb.zone.service.dns_cname_record.create” Method	665
9.6.22 “gslb.zone.service.dns_cname_record.update” Method	666
9.6.23 “gslb.zone.service.dns_cname_record.delete” Method	668
9.6.24 “gslb.zone.service.dns_cname_record.getAll” Method.....	669
9.6.25 “gslb.zone.service.dns_cname_record.deleteAll” Method	670
9.6.26 “gslb.zone.service.dns_cname_record.search” Method	671
9.6.27 “gslb.zone.service.dns_ns_record.create” Method.....	672
9.6.28 “gslb.zone.service.dns_ns_record.update” Method.....	673
9.6.29 “gslb.zone.service.dns_ns_record.delete” Method	674
9.6.30 “gslb.zone.service.dns_ns_record.getAll” Method	675
9.6.31 “gslb.zone.service.dns_ns_record.deleteAll” Method	676
9.6.32 “gslb.zone.service.dns_srv_record.create” Method	677
9.6.33 “gslb.zone.service.dns_srv_record.update” Method	678
9.6.34 “gslb.zone.service.dns_srv_record.delete” Method	680
9.6.35 “gslb.zone.service.dns_srv_record.getAll” Method	681
9.6.36 “gslb.zone.service.dns_srv_record.deleteAll” Method	682
9.6.37 “gslb.zone.service.dns_srv_record.search” Method	683
9.6.38 “gslb.zone.service.dns_ptr_record.create” Method	684
9.6.39 “gslb.zone.service.dns_ptr_record.update” Method	685
9.6.40 “gslb.zone.service.dns_ptr_record.delete” Method	686
9.6.41 “gslb.zone.service.dns_ptr_record.getAll” Method	687
9.6.42 “gslb.zone.service.dns_ptr_record.deleteAll” Method	688
9.6.43 “gslb.zone.service.dns_ptr_record.search” Method	689

9.6.44	“gslb.zone.service.dns_address_record.create” Method	690
9.6.45	“gslb.zone.service.dns_address_record.update” Method	691
9.6.46	“gslb.zone.service.dns_address_record.delete” Method	693
9.6.47	“gslb.zone.service.dns_address_record.getAll” Method	694
9.6.48	“gslb.zone.service.dns_address_record.deleteAll” Method	696
9.6.49	“gslb.zone.service.dns_address_record.search” Method.....	696
9.6.50	“gslb.zone.service.geo_location.create” Method.....	698
9.6.51	“gslb.zone.service.geo_location.update” Method	699
9.6.52	“gslb.zone.service.geo_location.delete” Method.....	701
9.6.53	“gslb.zone.service.geo_location.getAll” Method	702
9.6.54	“gslb.zone.service.geo_location.deleteAll” Method	703
9.6.55	“gslb.zone.service.geo_location.search” Method.....	704
9.7	GSLB Global	705
9.7.1	“gslb.global.get” Method.....	705
9.7.2	“gslb.global.set” Method	707
10	NAT Pool Management APIs (V2.0)	710
10.1	Pool.....	710
10.1.1	“nat.pool.getAll” Method	710
10.1.2	“nat.pool.search” Method	711
10.1.3	“nat.pool.fetchAllStatistics” Method.....	712
10.1.4	“nat.pool.fetchStatistics” Method.....	714
10.1.5	“nat.pool.create” Method.....	715
10.1.6	“nat.pool.update” Method	717
10.1.7	“nat.pool.delete” Method.....	718
10.2	Pool Group	720
10.2.1	“nat.pool_group.getAll” Method	720
10.2.2	“nat.pool_group.search” Method.....	721
10.2.3	“nat.pool_group.create” Method	722
10.2.4	“nat.pool_group.update” Method	723
10.2.5	“nat.pool_group.delete” Method	725
10.3	Static NAT	726
10.3.1	“nat.static_translation.getAll” Method	726
10.3.2	“nat.static_translation.search” Method.....	727
10.3.3	“nat.static_translation.fetchAllStatistics” Method.....	728
10.3.4	“nat.static_translation.fetchStatistics” Method.....	730
10.3.5	“nat.static_translation.create” Method	731
10.3.6	“slb.static_translation.update” Method.....	732
10.3.7	“nat.static_translation.delete” Method	733
10.4	NAT Range.....	734
10.4.1	“nat.range.getAll” Method	734
10.4.2	“nat.range.search” Method	736
10.4.3	“nat.range.create” Method.....	737
10.4.4	“nat.range.update” Method	738
10.4.5	“nat.range.delete” Method	740
10.5	ACL Bind	741
10.5.1	“nat.acl_bind.getAll” Method.....	741
10.5.2	“nat.acl_bind.create” Method	742
10.5.3	“nat.acl_bind.delete” Method	743
10.6	Interface	745
10.6.1	“nat.interface.getAll” Method	745
10.6.2	“nat.interface.create” Method	746

10.6.3 “nat.interface.delete” Method.....	747
10.7 Global	748
10.7.1 “nat.global.get” Method	748
10.7.2 “nat.global.set” Method.....	750
11 RBA/L3V Partition Management APIs	752
11.1 “system.partition.getAll” Method	752
11.2 “system.partition.search” Method	753
11.3 “system.partition.create” Method.....	754
11.4 “system.partition.update” Method.....	754
11.5 “system.partition.active” Method.....	755
11.6 “system.partition.delete” Method.....	756
12 CLI Deploying APIs	758
12.1 “cli.show_info” Method	759
12.2 “cli.deploy” Method	760
13 AXAPI Method Management APIs	762
13.1 “axapi.method.getAll” Method	762
13.2 “axapi.method.search” Method	764
14 Error Messages	767

1 Introduction

This document describes the AX Application Programming Interface (aXAPI). The aXAPI enables you to use custom third-party applications to configure and monitor AX devices.

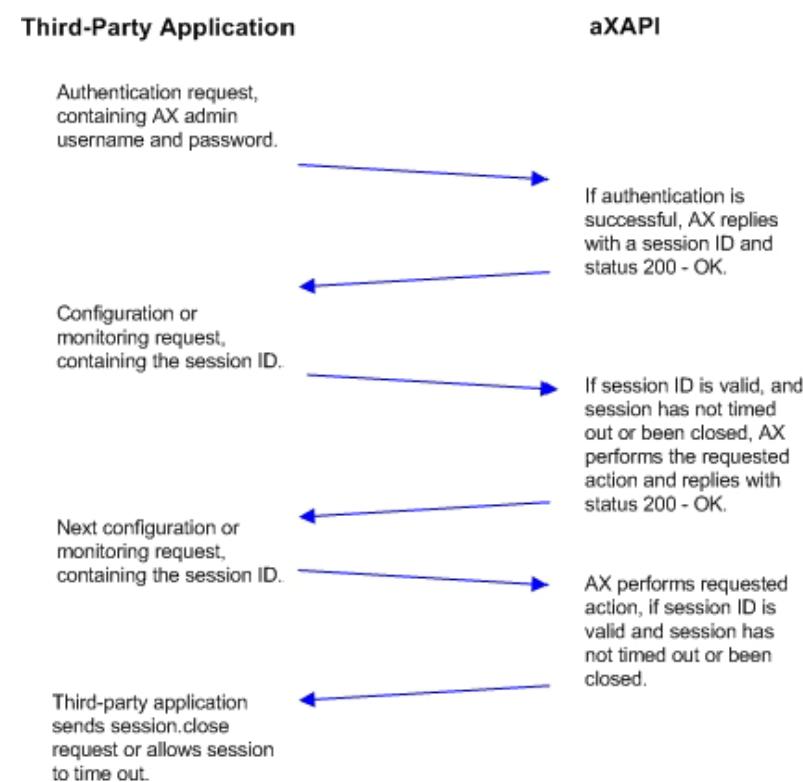
1.1 Audience

This document is for network architects for determining applicability and planning implementation, and for system administrators for provisioning and maintenance of A10 Networks AX Series devices.

1.2 aXAPI Architecture

The aXAPI is based on the Representational State Transfer (REST) architecture. The aXAPI uses the HTTPS transportation protocol with the request/response model to exchange data over HTTPS. Figure 1 – **aXAPI Session Flow** shows a typical aXAPI session flow.

Figure 1 – aXAPI Session Flow



1.2.1 Admin Authentication

The aXAPI uses the same admin authentication resources as those configured for CLI and GUI access. For example, if the AX device is configured to use RADIUS first to authenticate admins, RADIUS will be used first when authenticating an admin for an aXAPI session.

1.2.2 Session ID

The first request from the third-party application sends the authentication method along with a valid AX admin username and password.

If the username and password are valid, the AX device replies with a session ID. The third-party application must present the session ID with all future requests during that session. The session ID is valid until the third-party application sends a session close request or the session times out.

1.2.3 Encoding

The aXAPI expects all data to be UTF-8 encoded, and it checks for valid UTF-8 sequences. If an invalid sequence is found, the aXAPI assumes that the data is ISO-8859-1 encoded and converts it to UTF-8. The aXAPI discards data that is sent in any other format.

1.3 Request Format

A properly formatted request to the aXAPI is a URI request header and a request body. The request body can be a URI-based or JSON-based data structure. The request can be sent as an HTTP or HTTPS GET or POST action.

Note: To use HTTP, you must disable HTTP-to-HTTPS redirection on the AX device. In the CLI, use the **no web-service auto-redir** command at the global configuration level of the CLI. In the GUI, select Config > System > Settings. On the Web tab, de-select the Re-direct HTTP to HTTPS checkbox. Click OK.

1.3.1 Request Header

The request header is a URL in the following format:

```
https://<AX-ipaddr>:<port>/services/rest/<aXAPI->
version>/?session_id=<session id>&method=<aXAPI method>
name>&format=<data format>
```

The *AX-ipaddr* is the host name or IP address of the AX device (IPv4 or IPv6).

The *port* is the HTTPS service port on AX device. It is shared by both aXAPI and the AX web GUI. By default, the port number is 80 for HTTP and 443 for HTTPS. You can omit the port number when using the default ports.

To change the port number, use the following command from the CLI:

```
web-service port xxx
web-service secure-port xxx
```

Or change the port from the AX web GUI by navigating as follows:

```
Config mode -> system -> settings -> web
```

The *aXAPI-version* specifies the API version to use:

- V1 – The aXAPI version supported in AX Release 2.0 and later.
- V1.1 – The aXAPI version supported in AX Release 2.4.2 and later.
- V1.2 – The aXAPI version supported in AX Release 2.6.0 and later.
- V2.0 – The aXAPI version supported in AX Release 2.6.1-P3, 2.6.2, and later.

All methods supported in V1 are also supported in V1.1. Likewise, all methods supported in V1 or V1.1 are supported in V1.2.

If a method's implementation is changed from one aXAPI version to the next, the implementation in the specified aXAPI version will be used. Differences are noted in the description for the method. There are no differences in method implementations between V1, V1.1, and V1.2, but parts of the V2 methods may differ from older versions.

The *session id* is the string returned by the authentication method. For the authentication method, omit the following parameter (&session_id=<session id>), since you may not have the session ID at that time.

The *aXAPI method name* indicates the method name of this request.

The aXAPI is organized according to a series of methods and their corresponding data structure. Generally speaking, the method name defines the action of the request, and the data structure presents the data in the appropriate format.

The *data format* specifies which data format you wish to use in the aXAPI request and response. aXAPI has the following formats:

- **url:** (default) url-based data for requests and XML-based data for responses
- **json:** json-based data for both requests and responses
- **xml:** (not currently supported) XML-based data for both requests and responses

You can leave the *data format* field empty when using the 'default' data format.

Notes:

- Not all methods are supported in JSON-based format at present. For details on which methods support JSON format, please refer to the specific method within this document.
- The rest of the URL must be exactly as shown.
- You can use either HTTP or HTTPS to process your aXAPI request, but HTTPS is strongly recommended for enhanced security. To change the server certificate, log into the AX web GUI and navigate as follows:
Config mode > System > Settings > Web certificate

1.3.2 Request Body

The body consists of a set of parameters and the values for those parameters. The body can be URI-based or JSON-based. For the 'default' URL-based data format, you can either place the request body in the request header or in the HTTP POST Body. Examples of each are shown below.

Request body in Request Header (HTTP URL):

```
https://192.168.100.140/services/rest/V1.2/?session_id=XXXXXXXXXXXXXXXXXXXXXX  
XXXXXXXX&method=slb.vip.create&name=xyz_company  
&address=201.2.1.100  
&status=1  
&service-list=service1%03http_service_config1%02service2%03https_service_config2  
&service1=type%0314%02port%0380%02service_group%03http_srv_grp  
&http_service_config1=http_tmpl%03http_tmpl1  
&service2=type%0315%02port%03443%02service_group%03https_srv_grp  
&https_service_config2=client_ssl_tmpl%03clientSSLTemplate%02server_ssl_tmpl%03serverSSLTemplate
```

Request body in HTTP POST Body:

HTTP URL:

https://192.168.100.140/services/rest/V1.2/?session_id=XXXXXXXXXXXXXXXXXXXXXX
XXXXXXX&method=slb.vip.create

POST Body:

```
name=xyz_company
&address=201.2.1.100
&status=1
&service-
list=service1%03http_service_config1%02service2%03https_service_config2
&service1=type%0314%02port%0380%02service_group%03http_srv_grp
&http_service_config1=http_tmpl1%03http_tmpl1
&service2=type%0315%02port%03443%02service_group%03https_srv_grp
&https_service_config2=client_ssl_tmpl%03clientSSLTemplate%02server_ssl_tmpl%0
3serverSSLTemplate
```

Note: The length of the HTTP argument buffer is limited. Such limitations exist on both the AX web server and on your HTTP client. When sending large data structures, we recommend using HTTP POST body. Generally speaking, for methods which do ‘write’ operations, such as *slb.server.create*, and *slb.global.set*, you should always use POST body. Failure to follow this recommendation may cause the HTTP argument buffer to overflow, resulting in an HTTP 404 error code (bad request).

JSON-based format entries are supported in aXAPI version 2.0 or above. These JSON-based entries use the “format=json”. You must place the request body in HTTP POST Body, as follows:

HTTP URL:

https://192.168.100.140/services/rest/V2/?session_id=XXXXXXXXXXXXXXXXXXXXXX
XXXXXXX&method=slb.vip.create&format=json

POST Body: (JSON formatted data structure)

```
{"server": {"name": "s1", "host": "124.254.1.131", "gslb_external_address": "0.0.0.0", "weight": 1, "health_monitor": "(default)", "status": 1, "conn_limit": 8000000, "conn_limit_log": 1, "conn_resume": 0, "stats_data": 1, "extended_stats": 0, "slow_start": 0, "spoofing_cache": 0, "template": "default", "port_list": [{"port_num": 66, "protocol": 2, "status": 1, "weight": 1, "no_ssl": 0, "conn_limit": 100, "conn_limit_log": 1, "conn_resume": 100, "template": "default", "stats_data": 1, "health_monitor": "", "extended_stats": 0}, {"port_num": 20, "protocol": 3, "status": 1, "weight": 1, "no_ssl": 0, "conn_limit": 8000000, "conn_limit_log": 1, "conn_resume": 0, "template": "default", "stats_data": 1, "health_monitor": "(default)", "extended_stats": 0}, {"port_num": 21, "protocol": 2, "status": 1, "weight": 1, "no_ssl": 0, "conn_limit": 8000000, "conn_limit_log": 1, "conn_resume": 0, "template": "default", "stats_data": 1, "health_monitor": "(default)", "extended_stats": 0}, {"port_num": 80, "protocol": 2, "status": 1, "weight": 1, "no_ssl": 0, "conn_limit": 8000000, "conn_limit_log": 1, "conn_resume": 0, "template": "default", "stats_data": 1, "health_monitor": "(default)", "extended_stats": 0}]}}
```

Example for an aXAPI request with a default data format:

```
https://192.168.100.140/services/rest/V1.2/?
session_id=XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
&method=slb.vip.create&name=xyz_company
&address=201.2.1.100
&status=1
&service-
list=service1%03http_service_config1%02service2%03https_service_config2
&service1=type%0314%02port%0380%02service_group%03http_srv_grp
&http_service_config1=http_tmpl1%03http_tmpl1
&service2=type%0315%02port%03443%02service_group%03https_srv_grp
&https_service_config2=client_ssl_tmpl%03clientSSLTemplate%02server_ssl_tmpl%0
3serverSSLTemplate
```

This request would produce the following response:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Example for an aXAPI request with JSON data format:

```
https://192.168.100.140/services/rest/V2/?
session_id=XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
&method=slb.vip.create&format=json
```

POST Body:

```
{"server": {"name": "s1", "host": "124.254.1.131", "gslb_external_address": "0.0.0.0", "weight": 1, "health_monitor": "(default)", "status": 1, "conn_limit": 8000000, "conn_limit_log": 1, "conn_resume": 0, "stats_data": 1, "extended_stats": 0, "slow_start": 0, "spoofing_cache": 0, "template": "default", "port_list": [{"port_num": 66, "protocol": 2, "status": 1, "weight": 1, "no_ssl": 0, "conn_limit": 100, "conn_limit_log": 1, "conn_resume": 100, "template": "default", "stats_data": 1, "health_monitor": "", "extended_stats": 0}, {"port_num": 20, "protocol": 3, "status": 1, "weight": 1, "no_ssl": 0, "conn_limit": 8000000, "conn_limit_log": 1, "conn_resume": 0, "template": "default", "stats_data": 1, "health_monitor": "(default)", "extended_stats": 0}, {"port_num": 21, "protocol": 2, "status": 1, "weight": 1, "no_ssl": 0, "conn_limit": 8000000, "conn_limit_log": 1, "conn_resume": 0, "template": "default", "stats_data": 1, "health_monitor": "(default)", "extended_stats": 0}, {"port_num": 80, "protocol": 2, "status": 1, "weight": 1, "no_ssl": 0, "conn_limit": 8000000, "conn_limit_log": 1, "conn_resume": 0, "template": "default", "stats_data": 1, "health_monitor": "(default)", "extended_stats": 0}]}}}
```

You would get the following response: (json formatted)

```
{"response": {"status": "OK"}}
```

1.3.3 Request Delimiters

aXAPI requests use the delimiters described in the table below. The delimiters are shown in bold type in the example above, and coloring has been added to make the delimiters easier to identify in longer aXAPI requests.

TABLE 1 aXAPI Request Delimiters

Delimiter	Description
?	Separates the header from the body.
&	Separates each parameter. Note: For simplicity, each parameter is shown on a separate line in the example above. In a properly formatted request, there are no blank spaces or line breaks. (See Syntax Rules.)
=	Separates a parameter name and its group of values. Note: The = delimiter is not equivalent to %02 . The aXAPI interprets them differently. Attempting to substitute one delimiter for the other will result in an error.
%03	Separates a parameter value and its setting. Note: If your application does not use a web browser to send the request, you must use (char)3 instead of %03 .

%02	For parameters that have multiple values, separates each setting. Note: If your application does not use a web browser to send the request, you must use (char)2 instead of %02.
-----	---

1.3.4 Syntax Rules

The parameter values can not contain any special characters (#, \$, & and so on) or control characters (blank spaces, soft or hard line breaks, tabs, backspace, and so on).

Note: Line breaks were added to the example above for readability. An actual request can not use line breaks.

Parameters are not sequence-sensitive. They can appear in any order in the request. To delete a parameter or set it to its null (unconfigured) setting, include the parameter name but do not enter anything after the = delimiter. (See Deleting Parameter Settings on page 23.)

Response Format

The aXAPI uses the REST response format.

- All responses are encoded in UTF-8.
- The time is expressed in Unix epoch time format.
- List responses with zero elements include an empty container element and an HTTP response code of 200.
- Responses to requests for a single element, when the element does not exist, return an error message and an HTTP response code of 404.

Normally, a method call returns the following XML document:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
    [xml-payload]
</response>
```

Or the following JSON document:

```
{"response": {"status": "OK"}}
```

However, if an error occurs, the following is returned instead:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="fail">
    <error code="[error-code]" msg="[error-message]" />
</response>
```

Or the following JSON format:

```
{"response": {"status": "fail", "err": {"code": [error-code], "msg": [error-message]}}}
```

Note: Unless defined otherwise, the rest of this chapter assumes that you are using the default URL-based data format (and not JSON-based format).

1.3.5 Example Session

The following requests and responses represent a simple aXAPI session. The first request is an authentication request, to which the aXAPI replies with a session ID. The next request configures a virtual server.

Authentication Request:

```
https://192.168.100.140/services/rest/V1/?method=authenticate&username=admin&password=a10
```

The header sends the request to the aXAPI interface on an AX device at IP address 192.168.100.140.

The body contains the following parameters:

- method – This parameter states the type of operation being requested. In this example, the request is for a session ID to use with each subsequent request in the session.
- username – This parameter is an AX admin username.
- password – This is the password that goes with the username.

Response:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
<session_id>XXXXXXXXXXXXXXXXXXXXXXXXXXXX</session_id>
</response>
```

The response sends a session ID, shown in this example as “XXXXXXXXXXXXXXXXXXXXXXXXXXXX”.

Virtual Server Create Request:

```
https://192.168.100.140/services/rest/V1/?session_id=XXXXXXXXXXXXXXXXXXXXXX
XXXX&method=slb.vip.create&name=xyz_company&address=201.2.1.100&status=1&service-
e-
list=service1%03http_service_config1%02service2%03https_service_config2&service
1=type%0314%02port%0380%02service_group%03http_grp&http_service_config1=htt
p_temp1%03http_tmpl1&service2=type%0315%02port%03443%02service_group%03https_sr
v_grp&https_service_config2=client_ssl_temp1%03clientSSLTemplate%02server_ssl_t
empl%03serverSSLTemplate
```

The header sends the request to the aXAPI interface on an AX device at IP address 192.168.100.140.

The body contains the following configuration parameters:

- `session_id` – This parameter specifies the session ID returned by the aXAPI in response to the authentication request from the third-party application. The session ID must be included in the request and must be valid.
- `method` – This parameter states the type of operation being requested. In this example, the request is to create a virtual server (`slb.vip.create`). A method is required in every request. The following parameters are specific to the `slb.vip.create` method. All the parameters that are supported for each method are described in this guide. The descriptions below are specifically for this example.
 - `name` – This parameter specifies the name of the virtual server.
 - `address` – This parameter specifies the IP address of the virtual server.
 - `status` – This parameter specifies the status (1 = enabled, 0 = disabled).
 - `service-list` – This parameter specifies the service ports to configure on the server. Each service port is defined by `servicen`, where *n* uniquely identifies the service within the request, and by a set of parameters specific to the service type. The services and the sets of service-type parameters each appear later in the request as separate parameters. In this example, the service-list contains the following:
 - “service1” and its set of service-type parameters for HTTP, `http_service_config1`. The “`http_service_config`” portion specifies the service type and “1” identifies the specific instance of the service type. Use the same number for the service and the set of service type parameters to be set for the service.
 - “service2” and its set of service-type parameters for HTTPS, `https_service_config2`.
 - `service1` – This parameter defines settings that are not specific to the service type, for service1. In this example, the `type` parameter specifies the service type (14, which is HTTP), `port` specifies the protocol port number, and `service_group` specifies the name of the service group to which to bind the service port.
 - `http_service_config1` – This parameter defines settings for service1’s HTTP service port. In this example, `http_temp1` specifies an HTTP template for the service port.
 - `service2` – This parameter defines settings that are not specific to the service type, for service2.
 - `https_service_config2` – This parameter defines settings for service1’s HTTPS service port. In this example, `client_ssl_temp1` specifies the name of a client-SSL template, and `server_ssl_temp1` specifies the name of a server-SSL template.

Response:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

This response indicates that the request was executed, with no errors. The virtual server has been added to the AX device’s configuration.

1.4 Deleting Parameter Settings

The aXAPI provides methods for deleting SLB resources. For example, the slb.vip.delete method can be used to delete a virtual server.

You also can use the aXAPI to delete individual parameter settings for a resource. In this case, to delete the parameter setting, use the parameter in the request, but do not enter anything for the value.

For example, to clear the connection resume setting for a service port, send an update request such as the following:

```
https://192.168.100.140/services/rest/V1/?session_id=XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX  
XXXXa&method=slb.service-  
group.update&name=test3&type=2&lb_method=1&health_monitor=ping  
&conn_limit=1000000&conn_resume=&member-  
list=member1%02member2%02member3&member1=  
address%031.1.1.2%02port%031234%02weight%032&member2=address%031.1.1.3%02port%0  
31234%02priority%033&member3=address%031.1.1.4%02port%031234%02status%030
```

Instead of a number, there is no parameter value following “&conn_resume=”. Parameters that have default values will be reset to their defaults by the delete operation.

1.5 Error Handling and Error Codes

The aXAPI response can indicate an error condition, for example:

- The requested information does not exist.
- The aXAPI request is invalid; for example, a method or other parameter is misspelled or has an incorrect value.
- An internal error occurs.

An error response includes a numeric error code and a human-readable message. For a list of error responses, please refer to “on before aXAP” on page 727.

Note: A10 Networks recommends that applications use the error codes rather than the error text to detect and handle errors, because the message text may change over time.

1.6 HTTP Status Codes

In addition to an error code and message, the aXAPI returns an appropriate HTTP status code, such as the following:

- 200 – OK. This is the HTTP successful status code, and means the aXAPI request and response were successfully executed.
It is still possible for an error code and error message to be returned in the response, if a successfully executed request and response results in an error.
- 403 – Forbidden. Standard HTTP status code.
- 404 – Not Found. Standard HTTP status code.
- 500 – Internal Server Error. This status code is used if an internal error occurs.
- 999 – The admin username or password sent with an authentication request is not correct.

1.7 Error Codes and Messages

aXAPI requests that result in errors include both a code and a message. The error codes are divided into the following:

- 1001 to 9999. Indicate errors such as syntax errors, expired session, invalid username or password, unsupported response format, and so on.
- 10001 or above. Indicates an internal error.

1.8 Session Timeout

By default, an idle aXAPI session times out after 10 minutes. You can change the timeout using the AX command-line interface (CLI) or graphical user interface (GUI). The aXAPI timeout can be 0-60 minutes. If you specify 0, sessions never time out.

To display and configure the aXAPI session timeout, use either of the following methods.

USING THE GUI

1. Log into the management GUI with a valid admin account that has at least read-write privileges.
2. Select Config > System > Settings.
3. On the menu bar, select Web.
4. Click the arrow next to aXAPI to display the configuration field and value for the Timeout.
5. To change the timeout, edit the number in the aXAPI Timeout field.
6. Click OK.

USING THE CLI

1. Log into the CLI with a valid admin account that has at least read-write privileges.
2. Use the following command to display the current aXAPI timeout:
show web-service
3. To change the timeout, use the following command at the global configuration level of the CLI:
web-service axapi-timeout-policy idle minutes

2 JSON formatted data structure

2.1 Introduction

The aXAPI 2.0 supports the JavaScript Object Notation (JSON) format, and can be specified in the SLB API methods by including "format=json" in the URL request. This denote that the AX request and response should be in JSON format. The JSON format is defined as the standard JSON object structure with the AX API parameter tag names and parameter value types defined in the aXAPI Reference Manual.

The following is a list of aXAPI methods that offer support for JSON format:

- slb.server.*

- slb.service_group.*
- slb.virtual_server.*
- slb.global.*
- slb.class_list.*
- slb.glid.*
- slb.pbslb.*
- nat.*
- network.vlan.*
- gslb.*

2.2 "Read" APIs in JSON

When using the “read” APIs, such as “search”, “getAll”, “featchAllStatistics” or “fetchStatistics”, with the “format=json”, the HTTP action is GET, and parameters are specified in the URL string.

For example, to call the method “slb.service_group.search”, the URL string would be:

https://_AX_IP_address_/services/rest/V2/?session_id=_valid_sid_&format=json&method=slb.service_group.search&name=group1

This call will return the service group (“group 1”):

```
{
  "service_group": {
    "name": "group1",
    "protocol": 2,
    "lb_method": 3,
    "health_monitor": "",
    "min_active_member": {
      "status": 0,
      "priority_set": 0
    },
    "client_reset": 0,
    "stats_data": 1,
    "extended_stats": 0,
    "member_list": [
      {
        "server": "_s_31.3.3.3",
        "port": 80,
        "template": "",
        "priority": 1,
        "status": 0,
        "stats_data": 1
      },
      {
        "server": "_s_30.3.3.3",
        "port": 80,
        "template": "",
        "priority": 1,
        "status": 0,
        "stats_data": 1
      },
      {
        "server": "_s_14.14.14.2",
        "port": 80,
        "template": "",
        "priority": 1,
        "status": 0,
        "stats_data": 1
      },
      {
        "server": "_s_14.14.14.3",
        "port": 80,
        "template": "",
        "priority": 1,
        "status": 0,
        "stats_data": 1
      }
    ]
  }
}
```

```

        "template":"",
        "priority":1,
        "status":0,
        "stats_data":1
    }
]
}
}
```

This is the equivalent to the following CLI command:

```
!
slb service-group group1 tcp
method weighted-least-connection
member _s_14.14.14.3:80 disable
member _s_14.14.14.2:80 disable
member _s_30.3.3.3:80 disable
member _s_31.3.3.3:80 disable
!
```

2.3 "Write" APIs in JSON

When using the "write" APIs, such as "create", "update", or "delete" with the "format=json", the HTTP action has to be "POST". The URL only contains the "session_id", "method", and "format=json", and the HTTP request body has the POST data for the object structure.

For example, to call the "slb.service_group.create" method to create a new service group called "group2", the URL string is:

```
https://_AX_IP_address_/services/rest/V2/?session_id=_valid_sid_&format=json&method=slb.service_group.create
```

And the POST data is:

```
{
  "service_group": {
    "name": "group2",
    "protocol": 2,
    "lb_method": 3,
    "stats_data": 1,
    "member_list": [
      {
        "server": "5.5.5.5",
        "port": 80,
        "status": 1
      },
      {
        "server": "5.5.5.4",
        "port": 80,
        "status": 0
      },
      {
        "server": "5.5.5.3",
        "port": 80,
        "status": 1
      }
    ]
  }
}
```

This is the equivalent to the following CLI command:

```
!
slb service-group group2 tcp
method weighted-least-connection
member 5.5.5.5:80
member 5.5.5.4:80 disable
member 5.5.5.3:80
!
```

2.4 The "update" method implementation

The SLB object "update" method can be used to update the SLB object attributes, as well as the sub-level objects for given SLB object. For example, the method "slb.service_group.update" can be used to update the following load-balancing method:

```
{  
    "service_group":{  
        "name":"group2",  
        "lb_method":2 // set to "least-connection"  
    }  
}
```

This updates the "group2" load-balancing method from "weighted-least-connection" to "least-connection".

To update a member in the "group2", you could use the following:

```
{  
    "service_group":{  
        "name":"group2",  
        "member_list": [  
            {  
                "server":"5.5.5.4", // member key fields: server, port  
                "port":80,  
                "status":1  
            }  
        ]  
    }  
}
```

This will enable the existing member, 5.5.5.4:80

To add a member to "group2", you could use the following:

```
{  
    "service_group":{  
        "name":"group2",  
        "member_list": [  
            {  
                "server":"5.5.5.40",  
                "port":80,  
                "status":1  
            }  
        ]  
    }  
}
```

2.5 Sub-level object methods

The SLB sub-level objects are the real ports in the real server, service group members in the service group, and virtual ports in the virtual servers. As the SLB object implementation of the "update" methods, it can be used to modify the existing sub-level objects for a given SLB object, or to add new sub-level objects to SLB objects. However, keep in mind that the "update" method cannot be used to delete the existing sub-level objects from a given SLB object. Therefore, we offer several the following sub-level object methods to accomplish those tasks:

- slb.server.port.delete: delete a real port
- slb.server.port.update: update an existing real port
- slb.server.port.add: add a new real port

- slb.server.port.deleteAll: delete all real ports
- slb.server.port.updateAll: apply the same changes to all real ports

There are similar sub-level object methods for service group methods (slb.service_group.member.*), and virtual server methods (slb.virtual_server.vport.*)

2.6 Response Status in JSON

When the request format is JSON, the response status also appears in JSON format. If the request is successful, the following message appears:

```
{"response": {"status": "OK"}}
```

However, if the request fails:

```
{
  "response":{
    "status":"fail",
    "err":{
      "code":1165,
      "msg":"Parameter err. Parameter service_group.member_list.rest_auto_gen_item_1.server in method slb.service_group.update is required."
    }
  }
}
```

Note: A10 Networks is continuing to improve the JSON error message responses.

3 Role-Based Access Control

Admin roles enable you to restrict the aXAPI options that an admin is authorized to use. For each method, the admin role determines whether the admin is allowed to access (view) the page. If the admin is allowed to access the page, the role specifies whether the admin has read-only or read-write privileges for the page.

You can assign an admin to a preconfigured role or a custom role that you configure. You can also create customized admin roles.

3.1 Admin Roles

Admin Roles are a way to control who can configure the AX device. The admin role assigned to an AX user determines which features they are allowed to access via the GUI, CLI, or aXAPI.

In addition to determining whether a user is allowed to execute a particular aXAPI method (or access a page in the GUI), admin roles also determine which actions are available to an administrator, for example, read-only or read-write privileges.

Note: If you are an admin who has been assigned an admin role with limited privileges, such as “ReadOnlyAdmin” and you try to execute an aXAPI method for which you do not have permissions, you will see an error message similar to the following:
“Error Code 1006 – Access denied: no privilege for this method”

Note: Admin roles must be configured via the AX GUI and cannot be configured using aXAPI methods.

To assign one of the preconfigured admin roles or create a custom admin role of your own, access the admin role configuration page in the AX GUI by navigating as follows:

Config > System > Admin > Role

For additional details on configuring admin roles, please refer to the *AX GUI Reference*.

3.1.1 Preconfigured Admin Roles

The table on the following page lists preconfigured admin roles and AX features. You can refer to this table to find out if you have the permissions required to access a particular GUI page, and by association, the correspondent aXAPI method.

How to use the table:

- Take note of the admin role that has been assigned to you – you will need this later.
(The following page lists preconfigured admin roles.)

- Take note of the GUI path associated with the method you wish to look up. This path is listed for most of the methods in this document, and it typically appears under the “Menus Privilege” heading.
- Scan the table rows for the GUI path that is associated with the method you want to look up. For example, the “system.hostname.get” method on page 36 lists the associated GUI path as **Config Mode >> Network >> DNS**, so you can now scan the rows in the table for this method, which appears in the “Config” subsection.
- Once you have found the **row** that best correlates to your method, now scan the **columns** to find the one that matches your admin role. For example, if you had been assigned the “SystemAdmin” role, then you would use this information to look for the 3rd column in the table. In the cell where the path meets the role, a letter appears. This letter (“R”, “W”, or “H”) tells you if you have read or write privileges. If an “H” appears in the cell, this means the page is hidden from your view (in the GUI), or it means you do not have permission to run the associated method.
- As an example, consider the following:
 - If you have ReadOnlyAdmin privileges for the "http template" module, then for the aXAPI, you could only use the following two methods:
(1) "slb.template.http.getAll" and (2) "slb.template.http.search" method.
 - However, if you tried to execute the following methods, they would be refused:
(1) "slb.template.http.create", (2) "slb.template.http.update", and (3) "slb.template.http.delete".
- Similarly, if you have one of the admin roles for which the "http template" module privileges indicate “hidden”, then this means that your attempts to run any of the "slb.template.http.*" methods would be refused.

Table Column Descriptions

Numbers in the table columns indicate AX preconfigured admin roles.

<ul style="list-style-type: none"> • 1 – ReadOnlyAdmin • 2 – ReadWriteAdmin • 3 – SystemAdmin • 4 – NetworkAdmin • 5 – NetworkOperator • 6 – SlbServiceAdmin 	<ul style="list-style-type: none"> • 7 – SlbServiceOperator • 8 – PartitionNetworkAdmin • 9 – PartitionNetworkOperator • 10 – PartitionSlbServiceAdmin • 11 – PartitionSlbServiceOperator • 12 – PartitionReadOnly
--	--

The following letters indicate the access privileges for the GUI page:

- **R** – Read-only
- **W** – Read-write
- **H** – Hidden (page can not be viewed by the admin, or aXAPI method unavailable)

TABLE - Preconfigured GUI Access Roles

GUI Page ¹	Role and Access											
	1	2	3	4	5	6	7	8	9	10	11	12
Monitor Pages												
Monitor > Overview > Summary	R	R	R	R	R	R	R	R	R	R	R	R
Monitor > Overview > Status	R	R	H	H	H	R	R	R	H	R	R	R
Monitor > Overview > Statistics	R	R	H	H	H	R	R	R	H	R	R	R
Monitor > Overview > Performance	R	R	H	H	H	R	R	R	H	R	R	R
Monitor > Service > SLB	R	R	H	H	H	R	R	R	H	R	R	R
Monitor > Service > Health Monitor	R	R	H	H	H	R	R	H	H	H	H	H
Monitor > Service > PBSLB	R	R	H	H	H	R	R	H	H	H	H	H
Monitor > Service > GSLB	R	R	H	H	H	R	R	H	H	H	H	H
Monitor > Service > aFleX	R	R	H	H	H	R	R	R	H	R	R	R
Monitor > Service > IP Source NAT	R	R	H	H	H	R	R	H	H	H	H	H
Monitor > Service > Application	R	R	H	H	H	R	R	R	H	R	R	R

¹In some cases where the same access privileges apply to all pages at a given GUI level, only the high-level page name is listed in this table. However, access is configurable on an individual page basis for all GUI pages.

Monitor > Network > Interface	R	R	H	R	R	H	H	R	R	H	H	R
Monitor > Network > Trunk	R	R	H	R	R	H	H	R	R	H	H	R
Monitor > Network > VLAN	R	R	H	R	R	H	H	R	R	H	H	R
Monitor > Network > ACL	R	R	H	R	R	H	H	R	R	H	H	R
Monitor > Network > ARP	R	R	H	R	R	H	H	R	R	H	H	R
Monitor > Network > Route	R	R	H	R	R	H	H	R	R	H	H	R
Monitor > System > Admin	R	R	R	H	H	H	H	H	H	H	H	H
Monitor > System > Logging	R	R	R	H	H	H	H	H	H	H	H	H
Monitor > System > aVCS	R	R	R	H	H	H	H	H	H	H	H	H
Monitor > HA > Group	R	R	H	H	H	R	R	H	H	H	H	H
Monitor > HA > Status	R	R	H	H	H	R	R	H	H	H	H	H

Config Pages

Config > Get Started > Basic System	R	W	H	W	R	H	H	H	H	H	H	H
Config > Get Started > Smart Template	R	W	H	H	H	W	R	W	H	W	R	R
Config > Get Started > GSLB Easy Config	R	W	H	H	H	W	R	H	H	H	H	H
Config > Service > SLB	R	W	H	H	H	W	R	W	H	W	R	R
Config > Service > Template	R	W	H	H	H	W	R	W	H	W	R	R
Config > Service > Health Monitor	R	W	H	H	H	W	R	W	H	W	R	R
Config > Service > PBSLB	R	W	H	H	H	W	R	H	H	H	H	H
Config > Service > GSLB	R	W	H	H	H	W	R	H	H	H	H	H
Config > Service > aFleX	R	W	H	H	H	W	R	W	H	W	R	R
Config > Service > IP Source NAT ²	R	W	H	H	H	W	R	W	H	W	R	R
Config > Service > SSL Management	R	W	H	H	H	W	R	W	H	W	R	R
Config > Network > Interface [†]	R	W	H	W	R	H	H	W	R	H	H	R
Config > Network > Trunk [†]	R	W	H	W	R	H	H	W	H	H	H	H
Config > Network > VLAN [†]	R	W	H	W	R	H	H	W	R	H	H	R
Config > Network > ACL [†]	R	W	H	W	R	H	H	W	R	H	H	R
Config > Network > ARP [†]	R	W	H	W	R	H	H	W	R	H	H	R
Config > Network > Route [†]	R	W	H	W	R	H	H	W	R	H	H	R
Config > Network > DNS [†]	R	W	H	W	R	H	H	H	H	H	H	H

²For the partition roles (8-12), the access privileges shown in the table are for admins of partitions in which Layer 2/3 virtualization is enabled. If Layer 2/3 virtualization is disabled in the partition, this page is hidden.

Config > Network > ICMP Rate Limiting [†]	R	W	H	W	R	H	H	H	H	H	H	H	H
Config > Network > BPDU-Fwd-Group [†]	R	W	H	W	R	H	H	H	H	H	H	H	H
Config > System > Settings > Web	R	W	W	H	H	H	H	H	H	H	H	H	H
Config > System > Settings > Web Certificate	R	W	W	H	H	H	H	H	H	H	H	H	H
Config > System > Settings > Terminal	R	W	W	H	H	H	H	H	H	H	H	H	H
Config > System > Settings > Log	R	W	W	W	R	H	H	H	H	H	H	H	H
Config > System > Settings > General	R	W	W	H	H	H	H	H	H	H	H	H	H
Config > System > Settings > Boot	R	W	W	H	H	H	H	H	H	H	H	H	H
Config > System > Settings > Action	R	W	W	H	H	H	H	H	H	H	H	H	H
Config > System > Admin	R	W	W	H	H	H	H	H	H	H	H	H	H
Config > System > Access Control	R	W	W	W	R	H	H	H	H	H	H	H	H
Config > System > Time	R	W	W	W	R	H	H	H	H	H	H	H	H
Config > System > SNMP	R	W	W	W	R	H	H	H	H	H	H	H	H
Config > System > Maintenance	R	W	W	H	H	H	H	H	H	H	H	H	H
Config > System > Console	R	W	W	H	H	H	H	H	H	H	H	H	H
Config > System > Config File	R	W	W	H	H	H	H	H	H	H	H	H	H
Config > System > Diagnosis	R	W	W	H	H	H	H	H	H	H	H	H	H
Config > System > aVCS	R	W	W	H	H	H	H	H	H	H	H	H	H
Config > HA > Setting	R	W	H	H	H	W	R	W	H	H	H	H	H
Config > HA > Config Sync	R	W	H	H	H	W	R	W	H	H	H	H	H

4 aXAPI Methods: Authentication and Session Close

This chapter describes the aXAPI methods for starting and closing aXAPI sessions.

4.1.1 “authenticate” Method

Parameters

Parameter Name	Description	Data Type	Range	Default
username	Admin username for access to the AX device.	String	N/A	N/A
password	Password for the admin username.	String	N/A	N/A

Response Fields

The AX device authenticates the session based on the username and password. If authentication is successful, the AX device response includes a session ID. Each subsequent request must include the session ID as part of the request.

The session ID is valid until the session times out or the third-party application sends a session close request.

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V1/?method=authenticate&username=admin&password=a10
- HTTP Action:
GET or POST
- CLI configuration before aXAPI call:
N/A
- HTTP Body
N/A
- CLI configuration after aXAPI call:
N/A

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
<session_id>123456789</session_id>
</response>
```

Menus Privilege

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

4.1.2 “session.close” Method

Close an aXAPI session.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id	Session ID assigned by the AX device following successful authentication.	String	N/A	N/A

Response Fields

The AX device authenticates the session based on the username and password. If authentication is successful, the AX device response includes a session ID. Each subsequent request must include the session ID as part of the request.

The session ID is valid until the session times out or the third-party application sends a session close request.

Example Response

Request as:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V1/?session_id=123456789&method=session.create
- **HTTP Action:**
GET
- **CLI configuration before aXAPI call:**
N/A
- **HTTP Body**
N/A
- **CLI configuration after aXAPI call:**
N/A

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
< session_id>123456789</ session_id>
</response>
```

Menus Privilege

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

5 System Management APIs

5.1 System Hostname

5.1.1 “system.hostname.get” Method

Get DNS server host name that has been configured on the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	system.hostname.get	String		

Note: (*) parameter is required in the API.

Response Fields

Hostname XML tag for the hostname.

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
hostname AX2100-XLI
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=system.hostname.get
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
  <hostname>AX2100-XLI</hostname>
</response>
```

Menus Privilege

Config Mode >> Network >> DNS

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

5.1.2 “system.hostname.set” Method

This method enables you to set the DNS server host name on the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	system.hostname.set	String		
hostname (*)	hostname	String	1-31	

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
hostname AX2100-XLI
!
```

HTTP Data

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=system.hostname.set&hostname=AX2000
```

Configuration after call:

```
!
hostname AX2000
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Network >> DNS

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

5.2 Time

These methods enable you to set and retrieve the configured system time, date, and timezone.

5.2.1 “system.time.get” Method

This method enables you to get the configured system time, date, and timezone.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	system.time.get	String		

Note: (*) parameter is required in the API.

Response Fields

datetime XML tag for the collection of time
date system date info, format: 2010-07-10
time system time info, format: 03:41:55
time_zone system timezone info
(GMT-11:00)Midway Island, Samoa
(GMT-10:00)Hawaii
(GMT-09:00)Alaska
(GMT-08:00)Pacific Time – Tijuana
(GMT-08:00)Pacific Time(US & Canada)
(GMT-08:00)Pacific Time - west British Columbia
(GMT-07:00)Arizona
(GMT-07:00)Mountain Time(US & Canada)
(GMT-06:00)Central Time(US & Canada)
(GMT-06:00)Mexico City
(GMT-06:00)Saskatchewan
(GMT-06:00)Central America
(GMT-05:00)Eastern Time(US & Canada)
(GMT-05:00)Indiana(East)
(GMT-05:00)Eastern Time - Ontario & Quebec - most locations
(GMT-05:00)Eastern Time
(GMT-05:00)Eastern Time - Toronto, Ontario
(GMT-04:00)Caracas, La Paz
(GMT-04:00)Atlantic Time(Canada)
(GMT-04:00)Santiago
(GMT-03:30>Newfoundland
(GMT-03:00>Buenos Aires, Georgetown
(GMT-03:00>Greenland
(GMT-02:00>Mid-Atlantic
(GMT-01:00>Azores
(GMT-01:00>Cape Verde Is.
(GMT)Greenwich Mean Time: Dublin, Edinburgh, Lisbon, London
(GMT+01:00>West Central Africa
(GMT+01:00>Amsterdam, Berlin, Bern, Rome, Stockholm, Vienna
(GMT+01:00>Belgrade, Bratislava, Budapest, Ljubljana, Prague
(GMT+01:00>Brussels, Copenhagen, Madrid, Paris
(GMT+01:00>Sarajevo, Skopje, Sofija, Vilnius, Warsaw, Zagreb
(GMT+02:00>Bucharest
(GMT+02:00>Cairo
(GMT+02:00>Athens, Istanbul, Minsk
(GMT+02:00>Harare, Pretoria
(GMT+02:00>Jerusalem
(GMT+02:00>Helinki, Riga, Tallinn
(GMT+03:00>Nairobi
(GMT+03:00>Baghdad

(GMT+03:00)Kuwait, Riyadh
 (GMT+03:00)Moscow, St.Petersburg, Volgograd
 (GMT+03:30)Tehran
 (GMT+04:00)Baku, Tbilisi, Yerevan
 (GMT+04:00)Abu Dhabi, Muscat
 (GMT+04:30)Kabul
 (GMT+05:00)Islamabad, Karachi, Tashkent
 (GMT+05:00)Ekaterinburg
 (GMT+05:30)Calcutta, Chennai, Mumbai, New Delhi
 (GMT+05:45)Kathmandu
 (GMT+06:00)Almaty, Novosibirsk
 (GMT+06:00)Astana, Dhaka
 (GMT+06:00)Sri Jayawardenepura
 (GMT+06:30)Rangoon
 (GMT+07:00)Bangkok, Hanoi, Jakarta
 GMT+07:00)Krasnoyarsk
 (GMT+08:00)Irkutsk, Ulaan Bataar
 (GMT+08:00)Kuala Lumpur, Singapore
 (GMT+08:00)Beijing, Chongqing, Hong Kong, Urumqi
 (GMT+08:00)Taipei
 (GMT+08:00)Perth
 (GMT+09:00)Seoul
 (GMT+09:00)Osaka, Sapporo, Tokyo
 (GMT+09:00)Yakutsk
 (GMT+09:30)Adelaide
 (GMT+09:30)Darwin
 (GMT+10:00)Hobart
 (GMT+10:00)Brisbane
 (GMT+10:00)Vladivostok
 (GMT+10:00)Canberra, Melbourne, Sydney
 (GMT+10:00)Guam, Port Moresby
 (GMT+11:00)Magadan, Solomon., New Caledonia
 (GMT+12:00)Auckland, Wellington
 (GMT+12:00)Fiji, Kamchatka, Marshall Is.
 (GMT+12:00)Eniwetok, Kwajalein
 (GMT+13:00)Nuku'alofa

dst_status DST status either enabled(1) or disabled(0)

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
clock timezone Europe/Belgrade
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=system.time.get
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
  <datetime>
    <date>01/01/2000</date>
    <time>00:06:05</time>
    <time_zone>(GMT+01:00) Belgrade, Bratislava, Budapest, Ljubljana,
      Prague</time_zone>
    <dst_status>1</dst_status>
  </datetime>
</response>
```

Menus Privilege

Config Mode >> System >> Time

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

5.2.2 “system.time.set” Method

This method enables you to set the configured system time, date, and timezone.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	system.time.set	String		
date	date format: MM/DD/YYYY	String		
time	time format: HH:MM:SS	String		
time_zone	time zone, the full name of time zone, please refer to method 'system.time.get'	String		
dst_status	dst status either enable(1) or disable(0)	Int	1 or 0	

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
- **HTTP Action:**
POST

CLI configuration before aXAPI call:

```
!
clock timezone Asia/Shanghai
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=system.time.set&date=01/01/2000&time=00:00:00&dst_
status=1&time_zone=(GMT+01:00) Belgrade, Bratislava, Budapest, Ljubljana, Prague
```

CLI configuration after aXAPI call:

```
!
clock timezone Europe/Belgrade
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> System >> Time

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

5.3 NTP Server

Set and view information about the Network Time Protocol (NTP) server with which the AX device is synchronized.

5.3.1 “system.ntp.get” Method

This method enables you to get info about the configured NTP server.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	system.ntp.get	String		

Note: (*) parameter is required in the API.

Response Fields

ntp_list	XML tag for NTP servers
ntp	XML tag for the collection of NTP
server	name/IP-address
status	NTP server status either enabled(1) or disabled(0)

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
ntp server 1.2.3.4
!
ntp server 3ff6::6678
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxx&method=system.ntp.get

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
<ntp_list>
```

```

<ntp>
  <server>1.2.3.4</server>
  <status>1</status>
</ntp>
<ntp>
  <server>3ff6::6678</server>
  <status>1</status>
</ntp>
</ntp_list>
</response>

```

Menus Privilege

Config Mode >> System >> Time

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

5.3.2 “system.ntp.add” Method

This method enables you to add a NTP server, as well as enable or disable a particular NTP server.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	system.ntp.add	String		
ntp_list (*)	NTP server list to be added ntp1^Bntp2^B ... ntpN ^B: ASCII Code 0x02, URL-encode %02	String		
ntp<n> server (*) status	ntp at element <n> NTP server name/IP-address NTP server status either enable(1) or disable(0) Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03	String Int	1 or 0	1

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
ntp server 3ff6::6678
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=system.ntp.add&ntp_list=ntp1%02ntp2%02ntp3&ntp1=se
rver%031.1.1.2%02status%031&ntp2=server%031.1.1.3&ntp3=server%031.1.1.4
```

CLI configuration after aXAPI call:

```
!
ntp server 3ff6::6678
!
ntp server 1.1.1.2
!
ntp server 1.1.1.3
!
ntp server 1.1.1.4
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> System >> Time

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

5.3.3 “system.ntp.update” Method

This method updates any information configured on the AX for an NTP server.

Note: If the NTP server does not already exist, then this NTP server will be added instead of updated.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	system.dns.update	String		
ntp_list (*)	NTP server list to be added ntp1^Bntp2^B ... ntpN ^B: ASCII Code 0x02, URL-encode %02	String		
ntp<n> server (*) status	ntp at element <n> NTP server name/IP-address NTP server status either enable(1) or disable(0) Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03	String Int	1 or 0	1

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
- **HTTP Action:**

POST

CLI configuration before aXAPI call:

```
!
ntp server 3ff6::6678
!
ntp server 1.1.1.2
!
ntp server 1.1.1.3
!
ntp server 1.1.1.4
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=system.ntp.update&ntp_list=ntp1%02ntp2%02ntp3&ntp1
=server%031.1.1.2%02status%030&ntp2=server%031.1.1.3&ntp3=server%031.1.1.4%02status%030
```

CLI configuration after aXAPI call:

```
!
ntp server 3ff6::6678
!
ntp server 1.1.1.2
ntp disable 1.1.1.2
!
ntp server 1.1.1.3
!
ntp server 1.1.1.4
ntp disable 1.1.1.4
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> System >> Time

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

5.3.4 “system.ntp.delete” Method

This method deletes any configured NTP servers.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	system.ntp.delete	String		
ntp_list (*)	NTP server list to be added ntp1^Bntp2^B ... ntpN ^B: ASCII Code 0x02, URL-encode %02	String		
ntp<n> server (*)	ntp at element <n> NTP server name/IP-address Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03	String		

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
ntp server 3ff6::6678
!
ntp server 1.1.1.2
!
ntp server 1.1.1.3
!
ntp server 1.1.1.4
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=system.ntp.delete&ntp_list=ntp1%02ntp2%02ntp3&ntp1
=server%031.1.1.2&ntp2=server%031.1.1.3&ntp3=server%031.1.1.4
```

CLI configuration after aXAPI call:

```
!
ntp server 3ff6::6678
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> System >> Time

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

5.4 System Operation

5.4.1 “system.action.reload” Method

This method can be used to reload the AX device, which restarts system processes and reloads the startup-config, without reloading the system image. Note that this option closes all sessions.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	system.action.reload	String		
write_memory (*)	not write memory(0) or write memory(1)	Int	0 or 1	

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
- **HTTP Action:**
POST
- **HTTP Body**
session_id=xxxxxxxxxxxxxxxxxxxx&method=system.action.reload&write_memory=1

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> System >> Settings >> Action >> Reload

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

5.4.2 “system.action.reboot” Method

This method reboots the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	system.action.reboot	String		
write_memory (*)	not write memory(0) or write memory(1)	Int	0 or 1	

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
- **HTTP Action:**
POST
- **CLI configuration before aXAPI call:**
N/A
- **HTTP Body**
session_id=xxxxxxxxxxxxxxxxxxxx&method=system.action.reboot&write_memory=1
- **CLI configuration after aXAPI call:**
N/A

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> System >> Settings >> Action >> Reboot

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

5.4.3 “system.action.shutdown” Method

This method shuts down the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	system.action.shutdown	String		
write_memory (*)	not write memory(0) or write memory(1)	Int	0 or 1	

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST
- CLI configuration before aXAPI call:
N/A
- HTTP Body
session_id=xxxxxxxxxxxxxxxxxxxx&method=system.action.shutdown&write_memory=1
- CLI configuration after aXAPI call:
N/A

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> System >> Settings >> Action >> Shutdown

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

5.4.4 “system.action.write_memory” Method

This method writes the AX device’s running config to the startup config.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	system.action.write_memory	String		

media	system booting media	String	"cf" or "disk"	current boot device
target	startup-config profile "primary" write primary startup config "secondary" write secondary startup config or other profile file	String	"primary" "secondary" or other profile name	current profile
partition	partition name. "shared" will write shared partition memory "all" will write memory for all partitions	String	"shared", "all" or others	current partition

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST
- CLI configuration before axAPI call:
N/A
- HTTP Body
session_id=xxxxxxxxxxxxxxxxxxxx&method=system.action.write_memory&media=disk&targe t=secondary&partition=shared
- CLI configuration after aXAPI call:
N/A

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> System >> Settings >> Action >> Save

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

5.4.5 “system.startup.link” Method

This method is used to determine which startup config file the AX should boot from, as well as the location of the file (hard disk or compact-flash) and the partition (primary or secondary).

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	system.startup.link	String		
filename	the name of the startup config file. The file name can be 'default' or user specified file name.	String		default
media	hard disk(disk) or compact-flash(cf)	String	disk or cf	disk
partition	Both hard disk and cf card has a 'primary' partition and 'secondary' partition. Use this parameter to identify on which partition the operation will be done. If you specify 'primary' here, the primary partition on hard disk or cf card will use the startup config file which is identified by the given 'filename' parameter.	String	primary or secondary	primary

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST
- CLI configuration before aXAPI call:
N/A
- HTTP Body
session_id=xxxxxxxxxxxxxxxxxxxx&method=system.startup.link&filename=m2&media=disk&partition=secondary
- CLI configuration after aXAPI call:
N/A

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privileqe

Config Mode >> System >> Config File

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

5.5 Maintenance

5.5.1 “system.maintain.upgrade” Method

This method is used to upgrade the AX primary/secondary image on hard disk or compact-flash.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	system.maintain.upgrade	String		
media (*)	Hard disk(0) or compact-flash(1) or both(2).	int	0 or 1 or 2	
destination (*)	primary(0) or secondary(1)	int	0 or 1	
reboot	No(0) or Yes(1)	Int	0 or 1	1
use_mgmt_port	No(0) or Yes(1)	int	0 or 1	0
write_memory	No(0) or Yes(1)	Int	0 or 1	1
protocol (*) <i>Attention: if you are using http protocol, do not provide this option.</i>	File transfer protocol to use: • FTP – File Transfer Protocol • TFTP – Trivial File Transfer Protocol • RCP – Unix Remote Copy Protocol • SCP – Secure Copy Protocol	String		
host (*) <i>Attention: if you are using http protocol, do not provide this option.</i>	Hostname or IP address of the remote server	String		
port (*) <i>Attention: if you are using http</i>	Protocol port number for FTP/SCP. This parameter is applicable only for FTP/SCP, and only if the server does not use the default port	Int	1-65535	SCP:22 FTP:21

protocol, do not provide this option.	number.			
location (*)	Image filename and directory path on the remote server.	String		
username	Username required for access to the remote server.	String		
password	Password required for access to the remote server.	String		

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST
- CLI configuration before aXAPI call:
N/A
- HTTP Body
session_id=xxxxxxxxxxxxxxxxxxxxxx&method=system.maintain.upgrade&media=1&destination=1&reboot=0&protocol=scp&host=192.168.3.177&location=/home/image&username=root&password=a10
- CLI configuration after aXAPI call:
N/A

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> System >> Maintenance >> Upgrade

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

5.5.2 “system.backup” Method

This method is used to perform a system backup (backup configuration is implemented in the system.config-file.upload/restore).

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	system.backup	String		

Note: (*) parameter is required in the API.

Response Fields

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/

- **HTTP Action:**
(Multipart) GET
- **CLI configuration before aXAPI call:**
N/A
- **HTTP Body**
session_id=xxxxxxxxxxxxxxxxxxxx&method=system.backup
- **CLI configuration after aXAPI call:**
N/A

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> System >> Maintenance >> Backup >> System

Config Mode >> System >> Maintenance >> Backup >> Log

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

5.5.3 “system.restore” Method

This method is used to restore the AX Series to a saved backup configuration from a previously saved backup file on either a local or a remote host.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	system.restore	String		

Note: (*) parameter is required in the API.

Response Fields

Example Response

Request as:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
- **HTTP Action:**
(Multipart) POST
- **CLI configuration before aXAPI call:**
N/A
- **HTTP Body**
session_id=xxxxxxxxxxxxxxxxxxxx&method=system.restore
- **CLI configuration after aXAPI call:**
N/A

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> System >> Maintenance >> Restore >> System

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

5.5.4 “system.banner.get” Method

This method allows you to view the system login banner displayed in the AX CLI.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	system.banner.get	String		

Note: (*) parameter is required in the API.

Response Fields

banner

 login_banner the login baner
 exec_banner for the exec baner

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
GET
- CLI configuration before axAPI call:

```
!
banner exec multi-line %~
BBBBBBBBBB%~
!
banner login multi-line %~
AAAAAAAAAAA%~
!
!
```

- HTTP Body
session_id=xxxxxxxxxxxxxxxxxxxx&method=system.banner.get

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
    <banner>
        <login_banner><! [CDATA[AAAAAAAAAAA] ]></login_banner>
        <exec_banner><! [CDATA[BBBBBBBBBB] ]></exec_banner>
    </banner>
</response>
```

Menus Privilege

Config Mode >> System >> Settings >> Terminal >> Banner

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

5.5.5 “system.banner.set” Method

This method allows you to configure the banner messages displayed in the CLI.
By default, the banner message is “**Welcome to AX**”.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	system.banner.set	String		
login_banner	login banner content	String	Len 1-1021	
exec_banner	exec banner content	String	Len 1-1021	

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
`https://[AX_IP_ADDRESS]/services/rest/V2/`
- HTTP Action:
`POST`

CLI configuration before aXAPI call:

```
!
!
end
```

- HTTP Body
`session_id=xxxxxxxxxxxxxxxxxxxx&method=system.banner.set`
- Post Body
`login_banner=AAAAAAA&exec_banner=BBBBBBBB`

CLI configuration after aXAPI call:

```
!
banner exec multi-line %~
BBBBBBBBB%~
!
banner login multi-line %~
AAAAAAA%~
!
!
end
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> System >> Settings >> Terminal >> Banner

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

5.5.6 “system.boot_image.get” Method

This method displays the location of the boot image from which the system image will be loaded the next time the AX device is rebooted.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	system.boot_image.set	String		

Note: (*) parameter is required in the API.

Response Fields

boot_image	XML tag of boot image
hard_disk	0: primary 1: secondary
compact_flash	0: primary 1: secondary

Example Response

Request as:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
- **HTTP Action:**
GET
- **CLI configuration before axAPI call:**
N/A
- **HTTP Body:**
session_id=xxxxxxxxxxxxxxxxxxxx&method=system.boot_image.get

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
  <boot_image>
    <hard_disk>1</hard_disk>
    <compact_flash>1</compact_flash>
  </boot_image>
</response>
```

Menus Privilege

Config Mode >> System >> Settings >> Boot

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

5.5.7 “system.boot_image.set” Method

This method allows you to set the location of the boot image from which the system image will be loaded the next time the AX device is rebooted. The AX tries to boot using the Hard Disk first. The Compact Flash is used only if the hard drive is unavailable. You can select the primary or secondary image area on each boot device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	system.boot_image.set	String		
hard_disk	primary(0) or secondary(1)	Int	0 or 1	
compact_flash	primary(0) or secondary(1)	Int	0 or 1	

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST
- CLI configuration before aXAPI call:
N/A
- HTTP Body:
session_id=xxxxxxxxxxxxxxxxxxxx&method=system.boot_image.set&hard_disk=0
- CLI configuration after aXAPI call:
N/A

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> System >> Settings >> Boot

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

5.6 SNMP Configuration

5.6.1 “system.snmp.general.get” Method

This method displays general SNMP configuration information for the AX, including the name and location of the snmp server.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	system.snmp.general.get	String		

Note: (*) parameter is required in the API.

Response Fields

snmp_general_config

snmp_status	the status of AX snmp agent, enabled or disabled.
snmp_sys_name	the system name of the snmp server.
snmp_contact	the contact of snmp agent.
snmp_location	location of this snmp agent.

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/

- **HTTP Action:**
GET

CLI configuration before aXAPI call:

```
!
snmp-server location "Beijing-china"
snmp-server contact "xli@a10networks.com.cn"
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=system.snmp.general.get
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
  <snmp_general_config>
    <snmp_status>0</snmp_status>
    <snmp_sys_name>AX2100-ABC</snmp_sys_name>
    <snmp_contact>xli@a10networks.com.cn</snmp_contact>
    <snmp_location>Beijing-china</snmp_location>
  </snmp_general_config>
</response>
```

Menus Privilege

Config Mode >> System >> SNMP

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

5.6.2 “system.snmp.general.set” Method

This method displays general information about Simple Network Management Protocol (SNMP) configurations for the AX, including the name and location of the snmp server.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	system.snmp.general.set	String		
snmp_status	disabled(0) or enabled(1)	int	0 or 1	
snmp_location	system location	String		
snmp_contact	system contact	String		

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
- **HTTP Action:**
POST

CLI configuration before aXAPI call:

```
!
snmp-server location "Beijing-china"
snmp-server contact "xli@a10networks.com.cn"
!
```

HTTP Body
session_id=xxxxxxxxxxxxxxxxxxxxxx&method=system.snmp.global.set&snmp_status=1&snmp_contact=xli@a10networks.com.cn

CLI configuration after aXAPI call:

```
!
snmp-server enable
snmp-server location "Beijing-china"
snmp-server contact "xli@a10networks.com.cn"
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> System >> SNMP

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

5.6.3 “system.snmp.community.get” Method

This method displays the SNMP community configured on the AX.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	system.snmp.community.get	String		

Note: (*) parameter is required in the API.

Response Fields

snmp_community_list	XML tag for the collection of SNMP community
snmp_community	XML tag for the SNMP community
community	the read comminity of SNMP
host	host name, ip address or ip subnet of the remote SNMP
oid	which oid is allowed to access for this community.

Example Response

Request as:

- **URL:**
[https://\[AX_IP_ADDRESS\]/services/rest/V2/](https://[AX_IP_ADDRESS]/services/rest/V2/)
- **HTTP Action:**
GET

CLI configuration before aXAPI call:

```
!
snmp-server community read private123 oid .12.3.6 remote 1.2.3.0 /24
snmp-server community read private456 oid .12.3.9 remote 1.2.5.0 /24
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxxxx&method=system.snmp.community.get

```

Response as the HTTP body:
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
  <snmp_community_list>
    <snmp_community>
      <community> private123</community>
      <host> 1.2.3.0/24</host>
      <oid>.12.3.6</oid>
    </snmp_community>
    <snmp_community>
      <community> private456</community>
      <host> 1.2.5.0/24</host>
      <oid>.12.3.9</oid>
    </snmp_community>
  </snmp_community_list>
</response>

```

Menus Privilege

Config Mode >> System >> SNMP

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

5.6.4 “system.snmp.community.add” Method

This method allows you to add an SNMP community item from the community list.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	system.snmp.community.add	String		
community (*)	snmp community read	String		
host	snmp host, if leave it empty, system will use the default setting ‘default’ which means all host.	String		
oid	snmp community object identifier	String		

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
- **HTTP Action:**
POST

CLI configuration before aXAPI call:

```
!
snmp-server community read private123 oid .12.3.6 remote 1.2.3.0 /24
snmp-server community read private456 oid .12.3.9 remote 1.2.5.0 /24
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=system.snmp.community.add&community=1234567890&host=1.1.1.0&oid=.1.2.5.6.3
```

CLI configuration after aXAPI call:

```
!
snmp-server community read private123 oid .12.3.6 remote 1.2.3.0 /24
snmp-server community read private456 oid .12.3.9 remote 1.2.5.0 /24
snmp-server community read 1234567890 oid .1.2.5.6.3 remote 1.1.1.0
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> System >> SNMP

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

5.6.5 “system.snmp.community.delete” Method

This method allows you to delete an SNMP community item from the community list.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	system.snmp.community.delete	String		
community (*)	snmp community read	String		
host	snmp host, if leave it empty, system will use the default setting ‘default’ which means all host.	String		
oid	snmp community object identifier	String		

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
- **HTTP Action:**
POST

CLI configuration before aXAPI call:

```
!
snmp-server community read private123 oid .12.3.6 remote 1.2.3.0 /24
snmp-server community read private456 oid .12.3.9 remote 1.2.5.0 /24
snmp-server community read 1234567890 oid .1.2.5.6.3 remote 1.1.1.0
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=system.snmp.community.delete&community=1234567890&
host=1.1.1.0&oid=.1.2.5.6.3
```

CLI configuration after aXAPI call:

```
!
```

```

snmp-server community read private123 oid .12.3.6 remote 1.2.3.0 /24
snmp-server community read private456 oid .12.3.9 remote 1.2.5.0 /24
!

```

Response as the HTTP body:

```

<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>

```

Menus Privilege

Config Mode >> System >> SNMP

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

5.6.6 “system.snmp.trap.get” Method

This method retrieves information for configured SNMP traps.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	system.snmp.trap.get	String		

Note: (*) parameter is required in the API.

Response Fields

snmp_trap_list XML tag for the collection of SNMP trap

snmp_trap	XML tag for the SNMP trap
community	snmp community name
host	snmp trap ip address/host
port	snmp trap port
version	snmp trap version, v1(0) or v2c(1)

Example Response

Request as:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
- **HTTP Action:**
GET

CLI configuration before aXAPI call:

```

!
snmp-server host host1 version v2c private123 udp-port 883
snmp-server host host5 version v2c private456 udp-port 883
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxx&method=system.snmp.trap.get

Response as the HTTP body:

```

<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
<snmp_trap_list>

```

```

<snmp_trap>
<community> private123</community>
<host> host1</host>
<port>883</port>
<version>1</version>
</snmp_trap>
<snmp_trap>
<community> private456</community>
<host> host5</host>
<port>883</port>
<version>1</version>
</snmp_trap>
</snmp_trap_list>
</response>

```

Menus Privilege

Config Mode >> System >> SNMP

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

5.6.7 “system.snmp trap.add” Method

This method allows you to configure an SNMP trap.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	system.dns.trap.add	String		
community (*)	snmp community	String		
host (*)	snmp host	String		
port (*)	snmp port	Int	1 - 65535	162
version	snmp version (v1, v2c)	String	v1, v2c	v1

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
snmp-server host host1  version v2c private123 udp-port 883
snmp-server host host5  version v2c private456 udp-port 883
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxx&method=system.snmp.trap.add&community=public&port=555&version=v1&host=1.2.3.4

CLI configuration after aXAPI call:

```
!
snmp-server host host1  version v2c private123 udp-port 883
snmp-server host host5  version v2c private456 udp-port 883
snmp-server host 1.2.3.4  version v1 public udp-port 555

```

```
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> System >> SNMP

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

5.6.8 “system.snmp.trap.delete” Method

This method allows you to delete an SNMP trap.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	system.snmp.trap.delete	String		
community (*)	snmp community	String		
host (*)	snmp host	String		
port (*)	snmp port	Int	1 - 65535	162
version	snmp version (v1, v2c)	String	v1, v2c	v1

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
snmp-server host host1 version v2c private123 udp-port 883
snmp-server host host5 version v2c private456 udp-port 883
snmp-server host 1.2.3.4 version v2c public udp-port 555
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=system.snmp.trap.delete&community=pub&port=555&version=v2c&host=1.2.3.4
```

CLI configuration after aXAPI call:

```
!
snmp-server host host1 version v2c private123 udp-port 883
snmp-server host host5 version v2c private456 udp-port 883
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> System >> SNMP

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

5.6.9 “system.snmp.trap_enable.get” Method

This method retrieves the control list of an SNMP trap, showing which traps are enabled and which are not.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	system.snmp.trap_enable.get	String		

Note: (*) parameter is required in the API.

Response Fields

snmp_trap_control_list

snmp_group	xml tag of snmp standard trap
if_link_up	send trap when interface link up
if_link_down	send trap when interface link down.
slb_group	xml tag of trap control list in slb group.
service_up	send trap when a service (virtual server port) up.
service_down	send trap when a service (virtual server port) down.
server_up	send trap when a real server up.
server_down	send trap when a real server down.
service_conn_limit	send trap when the maximum connection limit is hit.
service_conn_rate_limit	send trap when the maximum connection rate is hit.
service_conn_resume	send trap when the connection of service resumed.
server_conn_limit	send trap when the maximum connection limit is hit.
server_conn_rate_limit	send trap when the maximum connection rate is hit.
server_conn_resume	send trap when the connection of server resumed.
virtual_port_up	send trap when the virtual server port up.
virtual_port_down	send trap when the virtual port down.
virtual_port_conn_limit	send trap when the connection limit of virtual port is hit.
virtual_port_conn_rate_limit	send trap when the connection rate limit is hit.
virtual_server_conn_limit	send trap when the connection limit is hit.
virtual_server_conn_rate_limit	send trap when the connection rate limit is hit.
application_buff_limit	send a trap when the application buffer limit is hit.
ha_group	xml tag of trap control list in ha group.
ha_standby	send trap when ax in a ha group become standby.
ha_active	send trap when ax in a ha group become active.
ha_active_active	send trap when ax runs to active-active mode.

net_group	xml tag of trap control list in net group.
trunk_port_threshold	port in a trunk port is lower than defined threshold.
system_group	xml tag of trap control list in system group.
system_start	send a trap when the system booted up.
system_shutdown	send a trap when the system shutdown.
system_restart	send a trap when the system restart.
temperature_hight	system temperature is higher than the defined value.
fan1_down	send trap when fan1 is down.
fan2_down	send trap when fan2 is down.
an3_down	send trap when fan3 is down.
fan4_down	send trap when fan4 is down.
fan5_down	send trap when fan5 is down.
lower_power_supply_down	send trap when the lower power supply is down.
upper_power_supply_down	send trap when the upper power supply is down.
primary_hd_down	send trap when the primary hard disk was down.
secondary_hd_down	send trap when the secondary hard disk was down.
memory_usage_high	send trap when the total memory usage is high.
control_cpu_usage_high	send trap when the control cpu usage is high.
data_cpu_usage_high	send trap when the data cpu usage is high.
disk-space_usage_hith	send a trap when the disk usage is high.
drop_package	send trap when any package is dropped.

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
GET
- CLI configuration before aXAPI call:

```
!
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxxxx&method=system.snmp.trap_enable.get
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
  <snmp_trap_control_list>
    <snmp_group>
      <if_link_up>0</if_link_up>
      <if_link_down>0</if_link_down>
    </snmp_group>
    <slb_group>
      <service_up>0</service_up>
      <service_down>0</service_down>
      <server_up>0</server_up>
      <server_down>0</server_down>
      <service_conn_limit>0</service_conn_limit>
      <service_conn_rate_limit>0</service_conn_rate_limit>
      <service_conn_resume>0</service_conn_resume>
      <server_conn_limit>0</server_conn_limit>
      <server_conn_rate_limit>0</server_conn_rate_limit>
      <server_conn_resume>0</server_conn_resume>
      <virtual_port_up>0</virtual_port_up>
      <virtual_port_down>0</virtual_port_down>
      <virtual_port_conn_limit>0</virtual_port_conn_limit>
```

```

<virtual_port_conn_rate_limit>0</virtual_port_conn_rate_limit>
<virtual_server_conn_limit>0</virtual_server_conn_limit>
<virtual_server_conn_rate_limit>0</virtual_server_conn_rate_limit>
<application_buffer_limit>0</application_buffer_limit>
</slb_group>
<ha_group>
    <ha_standby>0</ha_standby>
    <ha_active>0</ha_active>
    <ha_active_active>0</ha_active_active>
</ha_group>
<net_group>
    <trunk_port_threshold>0</trunk_port_threshold>
</net_group>
<system_group>
    <system_start>0</system_start>
    <system_shutdown>0</system_shutdown>
    <system_restart>0</system_restart>
    <temperature_high>0</temperature_high>
    <fan1_down>0</fan1_down>
    <fan2_down>0</fan2_down>
    <fan3_down>0</fan3_down>
    <fan4_down>0</fan4_down>
    <fan5_down>0</fan5_down>
    <lower_power_supply_down>0</lower_power_supply_down>
    <upper_power_supply_down>0</upper_power_supply_down>
    <primary_hd_down>0</primary_hd_down>
    <secondary_hd_down>0</secondary_hd_down>
    <memory_usage_high>0</memory_usage_high>
    <control_cpu_usage_high>0</control_cpu_usage_high>
    <data_cpu_usage_high>0</data_cpu_usage_high>
    <disk_space_usage_high>0</disk_space_usage_high>
    <drop_package>0</drop_package>
</system_group>
</snmp_trap_control_list>
</response>

```

Menus Privilege

Config Mode >> System >> SNMP

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

5.6.10 “system.snmp.trap_enable.set” Method

This method allows you to enable or disable SNMP traps.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	system.dns.trap_enable.set	String		
all_traps	0 disabled, 1 enabled	Int	0 or 1	
snmp_group	0 disabled, 1 enabled	Int	0 or 1	
if_link_up	0 disabled, 1 enabled	Int	0 or 1	
if_link_down	0 disabled, 1 enabled	Int	0 or 1	
slb_group	0 disabled, 1 enabled	Int	0 or 1	
service_up	0 disabled, 1 enabled	Int	0 or 1	
service_down	0 disabled, 1 enabled	Int	0 or 1	
server_up	0 disabled, 1 enabled	Int	0 or 1	
server_down	0 disabled, 1 enabled	Int	0 or 1	
service_conn_limit	0 disabled, 1 enabled	Int	0 or 1	
service_conn_rate_limit	0 disabled, 1 enabled	Int	0 or 1	

service_conn_resume	0 disabled, 1 enabled	Int	0 or 1	
server_conn_limit	0 disabled, 1 enabled	Int	0 or 1	
server_conn_rate_limit	0 disabled, 1 enabled	Int	0 or 1	
server_conn_resume	0 disabled, 1 enabled	Int	0 or 1	
virtual_port_up	0 disabled, 1 enabled	Int	0 or 1	
virtual_port_down	0 disabled, 1 enabled	Int	0 or 1	
virtual_port_conn_limit	0 disabled, 1 enabled	Int	0 or 1	
virtual_port_conn_rate_limit	0 disabled, 1 enabled	Int	0 or 1	
virtual_server_conn_limit	0 disabled, 1 enabled	Int	0 or 1	
virtual_server_conn_rate_limit	0 disabled, 1 enabled	Int	0 or 1	
application_buffer_limit	0 disabled, 1 enabled	Int	0 or 1	
ha_group	0 disabled, 1 enabled	Int	0 or 1	
ha_standby	0 disabled, 1 enabled	Int	0 or 1	
ha_active	0 disabled, 1 enabled	Int	0 or 1	
ha_active_active	0 disabled, 1 enabled	Int	0 or 1	
net_group	0 disabled, 1 enabled	Int	0 or 1	
trunk_port_threshold	0 disabled, 1 enabled	Int	0 or 1	
system_group	0 disabled, 1 enabled	Int	0 or 1	
system_start	0 disabled, 1 enabled	Int	0 or 1	
system_shutdown	0 disabled, 1 enabled	Int	0 or 1	
system_restart	0 disabled, 1 enabled	Int	0 or 1	
temperature_high	0 disabled, 1 enabled	Int	0 or 1	
fan1_down	0 disabled, 1 enabled	Int	0 or 1	
fan2_down	0 disabled, 1 enabled	Int	0 or 1	
fan3_down	0 disabled, 1 enabled	Int	0 or 1	
fan4_down	0 disabled, 1 enabled	Int	0 or 1	
fan5_down	0 disabled, 1 enabled	Int	0 or 1	
lower_power_supply_down	0 disabled, 1 enabled	Int	0 or 1	
upper_power_supply_down	0 disabled, 1 enabled	Int	0 or 1	
primary_hd_down	0 disabled, 1 enabled	Int	0 or 1	
secondary_hd_down	0 disabled, 1 enabled	Int	0 or 1	
memory_usage_high	0 disabled, 1 enabled	Int	0 or 1	
control_cpu_usage_high	0 disabled, 1 enabled	Int	0 or 1	
data_cpu_usage_high	0 disabled, 1 enabled	Int	0 or 1	
disk_space_usage_high	0 disabled, 1 enabled	Int	0 or 1	
drop_package	0 disabled, 1 enabled	Int	0 or 1	

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=system.snmp.trap_enable.set&slb_group=1&net_group=0&fan1_down=1
```

CLI configuration after aXAPI call:

```
!
snmp-server enable traps slb
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> System >> SNMP

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

5.6.11 “system.snmp.mib.download” Method

This method allows you to download the AX Management Information Base (MIB) files.
For information about the AX MIBs, see *AX Series MIB Reference*.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	system.snmp.mib.download	String		

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
(Multipart) GET
- CLI configuration before aXAPI call:
N/A
- HTTP Body
session_id=xxxxxxxxxxxxxxxxxxxx&method=system.snmp.mib.download&file_name=AX-MIB.txt
- CLI configuration after aXAPI call:
N/A

Response as the HTTP body:

HTTP code:

500 internal err. This commonly means that your request string is invalid.
404 not found This commonly means that the resource you requested is not exist.

Menus Privilege

Config Mode >> System >> SNMP

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

5.7 Logging Configuration

5.7.1 “system.log.download” Method

The method can be used to download the system log file.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	system.log.download	String		

Note: (*) parameter is required in the API.

Response Fields

System log file.

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
(Multipart) GET
- CLI configuration before aXAPI call:
N/A
- HTTP Body
session_id=xxxxxxxxxxxxxxxxxxxx&method=system.log.download
- CLI configuration after aXAPI call:
N/A

Response as the HTTP body:

File content.

Menus Privilege

Config Mode >> System >> Settings >> Log

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

404 not found

500 internal err

5.7.2 “system.show_tech.download” Method

This method can be used to display tech support contact information.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	system.show_tech.download	String		

Note: (*) parameter is required in the API.

Response Fields

Show tech file.

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
(Multipart) GET
- CLI configuration before aXAPI call:
N/A
- HTTP Body
session_id=xxxxxxxxxxxxxxxxxxxx&method=system.log.download
- CLI configuration after aXAPI call:
N/A

Response as the HTTP body:
File content.

Menus Privilege

Monitor Mode >> System >> Logging >> Show Techsupport
Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

404 not found
500 internal err

5.7.3 “system.log.level.get” Method

This method is used to retrieve the system log level setting.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	system.log.level.get	String		

Note: (*) parameter is required in the API.

Response Fields

tag: console, buffered, email, syslog, monitor
content:
emergency: emergency or above
alert: alert or above
critical: critical or above
error: error or above
warning: warning or above
notification: notification or above
information: information or above
debug: debug or above

Example Response

Request as:
• **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
• **HTTP Action:**
GET

CLI configuration before aXAPI call:

```
!
logging syslog error
logging monitor notification
logging email critical
logging trap critical
!
```

HTTP Body
session_id=xxxxxxxxxxxxxxxxxxxxxx&method=system.log.level.get

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
<log_level>
<console>error</console>
```

```

<buffered>debug</buffered>
<email>critical</email>
<snmp>critical</snmp>
<syslog>error</syslog>
<monitor>notification</monitor>
</log_level>
</response>

```

Menus Privilege

Config Mode >> System >> Settings >> Log

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

5.7.4 “system.log.level.set” Method

The method can be used to set the system log file preferences for email, SNMP, and other events.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	system.log.level.set	String		
console	The log level for console. You can specify one of the log level listed in ‘Range’ column of this table. If you set this option to empty, all log with all level will not be record. Otherwise, the log with this level or upper level will be record.	String	empty emergency alert critical error warning notification information debug	
buffered	The log level for buffered. You can specify one of the following log level. You can specify one of the log level listed in ‘Range’ column of this table. If you set this option to empty, all log with all level will not be record. Otherwise, the log with this level or upper level will be record.	String	empty emergency alert critical error warning notification information debug	
email	The log level for email. You can specify one of the following log level. If you set this option to empty, all log with all level will not be record. Otherwise, the log with this level or upper level will be record.	String	empty emergency alert critical error warning notification information debug	
snmp	The log level for snmp. You can specify one of the following log level. You can specify one of the log level listed in ‘Range’ column of this table. If you set this option to empty, all log with all level will not be record. Otherwise, the log with this level or upper level will be record.	String	empty emergency alert critical error warning notification information debug	
syslog	The log level for syslog. You can specify one of the following log level. You can specify one of the log level listed in ‘Range’ column of this table.	String	empty emergency alert critical	

	If you set this option to empty, all log with all level will not be record. Otherwise, the log with this level or upper level will be record.		error warning notification information debug	
monitor	The log level for monitor. You can specify one of the following log level. You can specify one of the log level listed in 'Range' column of this table. If you set this option to empty, all log with all level will not be record. Otherwise, the log with this level or upper level will be record.	String	empty emergency alert critical error warning notification information debug	

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
[https://\[AX_IP_ADDRESS\]/services/rest/V2/](https://[AX_IP_ADDRESS]/services/rest/V2/)
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
logging syslog error
logging monitor notification
logging email critical
logging trap critical
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxx&method=system.log.level.set&email=empty

CLI configuration after aXAPI call:

```
!
logging syslog error
logging monitor notification
logging trap critical
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> System >> Settings >> Log

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

5.7.5 “system.log.server.get” Method

This method retrieves information about the configured log server settings.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	system.log.server.get	String		

Note: (*) parameter is required in the API.

Response Fields

log_server	XML tag for system logging server
server_list	XML tag for the collection of system logging server
server	XML tag for the system logging server
port	XML tag of server port, now, this port applies to all log servers
facility	XML tag of log facility, now, this option applies to all log servers

Example Response

Request as:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
- **HTTP Action:**
GET

CLI configuration before aXAPI call:

```
!
logging host logserver1 logserver2 port 514
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxx&method=system.log.server.get

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
  <log_server>
    <server_list>
      <server>logserver1</server>
      <server>logserver2</server>
    </server_list>
    <port>514</port>
    <facility>Local0</facility>
  </log_server>
</response>
```

Menus Privilege

Config Mode >> System >> Settings >> Log

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

5.7.6 “system.log.server.set” Method

This method allows you to set the log server list. Running this command will fully replace the old host list with the new host list.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	system.logging.server.set	String		

server_list	Comma separated host list. For example, servername1,1.2.3.4,3ff5::345 6,a10.com	String	You can specify up to 10 hosts in this list. The string length of each host name should not be longer than 40 characters.	
port	The system log server port	Int		514
facility	Facility os system log	String	Local0 – Local7	

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
`https://[AX_IP_ADDRESS]/services/rest/V2/`
- HTTP Action:
`POST`

CLI configuration before aXAPI call:

```
!
logging host logserver1 logserver2 port 514
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxxxx&method=system.log.server.set&server_list=1.2.3.4,3ff5::23
45,logServer,aaa,a10.networks.com.cn,192.168.1.1&facility=Local7&port=514
```

CLI configuration after aXAPI call:

```
!
logging host 1.2.3.4 3ff5::2345 logServer aaa a10.networks.com.cn 192.168.1.1 port 514
logging facility local7
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> System >> Settings >> Log

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

5.7.7 “system.log.buffer.get” Method

The method can be used to get the maximum number of log entries the log buffer can store.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	system.log.buffer.get	String		

Note: (*) parameter is required in the API.

Response Fields

Buffer_size the buffer size

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
logging buffered 10000
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxx&method=system.log.buffer.get

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
  <buff_size>10000</buff_size>
</response>
```

Menus Privileges

Config Mode >> System >> Settings >> Log

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

5.7.8 “system.log.buffer.set” Method

The method can be used to set the maximum number of log entries the log buffer can store.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	system.log.buffer.set	String		
buff_size	log buff size	Int	1000 - 50000	30000

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
logging buffered 10000
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxx&method=system.log.buffer.set&buff_size=20000

CLI configuration after aXAPI call:

```
!
logging buffered 20000
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> System >> Settings >> Log

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

5.7.9 “system.log.smtp.get” Method

This method can be used to get the IP address, FQDN, port, user, password and other information for an email server using Simple Message Transfer Protocol (SMTP).

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	system.log.smtp.get	String		

Note: (*) parameter is required in the API.

Response Fields

smtp

email_buff_num	the email buffer size
email_buff_time	the email buffer timeout
email_server	the host name or address of email server
email_port	the port of smtp
email_list	destination email address list
email_from	the ‘email form’ field when sending a email
email_need_auth	if the smtp needs to authenticate the client, (1) need, (0) no
smtp_user	user name for smtp authentication
smtp_passwd	password for smtp authentication

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
logging email-address xli@al0networks.com.cn plumeer@126.com aaa@aaa
!
smtp 2.3.4.5
smtp username 123 password encrypted
/+mboU9rpJM8EIy41dsA5zwQjLjV2wDnPBCMuNXbAOc8EIy41dsA5zwQjLjV2wDn
smtp mailfrom xli@al0networks.com.cn
```

```
smtp needauthentication
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxxxx&method=system.log.smtp.get
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
  <smtp>
    <email_buff_num>50</email_buff_num>
    <email_buff_time>10</email_buff_time>
    <smtp_server>2.3.4.5</smtp_server>
    <smtp_port>25</smtp_port>
    <email_list>xli@a10networks.com.cn,plumeer@126.com,aaa@aaa</email_list>
    <email_from>xli@a10networks.com.cn</email_from>
    <need_auth>1</need_auth>
    <smtp_user>123</smtp_user>
    <smtp_passwd>123456</smtp_passwd>
  </smtp>
</response>
```

Menus Privilege

Config Mode >> System >> Settings >> Log

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

5.7.10 “system.log.smtp.set” Method

This method can be used to set the IP address, FQDN, port, user, password and other information for an email server using Simple Message Transfer Protocol (SMTP).

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	system.logging.smtp.set	String		
smtp_server	system logging SMTP server name	String		
smtp_port	system logging SMTP server port	Int	1 - 65535	
email_from	the source email address	String		
email_list	destination email address list separated by comma. This list will fully replace the old email list.	String		
need_auth	system logging SMTP auth, disabled(0) or enabled(1)	Init	0 or 1	
smtp_user	system logging SMTP server username	String		
smtp_passwd	system logging SMTP password	String		
email_buff_num	the email buff size	int	16-256	
email_buff_time	the time out of items in email buff	Int	10 - 1440	

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:

POST

CLI configuration before aXAPI call:

```
!
logging email-address xli@a10networks.com.cn plumeer@126.com aaa@aaa
!
smtp 2.3.4.5
smtp username 123 password encrypted
/+mboU9rpJM8EIy41dsA5zwQjLjV2wDnPBCMuNXbAOc8EIy41dsA5zwQjLjV2wDn
smtp mailfrom xli@a10networks.com.cn
smtp needauthentication
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=system.log.smtp.set&smtp_server=mail.a10networks.com&smtp_port=25
```

CLI configuration after aXAPI call:

```
!
logging email-address xli@a10networks.com.cn plumeer@126.com aaa@aaa
!
smtp mail.a10networks.com
smtp username 123 password encrypted
/+mboU9rpJM8EIy41dsA5zwQjLjV2wDnPBCMuNXbAOc8EIy41dsA5zwQjLjV2wDn
smtp mailfrom xli@a10networks.com.cn
smtp needauthentication
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> System >> Settings >> Log

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

5.7.11 “system.log.audit.get” Method

This method is used to get the audit log settings.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	system.log.audit.get	String		

Note: (*) parameter is required in the API.

Response Fields

audit

level	the audit log level, one of ‘privilege’, ‘disable’, ‘enable’
buff_num	the buff size of audit log

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
audit enable privilege
!
HTTP Body
session_id=xxxxxxxxxxxxxxxxxxxx&method=system.log.audit.get
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
  <audit>
    <level>privilege</level>
    <buff_num>20000</buff_num>
  </audit>
</response>
```

Menus Privilege

Config Mode >> System >> Settings >> Log

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

5.7.12 “system.log.audit.set” Method

This method is used to set the audit log settings, such as audit buffer size and the types of commands to be logged.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	system.log.audit.set	String		
level	audit log level	String	disable, enable, privilege	1
buff_num	audit log buff size	int	1000 - 20000	20000

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
audit enable privilege
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=system.log.audit.set&level=disable&buff_num=3000
```

CLI configuration after aXAPI call:

```
!
no audit enable
audit size 3000
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> System >> Settings >> Log

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

5.7.13 “system.log.clear” Method

This method is used to clear all buffered logs on the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	system.log.clear	String		

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
- **HTTP Action:**
POST
- **CLI configuration before aXAPI call:**
N/A
- **HTTP Body**
session_id=xxxxxxxxxxxxxxxxxxxx&method=system.log.clear
- **CLI configuration after aXAPI call:**
N/A

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Monitor Mode >> System >> Logging >> Logging

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

5.8 System Information

5.8.1 “system.information.get” Method

This method is used to get basic system information for the AX, such as software version, firmware version, serial number, startup mode, etc.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	system.information.get	String		

Note: (*) parameter is required in the API.

Response Fields

system_information	XML tag for AX system information
serial_number	AX system serial number
current_time	AX system current time
startup_mode	AX system startup mode
software_version	current running software version
advanced_core_os_on_harddisk1	
advanced_core_os_on_harddisk2	
advanced_core_os_on_compact_flash1	
advanced_core_os_on_compact_flash2	
firmware_version	firmware version
aflex_engine_version	aFlex engine version
last_config_saved	last config save at time
technical_support	technical support

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
GET
- CLI configuration before aXAPI call:
N/A
- HTTP Body
session_id=xxxxxxxxxxxxxxxxxxxx&method=system.information.get

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
  <system_information>
    <serial_number>AX21101108040006</serial_number>
    <current_time>22:57:14 CET Thu Jan 20 2011</current_time>
    <startup_mode>hard disk secondary</startup_mode>
    <software_version>2.6.2-MGMT(build: 0)</software_version>
    <advanced_core_os_on_harddisk1>2.6.0(build: 234)</advanced_core_os_on_harddisk1>
    <advanced_core_os_on_harddisk2>2.6.2-MGMT(build: 0)</advanced_core_os_on_harddisk2>
    <advanced_core_os_on_compact_flash1>
      2.4.2(build: 35)</advanced_core_os_on_compact_flash1>
    <advanced_core_os_on_compact_flash2>
      2.6.2-MGMT(build: 0)</advanced_core_os_on_compact_flash2>
```

```

<firmware_version></firmware_version>
<aflex_engine_version>2.0.0</aflex_engine_version>
<last_config_saved>20:09:57 CET Thu Jan 20 2011</last_config_saved>
<technical_support>www.a10networks.com/support</technical_support>
</system_information>
</response>

```

Menus Privilege

Monitor Mode >> Overview >> Summary

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

5.8.2 “system.device_info.get” Method

This method is used to get basic device information for the AX, such as fan status, CPU status, and disk usage.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	system.device_info.get	String		

Note: (*) parameter is required in the API.

Response Fields

device_information
 cpu_count
 cpu_status
 cpu_temperature
 disk_usage
 fan_status
 power_supply
 memory_usage
 disk_status

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
GET
- CLI configuration before aXAPI call:
N/A
- HTTP Body
session_id=xxxxxxxxxxxxxxxxxxxx&method=system.device_info.get

Response as the HTTP body:

```

<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
  <device_information>
    <cpu_count>3</cpu_count>
    <cpu_status>ALL_OK</cpu_status>
    <cpu_temperature>34C/93F</cpu_temperature>
    <disk_status>
      <disk1>active</disk1>
      <disk2>unknow</disk2>
    </disk_status>
  </device_information>
</response>

```

```

<disk_usage>7636444KB/156288352KB</disk_usage>
<fan_status>
  <Fan1>4655</Fan1>
  <Fan2>4623</Fan2>
  <Fan3>4720</Fan3>
</fan_status>
<power_supply>
  <supply1>on</supply1>
  <supply2>off</supply2>
</power_supply>
<memory_usage>1524416KB/2073996KB</memory_usage>
</device_information>
</response>

```

Menus Privilege

Monitor Mode >> Overview >> Summary

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

5.9 System Performance

5.9.1 “system.performance.get” Method

This method is used to get system performance information, such as SSL connections per second, hit ratio, L4 connections per second, and similar performance-related information.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	system.performance.get	String		

Note: (*) parameter is required in the API.

Response Fields

performance	XML tag for system performance information
following properties:	
total_throughput:	total throughput, Bits/sec
l4:	L4 connections/sec
l7_conn:	L7 connections/sec
l7_trans:	L7 transactions/sec
ssl:	SSL connections/sec
ip_nat:	IP NAT connection/sec
total_new_conn:	total new connections per second
total_curr_conn:	total current connections
http_proxy	XML tag for system http-proxy performance information
following properties:	
total_conn:	total connections
curr_conn:	current connections
server_conn_made:	server connections made
conn_reuse	XML tag for system connection reuse performance information
following properties:	
open_per_conn:	open persistent connections
active_per_conn:	active persistent connections

compression	XML tag for system compression performance information
following properties:	
in_data_rate:	in data rate, bytes/sec
out_data_rate:	out data rate, bytes/sec
bandwidth_save:	bandwidth savings (%)
caching	XML tag for system caching performance information
following properties:	
hit_ratio:	hit ratio
total_request:	total requests
cached_obj:	cached objects
attack_pre	XML tag for system attack prevention performance information
following properties:	
total_syn_rev:	total SYNs received
total_syn_fail:	total SYN-cookie failure
health_check	XML tag for system health check summary information
following properties:	
server_up:	server status is up number
server_down:	server status is down number
port_up:	server port status is up number
port_down:	server port status is down number

Example Response

Request as:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
- **HTTP Action:**
GET
- **CLI configuration before aXAPI call:**
N/A
- **HTTP Body**
session_id=xxxxxxxxxxxxxxxxxxxxxx&method=system.performance.get

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
<performance>
    <total_throughput_bits_per_sec>39120</total_throughput_bits_per_sec>
    <14_conns_per_sec>0</14_conns_per_sec>
    <17_conns_per_sec>0</17_conns_per_sec>
    <17_trans_per_sec>0</17_trans_per_sec>
    <ssl_conns_per_sec>0</ssl_conns_per_sec>
    <ip_nat_conns_per_sec>0</ip_nat_conns_per_sec>
    <total_new_conns_pe_sec>0</total_new_conns_pe_sec>
    <total_current_conns>0</total_current_conns>
</performance>
<attack_prevention>
    <total_tcp_syn_received>0</total_tcp_syn_received>
    <Total_syn_cookie_failures>0</Total_syn_cookie_failures>
</attack_prevention>
<health_check_summary>
<servers>
    <server_up_num>0</server_up_num>
    <server_down_num>3</server_down_num>
</servers>
<ports>
    <port_up_num>0</port_up_num>
    <port_down_num>2</port_down_num>
</ports>
</health_check_summary>
<http_proxy>
```

```

<total_conns>0</total_conns>
<current_conns>0</current_conns>
<server_conns_made>0</server_conns_made>
</http_proxy>
<conn_reuse>
  <open_persistent_conns>0</open_persistent_conns>
  <active_persistent_conns>0</active_persistent_conns>
</conn_reuse>
<compression>
  <in_data_rate_bytes_per_sec>0</in_data_rate_bytes_per_sec>
  <out_dat_rate_bytes_per_sec>0</out_dat_rate_bytes_per_sec>
  <bandwidth_savings>0</bandwidth_savings>
</compression>
<caching>
  <hit_ratio>0</hit_ratio>
  <total_requests>0</total_requests>
  <cached_objects>0</cached_objects>
</caching>
</response>

```

Menus Privilege

Monitor Mode >> Overview >> Performance >> Summary

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

5.10 System Resource Usage

5.10.1 “system.resource.get” Method

This method is used to get system resource information for the AX, such as L4 sessions, total SSL templates, and total virtual ports.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	system.resource.get	String		

Note: (*) parameter is required in the API.

Response Fields

client_ssl_template_count:	Total configurable Client SSL Templates
conn_reuse_template_count:	Total configurable Connection reuse Templates
fast_tcp_template_count:	Total configurable Fast TCP Templates
fast_udp_template_count:	Total configurable Fast UDP Templates
http_template_count:	Total configurable HTTP Templates
l4_session_count:	Total Sessions in the System
nat_pool_addr_count:	Total configurable NAT Pool addresses
persist_cookie_template_count:	Total configurable Persistent cookie Templates
persist_srcip_template_count:	Total configurable Source IP Persistent Templates
proxy_template_count:	Total configurable Proxy Templates
real_port_count:	Total Real Server Ports in the System
server_count:	Total Real Servers in the System
server_ssl_template_count:	Total configurable Server SSL Templates
service_group_count:	Total Service Groups in the System

stream_template_count:	Total configurable Streaming media
virtual_port_count:	Total Virtual Server Ports in the System
virtual_server_count:	Total Virtual Servers in the System

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
GET
- CLI configuration before axAPI call:
N/A
- HTTP Body
session_id=xxxxxxxxxxxxxxxxxxxxxx&method=system.resource.get

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
<resource>
  <client_ssl_template_count>256</client_ssl_template_count>
  <conn_reuse_template_count>256</conn_reuse_template_count>
  <fast_tcp_template_count>256</fast_tcp_template_count>
  <fast_udp_template_count>256</fast_udp_template_count>
  <http_template_count>256</http_template_count>
  <l4_session_count>1048576</l4_session_count>
  <nat_pool_addr_count>500</nat_pool_addr_count>
  <persist_cookie_template_count>256</persist_cookie_template_count>
  <persist_srcip_template_count>256</persist_srcip_template_count>
  <proxy_template_count>256</proxy_template_count>
  <real_port_count>2048</real_port_count>
  <server_count>1024</server_count>
  <server_ssl_template_count>256</server_ssl_template_count>
  <service_group_count>512</service_group_count>
  <stream_template_count>256</stream_template_count>
  <virtual_port_count>512</virtual_port_count>
  <virtual_server_count>512</virtual_server_count>
</resource>
</response>
```

Menus Privilege

Monitor Mode >> Overview >> Summary

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

5.10.2 “system.resource.set” Method

This method is used to set the system resource limitations.

Note: This method always return an error message, even if the operation is done successfully. If the error code is 999 and the internal error code is 654311441, then do not worry about this. This error just indicates that the change will come into effect the next time you reload the system.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	system.resource.set	String		
l4-session-count	Total Sessions in the System	Long	131072-8388608	1048576

real-server-count	Total Real Servers in the System	Long	512-2048	1024
real-port-count	Total Real Server Ports in the System	Long	512-4096	2048
service-group-count	Total Service Groups in the System	Long	512-1024	512
virtual-server-count	Total Virtual Servers in the System	Long	512-1024	512
virtual-port-count	Total Virtual Server Ports in the System	Long	256-1024	512
http-template-count	Total configurable HTTP Templates	Long	32-1024	256
proxy-template-count	Total configurable Proxy Templates	Long	32-1024	256
conn_reuse-template-count	Total configurable Connection reuse Templates	Long	32-1024	256
fast-tcp-template-count	Total configurable Fast TCP Templates	Long	32-1024	256
fast-udp-template-count	Total configurable Fast UDP Templates	Long	32-1024	256
client_ssl-template-count	Total configurable Client SSL Templates	Long	32-1024	256
server_ssl-template-count	Total configurable Server SSL Templates	Long	32-1024	256
stream-template-count	Total configurable Streaming media	Long	32-1024	256
persist-cookie-template-count	Total configurable Persistent cookie Templates	Long	32-1024	256
persist-srcip-template-count	Total configurable Source IP Persistent Templates	Long	32-1024	256
nat-pool-addr-count	Total configurable NAT Pool addresses	Long	131072-8388608	500

Note: (*) parameter is required in the API.

In the above table, the default value and value range depend on your AX model and hardware type. For example, the value range of AX 5200 may be larger than the range of AX 3000, and the one with more memory in two AX 2000 boxes would have the larger value range.
The above table only shows data for the AX 2000, which has 2G memory.

Response Fields

General XML status response

Example Response

Request as:

- **URL:**
`https://[AX_IP_ADDRESS]/services/rest/V2/`
- **HTTP Action:**
`POST`
- **CLI configuration before aXAPI call:**

```
!
!
```

- **HTTP Body**
`session_id=xxxxxxxxxxxxxxxxxxxx&method=system.resource.set&l4-session-count=132000`
- **CLI configuration after aXAPI call:**

```
!
system resource-usage l4-session-count 132000
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Monitor Mode >> Overview >> Summary

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

5.10.3 “system.config_status.get” Method

The first method is **system.config_status.get** and it does not require you to enter any additional parameters. This method retrieves the configuration status of the current partition.

Response Fields

The method returns the **config_status** object, which contains fields shown the table below.

TABLE config_status object fields

Field	Description
partition_name	Name of partition
config_dirty	0: If partition is saved 1: If partition is not saved
last_saved_time	UTC time when partition was last saved 1
last_modified_time	UTC time when partition was last modified

1. This represents the number of seconds that have elapsed since 1970/1/1 00:00:00.

Example Response (XML)

Request as:

- URL:
[https://\[AX_IP_ADDRESS\]/services/rest/V2/](https://[AX_IP_ADDRESS]/services/rest/V2/)
- HTTP Action:
GET
- HTTP Body
session_id=xxxxxxxxxxxxxxxxxxxx&method=system.config_status.get

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
  <config_status>
    <partition_name>aa</partition_name>
    <config_dirty>1</config_dirty>
    <last_saved_time>1328909513</last_saved_time>
    <last_modified_time>1328941727</last_modified_time>
  </config_status>
</response>
```

Example Response (JSON)

Request as:

- URL:
[https://\[AX_IP_ADDRESS\]/services/rest/V2/](https://[AX_IP_ADDRESS]/services/rest/V2/)
- HTTP Action:
GET
- HTTP Body
session_id=xxxxxxxxxxxxxxxxxxxx&method=system.config_status.get&format=json

Response as the HTTP body:

```
{
  "config_status": {
    "partition_name": "aa",
    "config_dirty": 1,
    "last_saved_time": 1328909513,
    "last_modified_time": 1328941727
  }
}
```

Error Codes

TBD

5.10.4 “system.config_status.getAll” Method

This method retrieves the configuration status for all partitions to which the admin has access. This may not necessarily mean every partition on the AX device, but rather, every partition that this administrator can access. This method does not require the entry of additional parameters.

TABLE Admin types and partition access levels

Admin type	Partition access
Root admin	Access to all partitions configured on AX device
Common admin	Same access as 'Root admin'
Partition user	Access to shared partition and home partition
Multiple-partition admin	Access to shared partitions, the home partition, and other partitions. Users may have additional priorities and can thus access additional partitions if they have been authenticated using TACACS+. For details, refer to “Authorization Based on Private Partition” in the <i>AX Series System Configuration and Administration Guide</i> .

Response Fields

The table below lists the response fields returned by the **config_status_summary** object.

TABLE config_status_summary object fields

Field	Description
config_dirty_summary	<ul style="list-style-type: none"> • 0: If all partition have been saved • 1: If one or more partitions have not been saved
shared_partition_config_dirty	<ul style="list-style-type: none"> • 0: If the shared partition has been saved • 1: If the shared partition has not been saved
current_partition_name	Name of the current partition
current_partition_config_dirty	<ul style="list-style-type: none"> • 0: If the current partition has been saved • 1: If the current partition has not been saved
default_partition_name	Name of the home partition
default_partition_config_dirty	<ul style="list-style-type: none"> • 0: If the home partition has been saved • 1: If the home partition has not been saved
partition_config_status_list	This contains configuration status for all partitions. This object lists other objects returned by the config_status object, which is returned by the system.config_status.get method.

Example Response (XML)

Request as:

- **URL:**
[https://\[AX IP ADDRESS\]/services/rest/v2/](https://[AX IP ADDRESS]/services/rest/v2/)
- **HTTP Action:**
GET
- **HTTP Body**
session_id=xxxxxxxxxxxxxxxxxxxx&method=system.config_status.getAll

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?><response status="ok">
<config_status_summary>
  <config_dirty_summary>1</config_dirty_summary>
  <shared_partition_config_dirty>0</shared_partition_config_dirty>
  <current_partition_name>aa</current_partition_name>
  <current_partition_config_dirty>1</current_partition_config_dirty>
  <default_partition_name>aa</default_partition_name>
  <default_partition_config_dirty>1</default_partition_config_dirty>
<partition_config_status_list>
```

```

<config_status>
    <partition_name>shared</partition_name>
    <config_dirty>0</config_dirty>
    <last_saved_time>1328940966</last_saved_time>
    <last_modified_time>1328940963</last_modified_time>
</config_status>
<config_status>
    <partition_name>aa</partition_name>
    <config_dirty>1</config_dirty>
    <last_saved_time>1328909513</last_saved_time>
    <last_modified_time>1328941727</last_modified_time>
</config_status>
<config_status>
    <partition_name>bb</partition_name>
    <config_dirty>0</config_dirty>
    <last_saved_time>1328909513</last_saved_time>
    <last_modified_time>1328909513</last_modified_time>
</config_status>
</partition_config_status_list>
</config_status_summary>
</response>

```

Example Response (JSON)

Request as:

- URL:
[https://\[AX_IP_ADDRESS\]/services/rest/V2/](https://[AX_IP_ADDRESS]/services/rest/V2/)
- HTTP Action:
GET
- HTTP Body
session_id=xxxxxxxxxxxxxxxxxxxxx&method=system.config_status.getAll&format=json

Response as the HTTP body:

```
{
  "config_status_summary": {
    "config_dirty_summary": 1,
    "shared_partition_config_dirty": 0,
    "current_partition_name": "aa",
    "current_partition_config_dirty": 1,
    "default_partition_name": "aa",
    "default_partition_config_dirty": 1,
    "partition_config_status_list": [
      {
        "partition_name": "shared",
        "config_dirty": 0,
        "last_saved_time": 1328940966,
        "last_modified_time": 1328940963
      },
      {
        "partition_name": "aa",
        "config_dirty": 1,
        "last_saved_time": 1328909513,
        "last_modified_time": 1328941727
      },
      {
        "partition_name": "bb",
        "config_dirty": 0,
        "last_saved_time": 1328909513,
        "last_modified_time": 1328909513
      }
    ]
  }
}
```

Error Codes

TBD

6 Network Management APIs

6.1 Interface

6.1.1 “network.interface.get” Method

This method is used to get the network settings for an interface, such as name, MAC address, and flow control.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	network.interface.get	String		
port_num(*)	Data interface number	Int	Depends on AX model	

Note: (*) parameter is required in the API.

Response Fields

interface	XML tag for interfaces config info
port_num	interface number
type	interface type, always be ‘ethernet’
name	interface name
status	interface status, disabled(0) or enabled(1)
mac_addr	interface MAC address
duplexity	interface duplexity setting, ‘auto’, ‘half’ or ‘full’
speed	interface speed setting, ‘auto’, ‘10M’, ‘100M’ or ‘1G’
flow_ctl	flow control
normal_rate	ICMP normal rate, if this option is set, ICMP rate limit is enabled, if this option is not set, ICMP rate limit is disabled on this virtual port.
lockup_rate	ICMP lockup rate
lockup_period	ICMP lockup period
ipv4_addr_list	XML tag for the collection of interface IPv4 info list
ipv4	XML tag for the collection of interface IPv4 info
ipv4_addr	IPv4 address of this interface
ipv4_mask	IPv4 mask
ipv4_acl	IPv4 access list
ipv6_addr_list	XML tag for the collection of interface IPv6 info list
ipv6	XML tag for interfaces IPv6 info
ipv6_addr	interface IPv6 address
ipv6_prefix_len	interface IPv6 prefix length
ipv6_is_any_cast_addr	yes(1), no(0)
ipv6_auto_link_local	whether to link local address automatic or not
ipv6_link_local_config	XML tag for interfaces IPv6 link local configuration.
ipv6_link_local_addr	IPv6 link local address
ipv6_link_local_prefix	IPv6 link local prefix length
ipv6_link_local_is_any_cast	yes(1), no(0)
ipv6_acl	IPv6 access list
allow_promiscuous_vip	allow promiscuous VIP status, disabled(0) or enabled(1)
tcp_sync_cookie	TCP syn cookie status, disabled(0) or enabled(1)
ha_status	interface HA status, disabled(0) or enabled(1)
ha_type	none(0), router-interface(1), server-interface(2) or both(3)
ha_heartbeat	status when HA status enabled, disabled(0) or enabled(1)

ha_vlan

vlan id, only when interface HA status is enabled(1)

Example Response

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
access-list 101 permit ip any any
!
ipv6 access-list test
  permit ipv6 any any
!
interface ethernet 1
  ip address 124.254.1.1 255.255.255.0
  ip address 124.254.1.3 255.0.0.0
  ip address 124.254.1.5 255.255.128.0
  ipv6 address 3ff1::3/64 anycast
  ipv6 address 2ab4:33f5:4565:a179::443/80
  ipv6 address fe80::1111/64 link-local anycast
  ip allow-promiscuous-vip
  access-list 101 in
  ipv6 access-list test in
  icmp-rate-limit 60000 lockup 60000 1000
  flow-control
!
ha interface ethernet 1 router-interface vlan 1
!
```

HTTP Body

session_id=123456789&method=network.interface.get&port_num=1

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
  <interface>
    <port_num>1</port_num>
    <type>ethernet</type>
    <name>data interface 1</name>
    <status>1</status>
    <mac_addr>0090.0B0C.838B</mac_addr>
    <duplexity>auto</duplexity>
    <speed>auto</speed>
    <flow_ctl>1</flow_ctl>
    <normal_rate>60000</normal_rate>
    <lookup_rate>60000</lookup_rate>
    <lookup_period>1000</lookup_period>
    <ipv4_addr_list>
      <ipv4>
        <ipv4_addr>124.254.1.1</ipv4_addr>
        <ipv4_mask>255.255.255.0</ipv4_mask>
      </ipv4>
      <ipv4>
        <ipv4_addr>124.254.1.3</ipv4_addr>
        <ipv4_mask>255.0.0.0</ipv4_mask>
      </ipv4>
      <ipv4>
        <ipv4_addr>124.254.1.5</ipv4_addr>
        <ipv4_mask>255.255.128.0</ipv4_mask>
      </ipv4>
      <ipv4_acl>101</ipv4_acl>
    </ipv4_addr_list>
    <ipv6_addr_list>
    <ipv6>
```

```

<ipv6_addr>3fff1::3</ipv6_addr>
<ipv6_prefix_len>64</ipv6_prefix_len>
<ipv6_is_any_cast_addr>1</ipv6_is_any_cast_addr>
</ipv6>
<ipv6>
<ipv6_addr>2ab4:33f5:4565:a179::443</ipv6_addr>
<ipv6_prefix>80</ipv6_prefix>
<ipv6_is_any_cast_addr>0</ipv6_is_any_cast_addr>
</ipv6>
</ipv6_addr_list>
<ipv6_auto_link_local>0</ipv6_auto_link_local>
<ipv6_link_local_cfg>
<ipv6_link_local_addr>fe80::1111</ipv6_link_local_addr>
<ipv6_link_local_prefix>64</ipv6_link_local_prefix>
<ipv6_link_local_is_any_cast>1</ipv6_link_local_is_any_cast>
</ipv6_link_local_cfg>
<ipv6_acl>test</ipv6_acl>
<allow_promiscuous_vip>1</allow_promiscuous_vip>
<tcp_sync_cookie>0</tcp_sync_cookie>
<ha_status>1</ha_status>
<ha_type>router_if</ha_type>
<ha_heartbeat>0</ha_heartbeat>
<ha_vlan>1</ha_vlan>
</interface>
</response>

```

Menus Privilege

Config Mode >> Network >> Interface >> LAN

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

6.1.2 “network.interface.getAll” Method

This method is used to get the network settings for all data interfaces, such as name, MAC address, and flow control. This method requires going through each data interface using the network.interface.get method.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	network.interface.getAll	String		

Note: (*) parameter is required in the API.

Response Fields

Interface_list	XML tag for data interface list
interface	XML tag for interfaces config info
port_num	interface number
type	interface type, always be ‘ethernet’
name	interface name
status	interface status, disabled(0) or enabled(1)
mac_addr	interface MAC address
duplexity	interface duplexity setting, ‘auto’, ‘half’ or ‘full’
speed	interface speed setting, ‘auto’, ‘10M’, ‘100M’ or ‘1G’
flow_ctl	flow control

normal_rate	ICMP normal rate, if option is set, ICMP rate limit is enabled, if this option is not set, ICMP rate limit is disabled on this virtual port.
lockup_rate	ICMP lockup rate
lockup_period	ICMP lockup period
ipv4_addr_list	XML tag for the collection of interface IPv4 info list
ipv4	XML tag for the collection of interface IPv4 info
ipv4_addr	IPv4 address of this interface
ipv4_mask	IPv4 mask
ipv4_acl	IPv4 access list
ipv6_addr_list	XML tag for the collection of interface IPv6 info list
ipv6	XML tag for interfaces IPv6 info
ipv6_addr	interface IPv6 address
ipv6_prefix_len	interface IPv6 prefix length
ipv6_is_any_cast_addr	yes(1), no(0)
ipv6_auto_link_local	whether to link local address automatic or not
ipv6_link_local_config	XML tag for interfaces IPv6 link local configuration.
ipv6_link_local_addr	IPv6 link local address
ipv6_link_local_prefix	IPv6 link local prefix length
ipv6_link_local_is_any_cast	yes(1), no(0)
ipv6_acl	IPv6 access list
allow_promiscuous_vip	allow promiscuous VIP status, disabled(0) or enabled(1)
tcp_sync_cookie	TCP syn cookie status, disabled(0) or enabled(1)
ha_status	interface HA status, disabled(0) or enabled(1)
ha_type	none(0), router-interface(1), server-interface(2) or both(3)
ha_heartbeat	status, when HA status is enabled, disabled(0) or enabled(1)
ha_vlan	vlan id, only when interface HA status is enabled(1)

Example Response

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
interface ethernet 12
 ip allow-promiscuous-vip
 flow-control
!
```

HTTP Body

session_id=123456789&method=network.interface.getAll

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
<interface_list>
<interface>
<interface>
<interface>
<interface>
<interface>
<interface>
```

```

<interface>
<interface>
<interface>
<interface>
<interface>
<interface>
<port_num>12</port_num>
<type>ethernet</type>
<name />
<status>0</status>
<mac_addr>0090.0B0C.83F0</mac_addr>
<duplexity>auto</duplexity>
<speed>auto</speed>
<flow_ctl>1</flow_ctl>
<ipv4_addr_list />
<ipv6_auto_link_local>0</ipv6_auto_link_local>
<ipv6_addr_list />
<ipv6_link_local_cfg />
<ipv6_acl>test</ipv6_acl>
<allow_promiscuous_vip>0</allow_promiscuous_vip>
<tcp_sync_cookie>0</tcp_sync_cookie>
<ha_status>0</ha_status>
<ha_type>none</ha_type>
<ha_heartbeat>0</ha_heartbeat>
<ha_vlan>0</ha_vlan>
<ipv4_acl>0</ipv4_acl>
</interface>
</interface_list>
</response>

```

Menus Privilege

Config Mode >> Network >> Interface >> LAN

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

6.1.3 “network.interface.set” Method

This method is used to configure the network settings for a data interface, such as name, MAC address, and flow control.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	network.interface.set	String		
name	Interface name	String		“”
port_num (*)	Interface number.	Int		
status	Interface status, disabled(0) or enabled(1)	Int	0 or 1	0
speed	Interface speed, auto, 10M, 100M or 1G	String	‘auto’, ‘10M’, ‘100M’, ‘1G’	‘auto’
duplexity	Interface duplexity, auto, half or full	String	‘auto’, ‘half’, ‘full’	‘auto’
flow_ctl	Flow control	Int	0 or 1	
normal_rate	Interface icmp normal rate, if not set, icmp rate limit will be sidabled	Int	1 - 65535	
lockup_rate	Interface icmp lockup rate	Int	1 - 65535	
lockup_period	Interface icmp lockup period	Int	1 - 16383	
I3_vlan_forward_disable	I3 vlan forward disable, disabled(0) or enabled(1)	Int	0 or 1	
ipv4_acl	IPv4 access list	Int	1-199, predefined	
ipv6_acl	IPv6 access list name	String	Predefined	
ipv6_link_local_addr	IPv6 link local address	String		
ipv6_link_local_prefix_len	IPv6 link local prefix length	Int	1-128	

ipv6_link_local_is_any_cast	Whether the link local address is a anycast address, yes(1), no(0)	Int	0 or 1	0
allow_promiscuous_vip	Allow promiscuous vip, disabled(0) or enabled(1)	Int	0 or 1	0
tcp_syn_cookie	Tcp syn cookie, disabled(0) or enabled(1)	Int	0 or 1	0
ha_status	Interface HA status, disabled(0) or enabled(1)	Int	0 or 1	0
ha_type	The type of HA interface. Possible value is none: not configured router_if: route HA interface server_if: server HA interface both: route-server HA interface	String	'none' 'router_if' 'server_if' 'both'	'none'
ha_heartbeat	Interface HA heartbeat, disabled(0) or enabled(1)	Int	0 or 1	0
ha_vlan	Interface HA vlan id	Int	1 - 4094	

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
interface ethernet 2
 ip address 22.22.22.22 255.255.255.0
 disable
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxxx&method=network.interface.set&port_num=2&status=1&name=tethernet_02&speed=auto&duplexity=full&flow_ctl=1&normal_rate=100&lockup_rate=101&lockup_period=101&acl_id=101&acl_name=test&ipv6_link_local_addr=fe80::2222&ipv6_link_local_prefix=64&ipv6_link_local_is_anycast=1&allow_promiscuous_vip=1&tcp_syn_cookie=1&ha_status=1&ha_type=router_if&ha_heartbeat=1&ha_vlan=2
```

CLI configuration after aXAPI call:

```
!
interface ethernet 2
 ip address 22.22.22.22 255.255.255.0
 ip allow-promiscuous-vip
 ip tcp syn-cookie
 icmp-rate-limit 100 lockup 101 101
 name "testtethernet_02"
 flow-control
 duplexity Full
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Network >> Interface >> LAN

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

6.1.4 “network.interface.ipv4.add” Method

This method is used to add an IPv4 interface.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	network.interface.ipv4.add	String		
port_num(*)	The interface number	Int		
ipv4_addr(*)	IPv4 address	String		“”
ipv4_mask(*)	IPv4 mask	String		“”

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
interface ethernet 2
 disable
!
```

HTTP Body

session_id=123456789&method=network.interface.ipv4.add&port_num=2&ipv4_addr=124.254.6.6&ipv4_mask=255.255.0.0

CLI configuration after aXAPI call:

```
!
interface ethernet 2
 ip address 124.254.6.6 255.255.0.0
 disable
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Network >> Interface >> LAN

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

6.1.5 “network.interface.ipv4.delete” Method

This method is used to delete an IPv4 interface.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	network.interface.ipv4.delete	String		
port_num(*)	The interface number	Int		
ipv4_addr(*)	IPv4 address	String		“”

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
interface ethernet 2
 ip address 124.254.6.6 255.255.0.0
 disable
!
```

HTTP Body
session_id=123456789&method=network.interface.ipv4.delete&port_num=2&ipv4_addr=124.254.6.6

CLI configuration after aXAPI call:

```
!
interface ethernet 2
 disable
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Network >> Interface >> LAN

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

6.1.6 “network.interface.ipv6.add” Method

This method is used to add an IPv6 interface.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	network.interface.ipv6.add	String		
port_num(*)	The interface number	Int		
ipv6_addr (*)	IPv6 address	String		"
ipv6_prefix_len(*)	IPv6 prefix length	Int		
ipv6_is_any_cast_addr	Whether it is a IPv6 any cast address	Int	0 or 1	0

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
interface ethernet 2
  disable
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=network.interface.ipv6.add&port_num=2&ipv6_addr=4f
f6::5678&ipv6_prefix_len=64&ipv6_is_any_cast_addr=1
```

CLI configuration after aXAPI call:

```
!
interface ethernet 2
  ipv6 address 4ff6::5678/64  anycast
  disable
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Network >> Interface >> LAN

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

6.1.7 “network.interface.ipv6.delete” Method

This method is used to delete an IPv6 interface.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		

method (*)	network.interface.ipv6.delete	String		
port_num(*)	The interface number	Int		
ipv6_addr (*)	IPv6 address	String		"

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
interface ethernet 2
 ipv6 address 4ff6::5678/64 anycast
 disable
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=network.interface.ipv6.delete&port_num=2&ipv6_addr=4ff6::5678
```

CLI configuration after aXAPI call:

```
!
interface ethernet 2
 ipv6 enable
 disable
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Network >> Interface >> LAN

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

6.1.8 “network.interface.global.get” Method

This method is used to get the global network interface settings.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	network.interface.global.get	String		

Note: (*) parameter is required in the API.

Response Fields

if_global	XML tag for interface global info
tcp_sync_cookie_threshold	TCP sync cookie threshold
l3_vlan_forward_disable	disabled(0) or enabled(1)

Example Response

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
ip tcp syn-cookie threshold 5
l3-vlan-fwd-disable
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxx&method=network.interface.global.get

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
  <interface_global>
    <tcp_sync_cookie_threshold>5</tcp_sync_cookie_threshold>
    <l3_vlan_forward_disable>1</l3_vlan_forward_disable>
  </interface_global>
</response>
```

Menus Privilege

Config Mode >> Network >> Interface >> Global

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

6.1.9 “network.interface.global.set” Method

This method is used to set global interface settings, such as threshold for TCP handshakes and L3 VLAN forwarding.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	network.interface.global.set	String		
tcp_sync_cookie_threshold	tcp sync cookie threshold	Int	1 – 100	
l3_vlan_forward_disable	disabled(0) or enabled(1)	Int	0 or 1	

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:

- https://_AX_IP_Address_/services/rest/V2/
 • HTTP Action:
 POST

CLI configuration before aXAPI call:

```
!
ip tcp syn-cookie threshold 5
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=network.interface.global.set&tcp_sync_cookie_threshold=7
```

CLI configuration after aXAPI call:

```
!
ip tcp syn-cookie threshold 7
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Network >> Interface >> Global

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

6.2 Virtual Interface

6.2.1 “network.ve.getAll” Method

This method is used to get configuration information for all virtual interfaces (VEs). The output is the collection of all network.ve.get methods for each VE.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	network.ve.getAll	String		

Note: (*) parameter is required in the API.

Response Fields

ve_list	XML tag for the collection of virtual interfaces info
ve	XML tag for virtual interface info
port_num	virtual interface number
name	virtual interface name
status	virtual interface status, disabled(0) or enabled(1)
normal_rate	icmp normal rate limit, if option is set, icmp rate limit is enabled, if option is not set, icmp rate limit is disabled on this virtual port.

lookup_rate	icmp lookup rate.
lookup_period	icmp lookup period.
ipv4_addr_list	XML tag of IPv4 address list.
ipv4	XML tag of IPv4
Ipv4_addr	IPv4 address
Ipv4_mask	IPv4 mask
ipv4_acl	IPv4 access list
ipv6_addr_list	XML tag of IPv6 address list
ipv6_addr	XML tag of IPv6 address
ipv6_addr:	IPv6 address
ipv6_prefix_len:	IPv6 prefix length
ipv6_is_any_cast_addr	any cast status of address, enabled (1), disabled (0)
ipv6_auto_link_local	automatic link local status, enabled(1), disabled(0)
ipv6_link_loal_cfg	xml tag of manual IPv6 link local address.
ipv6_link_local_addr	IPv6 link local address
ipv6_link_local_prefix_len	the prefix length of this IPv6 address
ipv6_link_local_is_any_cast	any cast status of address. Enabled (1), disabled (0)
ipv6_acl	IPv6 access name
allow_promiscuous_vip	status, enabled (1), disabled (0)
tcp_sync_cookie	TCP sync cookie status, enabled (1), disabled (0)

Example Response

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
vlan 127
tagged ethernet 8
router-interface ve 127
!
vlan 128
tagged ethernet 7
router-interface ve 128
!
access-list 101 permit ip any any
!
interface ve 127
interface ve 127
ip address 30.22.45.177 255.255.255.0
ip address 17.25.254.13 255.255.255.0
ip address 17.25.254.14 255.255.255.0
ipv6 address 3024::1234/64 anycast
ipv6 address 3211:2345:3456::5555/54
ipv6 address fe80::1113/46 link-local anycast
ip allow-promiscuous-vip
access-list 101 in
icmp-rate-limit 98 lockup 99 100
name "test ve 127"
!
interface ve 128
ip address 22.55.66.33 255.0.0.0
ip address 22.55.66.21 255.0.0.0
ip address 19.22.5.63 255.255.0.0
name "test vlan 128"
!
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxxxx&method=network.ve.getAll
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
  <ve_list>
    <ve>
      <port_num>127</port_num>
      <name>test ve 127</name>
      <status>1</status>
      <normal_rate>98</normal_rate>
      <lookup_rate>99</lookup_rate>
      <lookup_period>100</lookup_period>
      <ipv4_addr_list>
        <ipv4>
          <ipv4_addr>30.22.45.177</ipv4_addr>
          <ipv4_mask>255.255.255.0</ipv4_mask>
        </ipv4>
        <ipv4>
          <ipv4_addr>17.25.254.13</ipv4_addr>
          <ipv4_mask>255.255.255.0</ipv4_mask>
        </ipv4>
        <ipv4>
          <ipv4_addr>17.25.254.14</ipv4_addr>
          <ipv4_mask>255.255.255.0</ipv4_mask>
        </ipv4>
      </ipv4_addr_list>
      <ipv4_acl>101</ipv4_acl>
      <ipv6_addr_list>
        <ipv6>
          <ipv6_addr>3024::1234</ipv6_addr>
          <ipv6_prefix_len>64</ipv6_prefix_len>
          <ipv6_is_any_cast_addr>1</ipv6_is_any_cast_addr>
        </ipv6>
        <ipv6>
          <ipv6_addr>3211:2345:3456::5555</ipv6_addr>
          <ipv6_prefix_len>54</ipv6_prefix_len>
          <ipv6_is_any_cast_addr>0</ipv6_is_any_cast_addr>
        </ipv6>
      </ipv6_addr_list>
      <ipv6_auto_link_local>0</ipv6_auto_link_local>
      <ipv6_link_local_cfg>
        <ipv6_link_local_addr>fe80::1113</ipv6_link_local_addr>
        <ipv6_link_local_prefix_len>46</ipv6_link_local_prefix_len>
        <ipv6_link_local_is_any_cast>1</ipv6_link_local_is_any_cast>
      </ipv6_link_local_cfg>
      <allow_promiscuous_vip>1</allow_promiscuous_vip>
      <tcp_sync_cookie>0</tcp_sync_cookie>
    </ve>
    <ve>
      <port_num>128</port_num>
      <name>test vlan 128</name>
      <status>1</status>
      <ipv4_addr_list>
        <ipv4>
          <ipv4_addr>22.55.66.33</ipv4_addr>
          <ipv4_mask>255.0.0.0</ipv4_mask>
        </ipv4>
        <ipv4>
          <ipv4_addr>22.55.66.21</ipv4_addr>
          <ipv4_mask>255.0.0.0</ipv4_mask>
        </ipv4>
        <ipv4>
          <ipv4_addr>19.22.5.63</ipv4_addr>
        </ipv4>
      </ipv4_addr_list>
    </ve>
  </ve_list>
</response>
```

```

<ipv4_mask>255.255.0.0</ipv4_mask>
</ipv4>
</ipv4_addr_list>
<ipv4_acl>0</ipv4_acl>
<ipv6_addr_list />
<ipv6_auto_link_local>0</ipv6_auto_link_local>
<ipv6_link_local_cfg />
<allow_promiscuous_vip>0</allow_promiscuous_vip>
<tcp_sync_cookie>0</tcp_sync_cookie>
</ve>
</ve_list>
</response>

```

Menus Privilege

Config Mode >> Network >> Interface >> Virtual

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

6.2.2 “network.ve.get” Method

This method is used to get the configuration settings for the AX device’s Virtual Ethernet (VE) data ports.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	network.ve.get	String		
port_num(*)	Interface number	Int		

Note: (*) parameter is required in the API.

Response Fields

ve	XML tag for virtual interface info
port_num	virtual interface number
name	virtual interface name
status	virtual interface status, disabled(0) or enabled(1)
normal_rate	icmp normal rate limit, if this option us set, icmp rate limit is enabled, if this option is not set, icmp rate limit is disabled on this virtual port.
lookup_rate	icmp lookup rate.
lookup_period	icmp lookup period.
ipv4_addr_list	XML tag of IPv4 address list.
ipv4	XML tag of IPv4
Ipv4_addr	IPv4 address
Ipv4_mask	IPv4 mask
Ipv4_acl	IPv4 access list
ipv6_addr_list	XML tag of IPv6 address list
ipv6_addr	xml tag of IPv6 address
ipv6_addr:	IPv6 address
ipv6_prefix_len:	IPv6 prefix length
ipv6_is_any_cast_addr	any cast status of address, enabled (1), disabled (0)
ipv6_auto_link_local	automatic link local status, enabled(1), disabled(0)

ipv6_link_loal_cfg	xml tag of manual IPv6 link local address.
 ipv6_link_local_addr	IPv6 link local address
 ipv6_link_local_prefix_len	the prefix length of this IPv6 address
 ipv6_link_local_is_any_cast	any cast status of address. Enabled (1), disabled (0)
 ipv6_acl	IPv6 access name
 allow_promiscuous_vip	status, enabled (1), disabled (0)
 tcp_sync_cookie	TCP sync cookie status, enabled (1), disabled (0)

Example Response

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
vlan 127
tagged ethernet 8
router-interface ve 127
!
access-list 101 permit ip any any
!
interface ve 127
interface ve 127
ip address 30.22.45.177 255.255.255.0
ip address 17.25.254.13 255.255.255.0
ip address 17.25.254.14 255.255.255.0
ipv6 address 3024::1234/64 anycast
ipv6 address 3211:2345:3456::5555/54
ipv6 address fe80::1113/46 link-local anycast
ip allow-promiscuous-vip
access-list 101 in
icmp-rate-limit 98 lockup 99 100
name "test ve 127"
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxxxx&method=network.ve.get&port_num=127

Response as the HTTP body:

```
<ve>
<port_num>127</port_num>
<name>test ve 127</name>
<status>1</status>
<normal_rate>98</normal_rate>
<lookup_rate>99</lookup_rate>
<lookup_period>100</lookup_period>
<ipv4_addr_list>
  <ipv4>
    <ipv4_addr>30.22.45.177</ipv4_addr>
    <ipv4_mask>255.255.255.0</ipv4_mask>
  </ipv4>
  <ipv4>
    <ipv4_addr>17.25.254.13</ipv4_addr>
    <ipv4_mask>255.255.255.0</ipv4_mask>
  </ipv4>
  <ipv4>
    <ipv4_addr>17.25.254.14</ipv4_addr>
    <ipv4_mask>255.255.255.0</ipv4_mask>
  </ipv4>
</ipv4_addr_list>
<ipv4_acl>101</ipv4_acl>
<ipv6_addr_list>
</ipv6>
```

```

<ipv6_addr>3024::1234</ipv6_addr>
<ipv6_prefix_len>64</ipv6_prefix_len>
<ipv6_is_any_cast_addr>1</ipv6_is_any_cast_addr>
</ipv6>
<ipv6>
<ipv6_addr>3211:2345:3456::5555</ipv6_addr>
<ipv6_prefix_len>54</ipv6_prefix_len>
<ipv6_is_any_cast_addr>0</ipv6_is_any_cast_addr>
</ipv6>
</ipv6_addr_list>
<ipv6_auto_link_local>0</ipv6_auto_link_local>
<ipv6_link_local_cfg>
<ipv6_link_local_addr>fe80::1113</ipv6_link_local_addr>
<ipv6_link_local_prefix_len>46</ipv6_link_local_prefix_len>
<ipv6_link_local_is_any_cast>1</ipv6_link_local_is_any_cast>
</ipv6_link_local_cfg>
<allow_promiscuous_vip>1</allow_promiscuous_vip>
<tcp_sync_cookie>0</tcp_sync_cookie>
</ve>

```

Menus Privilege

Config Mode >> Network >> Interface >> Virtual

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

6.2.3 “network.ve.set” Method

This method is used to configure settings for the AX device’s Virtual Ethernet (VE) data ports.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	network.ve.set	String		
name	Interface name	String		“”
port_num (*)	Interface number.	Int		
status	Interface status, disabled(0) or enabled(1).	Int	0 or 1	0
normal_rate	Interface icmp normal rate, if not set, icmp rate limit will be disabled	Int	1 – 65535	
lockup_rate	Interface icmp lockup rate	Int	1 – 65535	
lockup_period	Interface lockup period	Int	1 - 16383	
ipv4_addr	IPv4 address (main address) of this interface	String		“”
ipv4_mask	IPv4 mask	String		“”
ipv6_acl	IPv6 access list name	String	Predefined	
ipv4_acl	IPv4 access list	Int	1-199, predefined	
ipv6_auto_link_local	IPv6 automatic link local status	Int	0 or 1	0
ipv6_link_local_addr	IPv6 link local address	String		
ipv6_link_local_prefix_len	IPv6 link local prefix length	Int	1-128	
ipv6_link_local_is_any_cast	Whether the link local address is a anycast address, yes(1), no(0)	Int	0 or 1	0
allow_promiscuous_vip	Allow promiscuous vip, disabled(0) or enabled(1)	Int	0 or 1	0
tcp_syn_cookie	Tcp syn cookie, disabled(0) or enabled(1)	Int	0 or 1	0

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
vlan 129
tagged ethernet 6
router-interface ve 129
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxx&method=network.ve.set&port_num=129&status=1&ipv4_addr=192.168.3.2&ipv4_mask=255.255.255.0

CLI configuration after aXAPI call:

```
!
vlan 129
tagged ethernet 6
router-interface ve 129
!
interface ve 129
ip address 192.168.3.2 255.255.255.0
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Network >> Interface >> Virtual

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

6.2.4 “network.ve.ipv4.add” Method

This method is used to configure IPv4 interface settings for the AX device’s Virtual Ethernet (VE) data ports.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	network.ve.ipv4.add	String		
port_num(*)	The interface number	Int		
ipv4_addr(*)	IPv4 address	String		“”
ipv4_mask(*)	IPv4 mask	String		“”

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
vlan 2
tagged ethernet 2
router-interface ve 2
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxx&method=network.ve.ipv4.add&port_num=2&ipv4_addr=124.254.6.6&ipv4_mask=255.255.0.0

CLI configuration after aXAPI call:

```
!
vlan 2
tagged ethernet 2
router-interface ve 2
!
interface ve 2
 ip address 124.254.6.6 255.255.0.0
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Network >> Interface >> Virtual

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

6.2.5 “network.ve.ipv4.delete” Method

This method is used to delete an IPv4 Virtual Ethernet (VE) data port.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	network.ve.ipv4.delete	String		
port_num(*)	The interface number	Int		
ipv4_addr(*)	IPv4 address	String		“”

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
vlan 2
tagged ethernet 2
router-interface ve 2
!
interface ve 2
ip address 124.254.6.6 255.255.0.0
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=network.ve.ipv4.delete&port_num=2&ipv4_addr=124.25
4.6.6
```

CLI configuration after aXAPI call:

```
!
vlan 2
tagged ethernet 2
router-interface ve 2
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Network >> Interface >> Virtual

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

6.2.6 “network.ve.ipv6.add” Method

This method is used to add an IPv6 Virtual Ethernet (VE) data port.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	network.ineface.ipv6.add	String		
port_num(*)	The interface number	Int		“”
ipv6_addr (*)	IPv6 address	String		“”
ipv6_prefix_len(*)	IPv6 prefix length	Int		“”
ipv6_is_any_cast_addr	Whether it is a IPv6 any cast address	Int	0 or 1	0

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:

- https://_AX_IP_Address_/services/rest/V2/
- [HTTP Action:](#)
POST

CLI configuration before aXAPI call:

```
!
vlan 2
tagged ethernet 2
router-interface ve 2
!
```

[HTTP Body](#)

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=network.ve.ipv6.add&port_num=2&ipv6_addr=4ff6::5678&ipv6_prefix_len=64&ipv6_is_any_cast_addr=1
```

CLI configuration after aXAPI call:

```
!
vlan 2
tagged ethernet 2
router-interface ve 2
!
interface ve 2
  ipv6 address 4ff6::5678/64  anycast
!
```

[Response as the HTTP body:](#)

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Network >> Interface >> Virtual

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

6.2.7 “network.ve.ipv6.delete” Method

This method is used to delete an IPv6 Virtual Ethernet (VE) data port.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	network.ve.ipv6.delete	String		
port_num(*)	The interface number	Int		
ipv6_addr (*)	IPv6 address	String		“”

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

[Request as:](#)

- [URL:](#)
https://_AX_IP_Address_/services/rest/V2/

- **HTTP Action:**
POST

CLI configuration before aXAPI call:

```
!
vlan 2
tagged ethernet 2
router-interface ve 2
!
interface ve 2
 ipv6 address 4ff6::5678/64 anycast
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxxxx&method=network.ve.ipv6.delete&port_num=2&ipv6_addr=4ff6::5678
```

CLI configuration after aXAPI call:

```
!
vlan 2
tagged ethernet 2
router-interface ve 2
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Network >> Interface >> Virtual

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

6.3 Management Interface

6.3.1 “network.mgmt_interface.get” Method

This method is used to get information about the status of an AX management interface.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	network.mgmt_interface.get	String		

Note: (*) parameter is required in the API.

Response Fields

mgmt_if	XML tag for management interface configuration info
status	management interface status, disabled(0) or enabled(1)
mac_addr	management interface MAC address
speed	auto, 10M, 100M or 1G
duplexity	auto, half or full

flow_ctl	flow control, disabled (0) or enabled (1)
apps_use_mgmt_port	control application use management port enabled (1), disabled (0)
ipv4_addr	management interface IPv4 address
ipv4_mask	management interface IPv4 mask
ipv4_gateway	management interface IPv4 default gateway
ipv6_address	management interface IPv6 address
ipv6_prefix_len	management interface IPv6 prefix length
ipv6_gateway	management interface IPv6 default gateway

Example Response

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
interface management
 ip address 192.168.100.44 255.255.255.0
 ip control-apps-use-mgmt-port
 ipv6 address 3ff7::1234/64
 ipv6 default-gateway 3ff1::1
 flow-control
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxx&method=network.mgmt_interface.get

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
  <mgmt_interface>
    <status>1</status>
    <mac_addr>0090.0B0C.885A</mac_addr>
    <speed>auto</speed>
    <duplexity>auto</duplexity>
    <flow_ctl>1</flow_ctl>
    <apps_use_mgmt_port>1</apps_use_mgmt_port>
    <ipv4_addr>192.168.100.44</ipv4_addr>
    <ipv4_mask>255.255.255.0</ipv4_mask>
    <ipv4_gateway />
    <ipv6_addr>3ff7::1234</ipv6_addr>
    <ipv6_prefix_len>64</ipv6_prefix_len>
    <ipv6_gateway>3ff1::1</ipv6_gateway>
  </mgmt_interface>
</response>
```

Menus Privilege

Config Mode >> Network >> Interface >> Management

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

6.3.2 “network.mgmt_interface.set” Method

This method is used to configure parameters (such as IP address, speed, and status) of an AX management interface.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	network.mgmt_interface.set	String		
status	Management Interface status, disabled (0) or enabled (1).	Int	0 or 1	0
speed	Interface speed, auto, 10M, 100M or 1G	String	'auto', '10M', '100M', '1G', '10G'	'auto'
duplexity	management Interface duplex, auto, half or full	String	'auto', 'half', 'full'	'auto'
flow_ctrl	management Interface flow control, disabled(0) or enabled(1)	Int	0 or 1	0
apps_use_mgmt_port	apps use mgmt port, disabled(0) or enabled(1)	Int	0 or 1	0
ipv4_addr	management interface IPv4 ip address	String	"	"
ipv4_mask	management interface IPv4 net mask	String	"	"
ipv4_gateway	management interface IPv4 default gateway	String	"	"
ipv6_addr	management interface IPv6 ip address	String	"	"
ipv6_prefix_len	management interface IPv6 prefix length	Int	1 - 128	
ipv6_gateway	management interface IPv6 default gateway	String	"	"

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
interface management
ip address 192.168.100.193 255.255.255.0
ip default-gateway 192.168.100.1
flow-control
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxx&method=network.mgmt_interface.set&status=1&speed=100M&duplexity=full

CLI configuration after aXAPI call:

```
!
interface management
ip address 192.168.100.193 255.255.255.0
ip default-gateway 192.168.100.1
speed 100
duplexity full
flow-control
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Network >> Interface >> Management

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

6.4 Gateway Configuration

6.4.1 “network.transparent_interface.get” Method

This method is used to get information (such as IP4v4/IPv6 address, mask, gateway) for an AX transparent interface.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	network.transparent_interface.get	String		

Note: (*) parameter is required in the API.

Response Fields

transparent_interface	XML tag for transparent interface configuration info
Ipv4_addr	transparent interface IPv4 address
Ipv4_mask	transparent interface IPv4 mask
Ipv4_gateway	transparent interface IPv4 default gateway
Ipv6_addr	transparent interface IPv6 address
Ipv6_prefix_len	transparent interface IPv6 prefix length
Ipv6_gateway	transparent interface IPv6 default gateway

Example Response

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
ip address 124.254.1.1 255.255.255.0
ip default-gateway 124.254.1.2
!
ipv6 address 3ff1::2/64
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxx&method=network.transparent_interface.get

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
<transparent_interface>
    <Ipv4_addr>124.254.1.1</Ipv4_addr>
    <Ipv4_mask>255.255.255.0</Ipv4_mask>
    <Ipv4_gateway>124.254.1.2</Ipv4_gateway>
    <Ipv6_addr>3ff1::2</Ipv6_addr>
    <Ipv6_prefix_len>64</Ipv6_prefix_len>
```

```
</transparent_interface>
</response>
```

Menus Privilege

Config Mode >> Network >> Interface >> Transparent

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

6.4.2 “network.transparent_interface.set” Method

This method is used to configure parameters (such as IP4v4/IPv6 address, mask, gateway) for an AX transparent interface.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	network.transparent-interface.set	String		
ipv4_addr	transparent interface IPv4 ip address	String		
ipv4_mask	transparent interface IPv4 net mask	String		
ipv4_gateway	transparent interface IPv4 default gateway	String		
ipv6_addr	transparent interface IPv6 ip address	String		
ipv6_prefix_len	transparent interface IPv6 prefix length	Int	1 - 128	
ipv6_gateway	transparent interface IPv6 default gateway	String		

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://AX_IP_Address/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
hostname AX1000
!
clock timezone America/Los_Angeles
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=network.transparent_interface.set&ipv4_addr=124.25
4.1.1&ipv4_mask=255.255.0.0&ipv4_gateway=124.254.1.3&ipv6_addr=3001::1&ipv6_prefix_len=48
&ipv6_gateway=3001::2
```

CLI configuration after aXAPI call:

```
!
hostname AX1000
!
clock timezone America/Los_Angeles
!
ip address 124.254.1.1 255.255.0.0
ip default-gateway 124.254.1.3
!
ipv6 address 3001::1/48
```

```
ipv6 default-gateway 3001::2
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Network >> Interface >> Transparent

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

6.5 Trunk

6.5.1 “network.trunk.getAll” Method

This method is used to get configuration information (such as trunk ID, status, threshold) for all trunks, where a trunk is defined as a set of Ethernet data ports configure as a single logical link.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	network.trunk.getAll	String		

Note: (*) parameter is required in the API.

Response Fields

trunk_list	XML tag for the collection of trunks
trunk	XML tag for trunk trunk
id	trunk id
name	trunk name
member_count	count of members
type	0: Static, 1: Dynamic (LACP)
status	0:down, 1:up
threshold	trunk ports threshold
working_lead	Working Lead
threshold_timer	trunk ports threshold timer, (second)
threshold_running	Running? 1:yes, 0:No
interface_list	XML tag for the collection of trunk interfaces
interface	XML tag for trunk interface
int_num:	trunk interface number
cfg_status:	trunk interface config status, disabled(0) or enabled(1)
oper_status	trunk interface operation status, disabled(0) or enabled(1)

Example Response

Request as:

- URL:

- [https://\[AX_IP_ADDRESS\]/services/rest/V2/](https://[AX_IP_ADDRESS]/services/rest/V2/)
- **HTTP Action:**
GET

CLI configuration before aXAPI call:

```
!
trunk 1
 ethernet 1
 disable ethernet 1
!
trunk 2
 ethernet 2
 disable ethernet 2
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxxxx&method=network.trunk.getAll

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
    <trunk_list>
        <trunk>
            <id>1</id>
            <name>trunk01</name>
            <member_count>1</member_count>
            <type>0</type>
            <status>0</status>
            <threshold>0</threshold>
            <working_lead>0</working_lead>
            <interface_list>
                <interface>
                    <if_num>1</if_num>
                    <cfg_status>0</cfg_status>
                    <oper_status>0</oper_status>
                </interface>
            </interface_list>
        </trunk>
        <trunk>
            <id>2</id>
            <name>trunk02</name>
            <member_count>1</member_count>
            <type>0</type>
            <status>0</status>
            <threshold>0</threshold>
            <working_lead>0</working_lead>
            <interface_list>
                <interface>
                    <if_num>2</if_num>
                    <cfg_status>0</cfg_status>
                    <oper_status>0</oper_status>
                </interface>
            </interface_list>
        </trunk>
    </trunk_list>
</response>
```

Menus Privilege

Config Mode >> Network >> Trunk

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

6.5.2 “network.trunk.search” Method

This method is used to get trunk configuration details using trunk ID.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	network.trunk.search	String		
id (*)	trunk id	Int		

Note: (*) parameter is required in the API.

Response Fields

trunk	XML tag for trunk
id	trunk id
name	trunk name
member_count	count of members
type	0: Static, 1: Dynamic (LACP)
status	0:down, 1:up
threshold	trunk ports threshold
working_lead	Working Lead
threshold_timer	trunk ports threshold timer, (second)
threshold_running	Running? 1:yes, 0:No
interface_list	XML tag for the collection of trunk interfaces
interface	XML tag for trunk interface
int_num:	trunk interface number
cfg_status:	trunk interface config status, disabled(0) or enabled(1)
oper_status	trunk interface operation status, disabled(0) or enabled(1)

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
trunk 1
 ethernet 2
 disable ethernet 2
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxx&method=network.trunk.search&id=1

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
    <trunk>
        <id>1</id>
        <name>trunk01</name>
        <member_count>1</member_count>
        <type>0</type>
        <status>0</status>
        <threshold>0</threshold>
        <working_lead>0</working_lead>
        <interface_list>
            <interface>
                <if_num>2</if_num>
                <cfg_status>0</cfg_status>
            
```

```

<oper_status>0</oper_status>
</interface>
</interface_list>
</trunk>
</response>

```

Menus Privilege

Config Mode >> Network >> Trunk

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

6.5.3 “network.trunk.create” Method

This method is used to create a trunk (set trunk ID, status, port threshold).

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	network.trunk.create	String		
id (*)	trunk id	Int		
name	Trunk name	String	1 - 63	
Interface_list (*)	trunk interface list to be added	String		
int<n>	interface at element <n>			
int_num (*)	trunk interface number	Int		
status (*)	trunk interface status, disabled(0) or enabled(1)	Int	0 or 1	
threshold	trunk ports threshold	Int		
threshold_timer	trunk ports threshold timer (second)	Int		

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=network.trunk.create&id=1&threshold=2&threshold_timer=50&interface_list=int1%02int2&int1=int_num%031%02status%030&int2=int_num%032%02status%031&name=trunk01
```

CLI configuration after aXAPI call:

```
!
trunk 1
  ethernet 1 to 2
  disable ethernet 1
  ports-threshold 2
  ports-threshold-timer 50
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Network >> Trunk

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

6.5.4 “network.trunk.update” Method

This method is used to update information (trunk ID, port threshold, port threshold timer) for a trunk. If the interface does not exist, this interface will be added.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	network.trunk.update	String		
id (*)	trunk id	Int		
name	Trunk name	String	1 - 63	
Interface_list (*)	trunk interface list to be added	String		
int<n>	interface at element <n>	Int		
int_num (*)	trunk interface number	Int		
status (*)	trunk interface status, disabled(0) or enabled(1)	Int	0 or 1	
threshold	trunk ports threshold	Int		
threshold_timer	trunk ports threshold timer (second)	Int		

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
trunk 1
 ethernet 1 to 2
 disable ethernet 1
 ports-threshold 2
 ports-threshold-timer 50
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=network.trunk.update&id=1&threshold=2&threshold_timer=50&interface_list=int1%02int2%02int3&int1=int_num%031%02status%030&int2=int_num%032%02status%031&int3=int_num%033%02status%030&name=trunk02
```

CLI configuration after aXAPI call:

```
!
```

```

trunk 1
 ethernet 1 to 3
 disable ethernet 1 ethernet 3
 ports-threshold 2
 ports-threshold-timer 50
!

```

Response as the HTTP body:

```

<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>

```

Menus Privilege

Config Mode >> Network >> Trunk

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

6.5.5 “network.trunk.delete” Method

This method is used to delete a trunk.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	network.trunk.delete	String		
id (*)	trunk id	Int		

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```

!
trunk 1
 ethernet 1
 disable ethernet 1
!
trunk 2
 ethernet 2
 disable ethernet 2
!

```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxx&method=network.trunk.delete&id=1

CLI configuration after aXAPI call:

```

!
trunk 2

```

```
! ethernet 2
! disable ethernet 2
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Network >> Trunk

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

6.6 VLAN

6.6.1 “network.vlan.getAll” Method

This method is used to get configuration information, such as ID, name, interface list, for all VLANs configured on the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	network.vlan.getAll	String		

Note: (*) parameter is required in the API.

Response Fields

vlan_list	XML tag for the collection of vlans
vlan	XML tag for vlan
id	vlan id
name	vlan name
virtual_interface	virtual interface number
interface_list	XML tag for the collection of vlan interfaces
interface	XML tag for vlan interface
int_num:	vlan interface number
tagged:	vlan interface status, untagged(0) or tagged(1)

Example Response

Request as:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
- **HTTP Action:**
GET

CLI configuration before aXAPI call:

```
!
vlan 100
tagged ethernet 2
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxxxx&method=network.vlan.getAll

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
  <vlan_list>
    <vlan>
      <id>1</id>
      <interface_list>
        <interface>
          <if_num>1</if_num>
          <tagged>0</tagged>
        </interface>
        <interface>
          <if_num>3</if_num>
          <tagged>0</tagged>
        </interface>
      </interface_list>
    </vlan>
    <vlan>
      <id>100</id>
      <interface_list>
        <interface>
          <if_num>2</if_num>
          <tagged>1</tagged>
        </interface>
      </interface_list>
    </vlan>
  </vlan_list>
</response>
```

Menus Privilege

Config Mode >> Network >> VLAN >> VLAN

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

6.6.2 “network.vlan.search” Method

This method is used to get vlan details for a specific vlan ID.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	network.vlan.search	String		
id (*)	vlan id	Int		

Note: (*) parameter is required in the API.

Response Fields

vlan	XML tag for vlan
id	vlan id
name	vlan name
virtual_interface	virtual interface number
interface_list	XML tag for the collection of vlan interfaces
interface	XML tag for vlan interface
int_num:	vlan interface number

tagged: vlan interface status, untagged(0) or tagged(1)

Example Response

- Request as:
- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
 - HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
vlan 2
  tagged ethernet 2 ethernet 4
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=network.vlan.search&id=2
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
  <vlan>
    <id>2</id>
    <interface_list>
      <interface>
        <if_num>2</if_num>
        <tagged>1</tagged>
      </interface>
      <interface>
        <if_num>4</if_num>
        <tagged>1</tagged>
      </interface>
    </interface_list>
  </vlan>
</response>
```

Menus Privilege

Config Mode >> Network >> VLAN >> VLAN

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

6.6.3 “network.vlan.create” Method

This method is used to create a VLAN on the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	network.vlan.create	String		
id (*)	vlan id	Int		
name	vlan name	String		
virtual_interface	vlan virtual interface	Int		
interface_list (*)	vlan interface list to be added	String		
int<n>	interface at element <n>	Int		
int_num (*)	vlan interface number	Int		
tagged(*)	vlan interface status, untagged(0) or tagged(1)	Int	0 or 1	

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=network.vlan.create&vlan=id%032%02name%03name2%02
interface_list&interface_list=int1%02int2&int1=int_num%031%02tagged%030&int2=int_num%032
%02tagged%031
```

CLI configuration after aXAPI call:

```
!
vlan 2
  untagged ethernet 1
  tagged ethernet 2
  name "name2"
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Network >> VLAN >> VLAN

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

6.6.4 “network.vlan.update” Method

This method is used to update a VLAN. If the interface by given interface number (statis MAC) does not exist, then the interface will be added. If the interface number exists, then this VLAN will be updated using the provided values.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	network.vlan.update	String		
id (*)	vlan id	Int		
name	vlan name	String		
virtual_interface	vlan virtual interface	Int		
interface_list (*)	vlan interface list to be added	String		
int<n>	interface at element <n>			
int_num (*)	vlan interface number	Int		
tagged (*)	vlan interface status, untagged(0) or tagged(1)	Int	0 or 1	

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
vlan 2
  untagged ethernet 1
  tagged ethernet 2
    name "name2"
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=network.vlan.update&vlan=id%032%02name%03name2%02
interface_list&interface_list=int1%02int2%02int3&int1=int_num%031%02tagged%030&int2=int_
num%032%02tagged%031&int3=int_num%033%02tagged%031
```

CLI configuration after aXAPI call:

```
!
vlan 2
  untagged ethernet 1
  tagged ethernet 2 to 3
    name "name2"
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Network >> VLAN >> VLAN

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

6.6.5 “network.vlan.delete” Method

This method is used to delete a vlan by entering a specific vlan ID.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	network.vlan.delete	String		
id (*)	vlan id	Int		

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
vlan 2
  untagged ethernet 1
  tagged ethernet 2 to 3
  name "name2"
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxxxx&method=network.vlan.delete&id=2

CLI configuration after aXAPI call:

```
!
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Network >> VLAN >> VLAN

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

6.7 Route

6.7.1 “network.route.ipv4static.getAll” Method

This method is used to get configuration information for all IPv4 static routes, such as address, gateway, and distance.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	network.route.ipv4static.getAll	String		

Note: (*) parameter is required in the API.

Response Fields

route_list	XML tag for the collection of static route IPv4
route	XML tag for static route IPv4

address	route IPv4 address
mask	route netmask
gateway	route gateway
distance	route distance

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
ip route 192.168.3.1 /32 10.2.0.3 10
ip route 192.168.3.2 /32 10.2.0.4 5
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxxxx&method=network.route.ipv4static.getAll

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
  <route_list>
    <route>
      <address>192.168.3.1</address>
      <mask>255.255.255.255</mask>
      <gateway>10.2.0.3</gateway>
      <distance>10</distance>
    </route>
    <route>
      <address>192.168.3.2</address>
      <mask>255.255.255.255</mask>
      <gateway>10.2.0.4</gateway>
      <distance>5</distance>
    </route>
  </route_list>
</response>
```

Menus Privilege

Config Mode >> Network >> Route >> IPv4 Static

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

6.7.2 “network.route.ipv4static.create” Method

This method is used to get create an IPv4 static route.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	network.route.ipv4static.create	String		
address (*)	route ipv4 address	String		
mask (*)	route netmask	String		
gateway (*)	route gateway	String		
distance	route distance	int	1 - 255	

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
ip route 192.168.3.2 /32 10.2.0.4 5
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxxxx&method=network.route.ipv4static.create&address=192.168.3.1&mask=255.255.255.255&gateway=10.2.0.3&distance=5
```

CLI configuration after aXAPI call:

```
!
ip route 192.168.3.1 /32 10.2.0.3 5
ip route 192.168.3.2 /32 10.2.0.4 5
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Network >> Route >> IPv4 Static

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

6.7.3 “network.route.ipv4static.update” Method

This method is used to update one or more parameters for an IPv4 static route.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	network.route.ipv4static.update	String		
address (*)	route ipv4 address	String		
mask	route netmask	String		
gateway	route gateway	String		
distance	route distance	int	1 - 255	

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
ip route 192.168.3.1 /32 10.2.0.3 5
ip route 192.168.3.2 /32 10.2.0.4 5
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxxxx&method=network.route.ipv4static.update&address=192.168.3.1&mask=255.255.255.255&gateway=10.2.0.3&distance=20
```

CLI configuration after aXAPI call:

```
!
ip route 192.168.3.1 /32 10.2.0.3 20
ip route 192.168.3.2 /32 10.2.0.4 5
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Network >> Route >> IPv4 Static

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

6.7.4 “network.route.ipv4static.delete” Method

This method is used to get delete an IPv4 static route.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	network.route.ipv4static.delete	String		
address (*)	Route ipv4 address	String		
mask (*)	route netmask	String		
gateway (*)	route gateway	String		

Notes: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/

- **HTTP Action:**
POST

CLI configuration before aXAPI call:

```
!
ip route 192.168.3.1 /32 10.2.0.3 20
ip route 192.168.3.2 /32 10.2.0.4 5
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxxxx&method=network.route.ipv4static.delete&address=192.168.3.1&mask=255.255.255.255&gateway=10.2.0.3
```

CLI configuration after aXAPI call:

```
!
ip route 192.168.3.2 /32 10.2.0.4 5
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Network >> Route >> IPv4 Static

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

6.7.5 “network.route.ipv6static.getAll” Method

This method is used to get all configuration information, such as IPv6 address, gateway, and route distance, for all IPv6 static routes.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	network.route.ipv6static.getAll	String		

Note: (*) parameter is required in the API.

Response Fields

route_list	XML tag for the collection of static route IPv6
route	XML tag for static route IPv6
address	route IPv6 address
prefix_len	route IPv6 prefix length
gateway	forwarding router address
out_going_port	
type	outgoing interface type
port_num	outgoing interface number
distance	route IPv6 distance

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
ipv6 route 3ff2::/64 3ff1::2 5
ipv6 route 3ff3::/64 3ff1::2 5
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxx&method=network.route.ipv6static.getAll

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
  <route_list>
    <route>
      <address>3ff2::</address>
      <prefix_len>64</prefix_len>
      <gateway>3ff1::2</gateway>
      <out_going_port>
        <type>unknown</type>
        <port_num>0</port_num>
      </out_going_port>
      <distance>5</distance>
    </route>
    <route>
      <address>3ff3::</address>
      <prefix_len>64</prefix_len>
      <gateway>3ff1::2</gateway>
      <out_going_port>
        <type>unknown</type>
        <port_num>0</port_num>
      </out_going_port>
      <distance>5</distance>
    </route>
  </route_list>
</response>
```

Menus Privilege

Config Mode >> Network >> Route >> IPv6 Static

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

6.7.6 “network.route.ipv6static.create” Method

This method is used to create an IPv6 static route.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	network.route.ipv6static.create	String		
address (*)	route IPv6 address	String		
prefix_len(*)	route IPv6 prefix length	Int	1 – 128	
gateway (*)	route forwarding router address	String		
type	route outgoing port type	String	one of ‘ethernet’, ‘ve’ or ‘trunk’	
port_num	route outgoing port type	Int		

distance	route distance	Int	1 - 255	
-----------------	----------------	-----	---------	--

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
vlan 3
tagged ethernet 3
router-interface ve 3
!
interface ve 3
 ipv6 address fe80::1111/64 link-local anycast
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxxx&method=network.route.ipv6static.create&address=7ff7::&prefix_len=64&gateway=fe80::5234&type=ve&port_num=3&distance=5
```

CLI configuration after aXAPI call:

```
!
vlan 3
tagged ethernet 3
router-interface ve 3
!
interface ve 3
 ipv6 address fe80::1111/64 link-local anycast
!
ipv6 route 7ff7::/64 fe80::5234 ve 3 5
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Network >> Route >> IPv6 Static

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

6.7.7 “network.route.ipv6static.update” Method

This method is used to update one or more parameters for an IPv6 static route.

Parameters

Parameter Name	Description	Data Type	Range	Default
----------------	-------------	-----------	-------	---------

session_id (*)	user authenticated session id.	String		
method (*)	network.route.ipv6static.update	String		
address (*)	route IPv6 address	String		
prefix_len	route IPv6 prefix length	Int	1 – 128	
gateway (*)	route forwarding router address	String		
gateway_new	new gateway to replace the old one	String		
type	Route outgoing port type	String	one of 'ethernet', 've' or 'trunk'	
port_num	Route outgoing port type	Int		
distance	route distance	Int	1 - 255	

Note: (*) parameter is required in the API.

Address is the index.

Response Fields

General XML status response

Example Response

Request as:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
- **HTTP Action:**
POST

CLI configuration before aXAPI call:

```
!
ipv6 route 7ff7::/64 fe80::5234 ve 3 5
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=network.route.ipv6static.update&address=7ff7::&prefix_len=32&gateway=fe80::5236&gateway_new=fe80::5238&type=ve&port_num=3&distance=5
```

CLI configuration after aXAPI call:

```
!
ipv6 route 7ff7::/32 fe80::5238 ve 3 5
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Network >> Route >> IPv6 Static

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

6.7.8 “network.route.ipv6static.delete” Method

This method is used to delete the parameters for an IPv6 static route.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		

method (*)	network.route.ipv6static.delete	String		
address (*)	route IPv6 address	String		
gateway (*)	route forwarding router address	String		

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
ipv6 route 7ff7::/32 2011::1112 5
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxxxx&method=method=network.route.ipv6static.delete&address=7ff
7::&gateway=2011::1112

CLI configuration after aXAPI call:

```
!
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Network >> Route >> IPv6 Static

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

6.8 ARP

6.8.1 “network.arp.getAll” Method

This method is used to get configuration information for all IPv4 ARPs. The output is the collection of all network.arp.get methods for each IPv4 ARP.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	network.arp.getAll	String		

Note: (*) parameter is required in the API.

Response Fields

arp_list	XML tag for the collection of IPv4 ARPs
arp	XML tag for IPv4 ARP
address	ARP IPv4 address
mac	ARP MAC address
type	ARP type. Static or Dynamic
age	ARP age
interface	ARP interface name
vlan_id	ARP VLAN id

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
```

```
!
```

Notes: In this example, no static ARP is configured. So only dynamic ARP is returned.

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=network.arp.getAll
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
    <arp_list>
        <arp>
            <address>192.168.100.1</address>
            <mac>001f.a010.3a41</mac>
            <type>Dynamic</type>
            <age>208</age>
            <interface>Management</interface>
        </arp>
    </arp_list>
</response>
```

Menus Privilege

Config Mode >> Network >> ARP >> IPv4

Config Mode >> Network >> ARP >> IPv6 Neighbor

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

6.8.2 “network.arp.search” Method

This method is used to get ARP configuration information using an arp IP address.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	network.arp.search	String		
address	arp ip address	String	IPv4 or IPv6	

ip_version	4: get ipv4 list	6:get ipv6 list	int	4 or 6	
-------------------	------------------	-----------------	-----	--------	--

Note: (*) parameter is required in the API.

Response Fields

arp	XML tag for IPv4 ARP
address	ARP IPv4 address
mac	ARP MAC address
type	ARP type. Static or Dynamic
age	ARP age
interface	ARP interface name
vlan_id	ARP VLAN id

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
!
```

Notes: In this example, no static ARP is configured. So only dynamic ARP is returned.

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=network.arp.search&address=192.168.100.1&ip_version=4
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
  <arp_list>
    <arp>
      <address>192.168.100.1</address>
      <mac>001f.a010.3a41</mac>
      <type>Dynamic</type>
      <age>208</age>
      <interface>Management</interface>
    </arp>
  </arp_list>
</response>
```

Menus Privilege

Config Mode >> Network >> ARP >> IPv4

Config Mode >> Network >> ARP >> IPv6 Neighbor

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

6.8.3 “network.arp.create” Method

This method is used to add an ARP mapping for a given IPv4 or IPv6 address.

Parameters

Parameter Name	Description	Data Type	Range	Default
----------------	-------------	-----------	-------	---------

session_id (*)	user authenticated session id.	String		
method (*)	network.arp.create	String		
address (*)	arp IPv4 or IPv6 address	String		
mac (*)	arp MAC address	String	xxxx.xxxx.xxxx	
int_num	arp interface number	Int		
vlan_id	arp vlan id	Int		

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
- **HTTP Action:**
POST

CLI configuration before aXAPI call:

```
!
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=network.arp.create&address=192.168.3.1&mac=aaaa:bb  
bb:cccc
```

CLI configuration after aXAPI call:

```
!
arp 192.168.3.1 aaaa.bbbb.cccc
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Network >> ARP >> IPv4

Config Mode >> Network >> ARP >> IPv6 Neighbor

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

6.8.4 “network.arp.update” Method

This method is used to update the parameters for an ARP entry.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	network.arp.update	String		
address (*)	arp ipv4 or IPv6 address	String		
mac	arp MAC address	String	xxxx.xxxx.xxxx	
int_num	arp interface number	Int		
vlan_id	arp vlan id	Int		

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
arp 192.168.3.1 aaaa.bbbb.cccc
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=network.arp.update&address=192.168.3.1&mac=aaaa.bb
bb.dddd&int_num=1
```

CLI configuration after aXAPI call:

```
!
arp 192.168.3.1 aaaa.bbbb.dddd
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Network >> ARP >> IPv4

Config Mode >> Network >> ARP >> IPv6 Neighbor

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

6.8.5 “network.arp.delete” Method

This method is used to get delete an ARP entry.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	network.arp.delete	String		
address (*)	arp ipv4 or IPv6 address	String		

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
- **HTTP Action:**
POST

CLI configuration before aXAPI call:

```
!
arp 192.168.3.1 aaaa.bbbb.dddd
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=network.arp.delete&address=192.168.3.1
```

CLI configuration after aXAPI call:

```
!
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Network >> ARP >> IPv4

Config Mode >> Network >> ARP >> IPv6 Neighbor

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

6.9 ACL

6.9.1 “network.acl.getAll” Method

This method is used to get configuration information for all Access Control Lists (ACLs). The output is the collection of all network.acl.get methods for each ACL.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	network.acl.getAll	String		

Note: (*) parameter is required in the API.

Response Fields

ACL_List:

XML tag of ACL list

total_num:

total number of ACLs

ACL:

XML tag of ACL

id:

ACL ID, only for ipv4 acl

name:

ACL name, only for IPv6 acl

remark:

ACL remark

ACL_Item:

(if IPv4 standard ACL, ACL id < 100) XML tag of ACL content

id:

ACL id, only for IPv4 ACL

seq_num:

ACL item sequence number

action: 0(drop), 1(permit), 2(l3-vlan-fwd-disable)
 src_ip: source IP address, 0.0.0.0 means any
 src_mask: source IP mask, 0 means apply, 255 means ignore
 log: 0(no log), 1(log), 2(only log transparent sessions)
 hits: hit number
ACL_Item: (if IPv4 externed ACL, ACL ID > 100) XML tag of ACL content, can be one or more
 id: ACL id, only for IPv4 ACL
 seq_num: ACL item sequence number
 action: 0(drop), 1(permit), 2(l3-vlan-fwd-disable)
 protocol: protocol, 0(IP), 1(ICMP), 6(TCP), 17(UDP)
 icmp_code: icmp code, only when protocol is ICMP
 icmp_type: icmp type, only when protocol is ICMP
 src_ip: source IP address, 0.0.0.0 means any
 src_mask: source IP mask, 0 means apply, 255 means ignore
 src_port_start: source port start, only when protocol is TCP or UDP
 src_port_end: source port end, only when protocol is TCP or UDP
 dst_ip: destination IP address
 dst_mask: destination IP mask
 dst_port_start: destination port start, only when protocol is TCP or UDP
 dst_port_end: destination port end, only when protocol is TCP or UDP
 log: 0(no log), 1(log), 2(only log transparent sessions)
 dscp: DSCP
 vlan: VLAN ID
 fragments: IP fragments
 tcp_est: TCP established, only when protocol is TCP or UDP
 hits:
ACL_Item: (if IPv6 ACL, replaced ACL ID by ACL name) XML tag of ACL content, can be one or more
 name: ACL name, only for IPv6 ACL
 seq_num: ACL item sequence number
 action: 0(drop), 1(permit), 2(l3-vlan-fwd-disable)
 protocol: protocol, 41(IPv6), 58(ICMP), 6(TCP), 17(UDP)
 src_ip: source IP address, 0.0.0.0 means any
 src_mask: source IP mask, 0 means apply, 255 means ignore
 src_port_start: source port start, only when protocol is TCP or UDP
 src_port_end: source port end, only when protocol is TCP or UDP
 dst_ip: destination IP address
 dst_mask: destination IP mask
 dst_port_start: destination port start, only when protocol is TCP or UDP
 dst_port_end: destination port end, only when protocol is TCP or UDP
 log: 0(no log), 1(log), 2(only log transparent sessions)
 dscp: DSCP
 vlan: VLAN ID
 fragments: IP fragments
 tcp_est: TCP established, only when protocol is TCP or UDP
 hits:

Example Response

Request as:
 • URL:

● [https://\[AX_IP_ADDRESS\]/services/rest/V2/](https://[AX_IP_ADDRESS]/services/rest/V2/)

● HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
access-list 1 13-vlan-fwd-disable host 1.1.1.1
access-list 100 permit icmp type 17 code 200 100.100.100.0 0.0.0.255 host
    200.200.200.200 fragments dscp 63 log
access-list 100 13-vlan-fwd-disable tcp 101.101.101.0 0.0.0.255 eq 500 host
    102.102.102.102 range 100 200 vlan 9 dscp 63 established log transparent-
        session-only
ipv6 access-list ipv6_acl_name1
    deny icmp host 2002::2 any fragments vlan 100 dscp 63 log transparent-session-only
    permit ipv6 any any fragments vlan 9 dscp 1 log transparent-session-only
    13-vlan-fwd-disable udp any any
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=network.acl.getAll
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
<ACL_List>
    <total_num>3</total_num>
    <ACL>
        <id>1</id>
        <remark/>
        <ACL_Item>
            <id>1</id>
            <seq_num>10</seq_num>
            <action>2</action>
            <src_ip>1.1.1.1</src_ip>
            <src_mask>255.255.255.255</src_mask>
            <fragments>0</fragments>
            <tcp_est>0</tcp_est>
            <log>2</log>
            <hits>0</hits>
        </ACL_Item>
    </ACL>
    <ACL>
        <id>100</id>
        <remark/>
        <ACL_Item>
            <id>100</id>
            <seq_num>10</seq_num>
            <action>1</action>
            <protocol>1</protocol>
            <icmp_type>17</icmp_type>
            <icmp_code>200</icmp_code>
            <src_ip>100.100.100.0</src_ip>
            <src_mask>0.0.0.255</src_mask>
            <src_port_start>17</src_port_start>
            <src_port_end>200</src_port_end>
            <dst_ip>200.200.200.200</dst_ip>
            <dst_mask>255.255.255.255</dst_mask>
            <fragments>1</fragments>
            <dscp>63</dscp>
            <tcp_est>0</tcp_est>
            <log>1</log>
            <hits>0</hits>
        </ACL_Item>
        <ACL_Item>
            <id>100</id>
            <seq_num>20</seq_num>
            <action>2</action>
            <protocol>6</protocol>
            <icmp_type>244</icmp_type>
        </ACL_Item>
    </ACL>
</ACL_List>
```

```

<icmp_code>244</icmp_code>
<src_ip>101.101.101.0</src_ip>
<src_mask>0.0.0.255</src_mask>
<src_port_start>500</src_port_start>
<src_port_end>500</src_port_end>
<dst_ip>102.102.102.102</dst_ip>
<dst_mask>255.255.255.255</dst_mask>
<dst_port_start>100</dst_port_start>
<dst_port_end>200</dst_port_end>
<fragments>0</fragments>
<vlan>9</vlan>
<dscp>63</dscp>
<tcp_est>1</tcp_est>
<log>2</log>
<hits>0</hits>
</ACL_Item>
</ACL>
<ACL>
<name>ipv6_acl_name1</name>
<remark/>
<ACL_Item>
<name>ipv6_acl_name1</name>
<seq_num>10</seq_num>
<action>0</action>
<protocol>58</protocol>
<src_ip>2002::2</src_ip>
<src_mask>128</src_mask>
<dst_ip>::</dst_ip>
<dst_mask>0</dst_mask>
<fragments>1</fragments>
<vlan>100</vlan>
<dscp>63</dscp>
<tcp_est>0</tcp_est>
<log>2</log>
<hits>0</hits>
</ACL_Item>
<ACL_Item>
<name>ipv6_acl_name1</name>
<seq_num>20</seq_num>
<action>1</action>
<protocol>41</protocol>
<src_ip>::</src_ip>
<src_mask>0</src_mask>
<dst_ip>::</dst_ip>
<dst_mask>0</dst_mask>
<fragments>1</fragments>
<vlan>9</vlan>
<dscp>1</dscp>
<tcp_est>0</tcp_est>
<log>2</log>
<hits>0</hits>
</ACL_Item>
<ACL_Item>
<name>ipv6_acl_name1</name>
<seq_num>10</seq_num>
<action>2</action>
<protocol>6</protocol>
<src_ip>::</src_ip>
<src_mask>0</src_mask>
<dst_ip>::</dst_ip>
<dst_mask>0</dst_mask>
<fragments>0</fragments>
<tcp_est>0</tcp_est>
<log>0</log>
<hits>0</hits>
</ACL_Item>
</ACL>
</ACL_List>
</response>

```

Menus Privilege

Config Mode >> Network >> ACL >> Standard
 Config Mode >> Network >> ACL >> Extended
 Config Mode >> Network >> ACL >> IPv6

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

6.9.2 “network.acl.search” Method

This method is used to get ACL configuration information by ACL ID or name.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	network.acl.search	String		
id(*)	ACL ID, only for IPv4 ACL,including standard ACL and externed ACL (IPv4 ACL)	Int	1 – 199	
name(*)	ACL name, only for IPv6 ACL	String		

Note: (*) parameter is required in the API.

Response Fields

ACL_List:	XML tag of acl list
total_num:	total number of ACLs
ACL:	XML tag of ACL
id:	ACL ID, only for ipv4 acl
name:	ACL name, only for IPv6 acl
remark:	ACL remark
ACL_Item:	(if IPv4 standard ACL, ACL id < 100) XML tag of ACL content
id:	ACL id, only for IPv4 ACL
seq_num:	ACL item sequence number
action:	0(drop), 1(permit), 2(l3-vlan-fwd-disable)
src_ip:	source IP address, 0.0.0.0 means any
src_mask:	source IP mask, 0 means apply, 255 means ignore
log:	0(no log), 1(log), 2(only log transparent sessions)
hits:	hit number
ACL_Item:	(if IPv4 externed ACL, ACL ID > 100) XML tag of ACL content, can be one or more
id:	ACL id, only for IPv4 ACL
seq_num:	ACL item sequence number
action:	0(drop), 1(permit), 2(l3-vlan-fwd-disable)
protocol:	protocol, 0(IP), 1(ICMP), 6(TCP), 17(UDP)
icmp_code:	icmp code, only when protocol is ICMP
icmp_type:	icmp type, only when protocol is ICMP
src_ip:	source IP address, 0.0.0.0 means any
src_mask:	source IP mask, 0 means apply, 255 means ignore
src_port_start:	source port start, only when protocol is TCP or UDP
src_port_end:	source port end, only when protocol is TCP or UDP
dst_ip:	destination IP address
dst_mask:	destination IP mask
dst_port_start:	destination port start, only when protocol is TCP or UDP

dst_port_end:	destination port end, only when protocol is TCP or UDP
log:	0(no log), 1(log), 2(only log transparent sessions)
dscp:	DSCP
vlan:	VLAN ID
fragments:	IP fragments
tcp_est:	TCP established, only when protocol is TCP or UDP
hits:	hit number
ACL_Item:	(if IPv6 ACL, replaced ACL ID by ACL name) XML tag of ACL
content, can be one or more	
name:	ACL name, only for IPv6 ACL
seq_num:	ACL item sequence number
action:	0(drop), 1(permit), 2(13-vlan-fwd-disable)
protocol:	protocol, 41(IPv6), 58(ICMP), 6(TCP), 17(UDP)
src_ip:	source IP address, 0.0.0.0 means any
src_mask:	source IP mask, 0 means apply, 255 means ignore
src_port_start:	source port start, only when protocol is TCP or UDP
src_port_end:	source port end, only when protocol is TCP or UDP
dst_ip:	destination IP address
dst_mask:	destination IP mask
dst_port_start:	destination port start, only when protocol is TCP or UDP
dst_port_end:	destination port end, only when protocol is TCP or UDP
log:	0(no log), 1(log), 2(only log transparent sessions)
dscp:	DSCP
vlan:	VLAN ID
fragments:	IP fragments
tcp_est:	TCP established, only when protocol is TCP or UDP
hits:	hit number

Example Response1

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
access-list 100 permit icmp type 17 code 200 100.100.100.0 0.0.0.255 host
    200.200.200.200 fragments dscp 63 log
access-list 100 13-vlan-fwd-disable tcp 101.101.101.0 0.0.0.255 eq 500 host
    102.102.102.102 range 100 200 vlan 9 dscp 63 established log transparent-
        session-only
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxx&method=network.acl.search&id=100

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
<ACL_List>
    <total_num>1</total_num>
    <ACL>
        <id>100</id>
        <remark/>
        <ACL_Item>
            <id>100</id>
```

```

<seq_num>10</seq_num>
<action>1</action>
<protocol>1</protocol>
<icmp_type>17</icmp_type>
<icmp_code>200</icmp_code>
<src_ip>100.100.100.0</src_ip>
<src_mask>0.0.0.255</src_mask>
<dst_ip>200.200.200.200</dst_ip>
<dst_mask>255.255.255.255</dst_mask>
<fragments>1</fragments>
<dscp>63</dscp>
<tcp_est>0</tcp_est>
<log>1</log>
<hits>0</hits>
</ACL_Item>
<ACL_Item>
<id>100</id>
<seq_num>20</seq_num>
<action>2</action>
<protocol>6</protocol>
<icmp_type>244</icmp_type>
<icmp_code>244</icmp_code>
<src_ip>101.101.101.0</src_ip>
<src_mask>0.0.0.255</src_mask>
<src_port_start>500</src_port_start>
<src_port_end>500</src_port_end>
<dst_ip>102.102.102.102</dst_ip>
<dst_mask>255.255.255.255</dst_mask>
<dst_port_start>100</dst_port_start>
<dst_port_end>200</dst_port_end>
<fragments>0</fragments>
<vlan>9</vlan>
<dscp>63</dscp>
<tcp_est>1</tcp_est>
<log>2</log>
<hits>0</hits>
</ACL_Item>
</ACL>
</ACL_List>
</response>

```

Example Response2

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
ipv6 access-list ipv6_acl_name1
deny icmp host 2002::2 any fragments vlan 100 dscp 63 log transparent-session-only
permit ipv6 any any fragments vlan 9 dscp 1 log transparent-session-only
13-vlan-fwd-disable udp any any
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxxxx&method=network.acl.search&name=ipv6_acl_name1

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
<ACL_List>
  <total_num>1</total_num>
  <ACL>
    <name>ipv6_acl_name1</name>
    <remark/>
    <ACL_Item>
```

```

<name>ipv6_acl_name1</name>
<seq_num>10</seq_num>
<action>0</action>
<protocol>58</protocol>
<src_ip>2002::2</src_ip>
<src_mask>128</src_mask>
<dst_ip>::</dst_ip>
<dst_mask>0</dst_mask>
<fragments>1</fragments>
<vlan>100</vlan>
<dscp>63</dscp>
<log>2</log>
<hits>0</hits>
</ACL_Item>
<ACL_Item>
<name>ipv6_acl_name1</name>
<seq_num>20</seq_num>
<action>1</action>
<protocol>41</protocol>
<src_ip>::</src_ip>
<src_mask>0</src_mask>
<dst_ip>::</dst_ip>
<dst_mask>0</dst_mask>
<fragments>1</fragments>
<vlan>9</vlan>
<dscp>1</dscp>
<tcp_est>0</tcp_est>
<log>2</log>
<hits>0</hits>
</ACL_Item>
<ACL_Item>
<name>ipv6_acl_name1</name>
<seq_num>30</seq_num>
<action>2</action>
<protocol>6</protocol>
<src_ip>::</src_ip>
<src_mask>0</src_mask>
<dst_ip>::</dst_ip>
<dst_mask>0</dst_mask>
<fragments>0</fragments>
<tcp_est>0</tcp_est>
<log>0</log>
<hits>0</hits>
</ACL_Item>
</ACL>
</ACL_List>
</response>

```

Menus Privilege

Config Mode >> Network >> ACL >> Standard

Config Mode >> Network >> ACL >> Extended

Config Mode >> Network >> ACL >> IPv6

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

6.9.3 “network.acl.create” Method

This method is used to add an entry to an Access Control List (ACL).

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	network.acl.create	String		

If you want to add an IPv4 standard ACL, please use the following parameter list

Parameter Name	Description	Data Type	Range	Default
id(*)	ACL ID, 1 – 99	Int	1 - 99	
seq_num(*)	ACL sequence number	int	1 - 4094	
action	0: drop, 1: permit, 2: l3-vlan-fwd-disable	Int	0, 1 or 2	0
src_ip	ACL source address	String		
src_mask	ACL source mask	String		
log	0: no log, 1: log, 2: only log transparent sessions	Int	0, 1 or 2	0
remark (**)	ACL remark	String		

If you want to add an IPv4 extened ACL, please use the following parameter list

Parameter Name	Description	Data Type	Range	Default
id(*)	ACL ID, 100 - 199	Int	100 - 199	
seq_num(*)	ACL sequence number	Int	1 - 4094	
action	0: drop, 1: permit, 2: l3-vlan-fwd-disable	Int	0, 1 or 2	0
protocol(*)	protocol(icmp, ip, tcp, udp)	String		
icmp_code	ICMP code, only when protocol is ICMP	Int	0 - 254	
icmp_type	ICMP type, only when protocol is ICMP	int	0 - 254	
src_ip	ACL source address	String		
src_mask	ACL source mask	String		
src_port_start	source port start, only when protocol is TCP or UDP	Int	1 - 65535	0
src_port_end	Source port end, only when protocol is TCP or UDP	Int	1 - 65535	0
dst_ip	Destination ip address	String		
dst_mask	Destination ip mask	String		
dst_port_start	Destination port start, only when protocol is TCP or UDP	Int	1 - 65535	0
dst_port_end	Destination port end, only when protocol is TCP or UDP	Int	1 - 65535	0
dscp	DSCP	Int	1- 63	
vlan	VLAN ID	Int	1 - 4094	
fragments	IP fragments	Int	0 or 1	0
tcp_est	TCP established, only when protocol is TCP or UDP	Int	0 or 1	0
log	0: no log, 1: log, 2: only log transparent sessions	Int	0, 1 or 2	0
remark (**)	ACL remark	String		

If you want to add an IPv6 ACL, please use the following parameter list

Parameter Name	Description	Data Type	Range	Default
name(*)	ACL name	String		
seq_num	Acl sequence number	Int	1- 4094	
action	0: deny, 1: permit, 2: l3-vlan-fwd-disable	Int	0, 1 or 2	0
protocol(*)	protocol(icmp, ip, tcp, udp)	String		
src_ip	ACL source address	String		
src_mask	ACL source mask	String		
src_port_start	source port start, only when protocol is TCP or UDP	Int	1 - 65535	0
src_port_end	Source port end, only when protocol is TCP or UDP	Int	1 - 65535	0
dst_ip	Destination ip address	String		
dst_mask	Destination ip mask	String		
dst_port_start	Destination port start, only when protocol is TCP or UDP	Int	1 - 65535	0
dst_port_end	Destination port end, only when protocol is TCP or UDP	Int	1 - 65535	0
dscp	DSCP	Int	1- 63	
vlan	VLAN ID	Int	1 - 4094	
fragments	IP fragments	Int	0 or 1	0
tcp_est	TCP established, only when protocol is TCP or UDP	Int	0 or 1	0
log	0: no log, 1: log, 2: only log transparent sessions	Int	0, 1 or 2	0
remark (**)	ACL remark	String		

Note: (*) parameter is required in the API.

Either remark or other ACL entry is valid.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=network.acl.create&name=ipv6_acl_name1&seq_num=10&
protocol=udp&src_ip=4ff6::5555&src_mask=64&dst_ip=3ff7::8888&dst_mask=64&dst_port_start=7
7&dst_port_end=99
```

CLI configuration after aXAPI call:

```
!
ipv6 access-list ipv6_acl_name1
  deny udp 4ff6::5555/64 3ff7::8888/64 range 77 99
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Network >> ACL >> Standard
Config Mode >> Network >> ACL >> Extended
Config Mode >> Network >> ACL >> IPv6

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

6.9.4 “network.acl.update” Method

This method is used to update one or more parameters in an Access Control List (ACL).

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	network.acl.update	String		

If you want to add an IPv4 standard ACL, please use the following parameter list

Parameter Name	Description	Data Type	Range	Default
id(*)	ACL ID, 1 – 99	Int	1 - 99	
seq_num(*)	ACL sequence number	int	1 - 4094	
action	0: drop, 1: permit, 2: l3-vlan-fwd-disable	Int	0, 1 or 2	0
src_ip	ACL source address	String		
src_mask	ACL source mask	String		
log	0: no log, 1: log, 2: only log transparent sessions	Int	0, 1 or 2	0
remark (**)	ACL remark	String		

If you want to add an IPv4 extened ACL, please use the following parameter list

Parameter Name	Description	Data Type	Range	Default
<code>id(*)</code>	ACL ID, 100 - 199	Int	100 - 199	
<code>seq_num(*)</code>	ACL sequence number	Int	1 - 4094	
<code>action</code>	0: drop, 1: permit, 2: l3-vlan-fwd-disable	Int	0, 1 or 2	0
<code>protocol(*)</code>	protocol(icmp, ip, tcp, udp)	String		
<code>icmp_code</code>	ICMP code, only when protocol is ICMP	Int	0 - 254	
<code>icmp_type</code>	ICMP type, only when protocol is ICMP	int	0 - 254	
<code>src_ip</code>	ACL source address	String		
<code>src_mask</code>	ACL source mask	String		
<code>src_port_start</code>	source port start, only when protocol is TCP or UDP	Int	1 - 65535	0
<code>src_port_end</code>	Source port end, only when protocol is TCP or UDP	Int	1 - 65535	0
<code>dst_ip</code>	Destination ip address	String		
<code>dst_mask</code>	Destination ip mask	String		
<code>dst_port_start</code>	Destination port start, only when protocol is TCP or UDP	Int	1 - 65535	0
<code>dst_port_end</code>	Destination port end, only when protocol is TCP or UDP	Int	1 - 65535	0
<code>dscp</code>	DSCP	Int	1- 63	
<code>vlan</code>	VLAN ID	Int	1 - 4094	
<code>fragments</code>	IP fragments	Int	0 or 1	0
<code>tcp_est</code>	TCP established, only when protocol is TCP or UDP	Int	0 or 1	0
<code>log</code>	0: no log, 1: log, 2: only log transparent sessions	Int	0, 1 or 2	0
<code>remark (**)</code>	ACL remark	String		

If you want to add an IPv6 ACL, please use the following parameter list

Parameter Name	Description	Data Type	Range	Default
<code>name(*)</code>	ACL name	String		
<code>seq_num(*)</code>	acl sequence number	Int	1- 4094	
<code>action</code>	0: drop, 1: permit, 2: l3-vlan-fwd-disable	Int	0, 1 or 2	0
<code>protocol(*)</code>	protocol(icmp, ip, tcp, udp)	String		
<code>src_ip</code>	ACL source address	String		
<code>src_mask</code>	ACL source mask	String		
<code>src_port_start</code>	source port start, only when protocol is TCP or UDP	Int	1 - 65535	0
<code>src_port_end</code>	Source port end, only when protocol is TCP or UDP	Int	1 - 65535	0
<code>dst_ip</code>	Destination ip address	String		
<code>dst_mask</code>	Destination ip mask	String		
<code>dst_port_start</code>	Destination port start, only when protocol is TCP or UDP	Int	1 - 65535	0
<code>dst_port_end</code>	Destination port end, only when protocol is TCP or UDP	Int	1 - 65535	0
<code>dscp</code>	DSCP	Int	1- 63	
<code>vlan</code>	VLAN ID	Int	1 - 4094	
<code>fragments</code>	IP fragments	Int	0 or 1	0
<code>tcp_est</code>	TCP established, only when protocol is TCP or UDP	Int	0 or 1	0
<code>log</code>	0: no log, 1: log, 2: only log transparent sessions	Int	0, 1 or 2	0
<code>remark (**)</code>	ACL remark	String		

Note: (*) parameter is required in the API.

Either remark or other ACL entry is valid.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
ipv6 access-list ipv6_acl_name1
deny udp 4ff6::5555/64 3ff7::8888/64 range 77 99
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxxxx&method=network.acl.update&name=ipv6_acl_name1&seq_num=10&
protocol=udp&src_ip=4ff6::5555&src_mask=64&dst_ip=3ff7::8888&dst_mask=64&dst_port_start=7
7&dst_port_end=111
```

CLI configuration after aXAPI call:

```
!
ipv6 access-list ipv6_acl_name1
deny udp 4ff6::5555/64 3ff7::8888/64 range 77 111
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Network >> ACL >> Standard
Config Mode >> Network >> ACL >> Extended
Config Mode >> Network >> ACL >> IPv6

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

6.9.5 “network.acl.delete” Method

This method is used to delete an Access Control List (ACL).

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	network.acl.delete	String		
id(*)	ACL ID, only for IPv4 ACL, 1 - 199	Int	1 - 199	
name(*)	ACL name, only for IPv6 ACL	String		
seq_num	sequence number	int	1 - 4094	

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response1

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
ipv6 access-list test
permit ipv6 any any vlan 101
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxxxx&method=network.acl.delete&name=test
```

CLI configuration after aXAPI call:

```
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Example Response2

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
access-list 2 permit host 1.1.1.1 log
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxxxx&method=network.acl.delete&id=2
```

CLI configuration after aXAPI call:

```
!
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Network >> ACL >> Standard

Config Mode >> Network >> ACL >> Extended

Config Mode >> Network >> ACL >> IPv6

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

6.10 DNS

6.10.1 “**network.dns.server.get**” Method

This method is used to get DNS parameters, such as the primary/secondary DNS servers, or the DNS suffix.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	network.dns.server.get	String		

Note: (*) parameter is required in the API.

Response Fields

dns	XML tag for the DNS
primary_dns	primary DNS server
secondary_dns	secondary DNS server
dns_suffix	DNS suffix

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
ip dns primary 1.1.1.1
ip dns secondary 2.2.2.2
ip dns suffix a10networks.com
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxxxx&method=network.dns.server.get

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
  <dns>
    <primary_dns>1.1.1.1</primary_dns>
    <secondary_dns>2.2.2.2</secondary_dns>
    <dns_suffix>a10networks.com</dns_suffix>
  </dns>
</response>
```

Menus Privilege

Config Mode >> Network >> DNS

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

6.10.2 “network.dns.server.set” Method

This method is used to set such DNS parameters as the primary/secondary DNS servers or DNS suffix.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	network.dns.hostname.set	String		
primary_dns	primary DNS server	String		
secondary_dns	secondary DNS server	String		
dns_suffix	DNS suffix	String		

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!  
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=network.dns.server.set&primary_dns=192.168.3.224&secondary_dns=192.168.3.225&dns_suffix=a10networks.com
```

CLI configuration after aXAPI call:

```
!  
ip dns primary 192.168.3.224  
ip dns secondary 192.168.3.225  
ip dns suffix a10networks.com  
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>  
<response status="ok">  
</response>
```

Menus Privilege

Config Mode >> Network >> DNS

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7 SLB Management APIs

7.1 Service Group

7.1.1 “slb.service_group.getAll” Method

This method is used to get configuration information for all service groups with information for each group member, such as server name, server port number, and priority.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.service_group.getAll	String		

Note: (*) parameter is required in the API.

Response Fields

The response XML is the collection of service groups, and each service group will have a collection of the group members.

service_group_list	XML tag for the collection of the service groups
service_group	XML tag for the service group
name	service group name.
protocol	service group type, either TCP(2) or UDP(3).
lb_method	desired load-balancing algorithm on the service group with: 0 RoundRobin 1 WeightedRoundRobin 2 LeastConnection 3 WeightedLeastConnection 4 LeastConnectionOnServicePort 5 WeightedLeastConnectionOnServicePort 6 FastResponseTime 7 LeastRequest 8 StrictRoundRobin 9 StateLessSourceIPHash 10 StateLessSourceIPHashOnly 11 StateLessDestinationIPHash 12 StateLessSourceDestinationIPHash 13 StateLessPerPacketRoundRobin
health_monitor	health monitor method used at the service group member.
min_active_member	min active members.
status	status of min active members, enabled(1) or disabled(0)
number	number of min active members.
priority_set	priority option of min active members: 0 Do nothing 1 Skip Priority Set 2 Dynamic PriorityRequired, please fill in a valid number.(1 - 63)
client_reset	send client reset when server selection fail enabled(1) or disabled(0)
stats_data	stats data, either enabled(1) or disabled(0)
extended_stats	extended stats, either enabled(1) or disabled(0)

member_list	XML tag for service group members
member	XML tag for a single service group member, and contains the following properties:
server	server name (or IPv4, IPv6 address) of this member
port	server port number
template	server port template name
priority	member priority
stats_data	member stats data, either enabled(1) or disabled(0)

Example Response

Request as:

- **URL:**
https://_AX_IP_Address_/services/rest/V2/
- **HTTP Action:**
GET

CLI configuration before aXAPI call:

```
!
slb service-group http tcp
method least-request
min-active-member 63 dynamic-priority
health-check ping7
reset-on-server-selection-fail
extended-stats
member _s_124.254.3.3:80
member _s_124.254.3.1:80
!
```

1. HTTP Request in URL Format:

session_id=123456789&method=slb.service_group.getAll&format=url

Response as the HTTP body (URL Format):

```
<response status="ok">
  <service_group_list>
    <service_group>
      <name>http</name>
      <protocol>2</protocol>
      <lb_method>7</lb_method>
      <health_monitor>ping7</health_monitor>
      <min_active_member>
        <status>1</status>
        <number>63</number>
        <priority_set>1</priority_set>
      </min_active_member>
      <client_reset>1</client_reset>
      <stats_data>1</stats_data>
      <extended_stats>1</extended_stats>
      <member_list>
        <member>
          <server>_s_124.254.3.3</server>
          <port>80</port>
          <template>default</template>
          <priority>1</priority>
          <stats_data>1</stats_data>
        </member>
        <member>
          <server>_s_124.254.3.1</server>
          <port>80</port>
          <template>default</template>
          <priority>1</priority>
          <stats_data>1</stats_data>
        </member>
      </member_list>
    </service_group>
  </service_group_list>
</response>
```

```

        </member>
    </member_list>
</service_group>
</service_group_list>
</response>

```

2. HTTP Request in JSON Format:

session_id=123456789&method=slb.service_group.getAll&format=json

Response as the HTTP body (JSON Format):

```

{
    "service_group_list": [
        {
            "name": "http",
            "protocol": 2,
            "lb_method": 7,
            "health_monitor": "ping7",
            "min_active_member": {
                {
                    "status": 1,
                    "number": 63,
                    "priority_set": 2
                },
                "backup_server_event_log_enable": 0,
                "client_reset": 1,
                "stats_data": 1,
                "extended_stats": 1,
                "member_list": [
                    {
                        "server": "_s_124.254.3.1",
                        "port": 80,
                        "template": "default",
                        "priority": 1,
                        "status": 1,
                        "stats_data": 1
                    },
                    {
                        "server": "_s_124.254.3.3",
                        "port": 80,
                        "template": "default",
                        "priority": 1,
                        "status": 1,
                        "stats_data": 1
                    }
                ]
            },
            {
                "name": "g8",
                "protocol": 3,
                "lb_method": 0,
                "health_monitor": "",
                "min_active_member": {
                    {
                        "status": 0,
                        "priority_set": 0
                    },
                    "backup_server_event_log_enable": 0,
                    "stats_data": 1,
                    "extended_stats": 0,
                    "member_list": []
                }
            }
        ]
    }
}

```

Menus Privilege

Config Mode >> Service >> SLB >> Service Group

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.1.2 “slb.service_group.search” Method

This method is used to get service group and members for a given service group name.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.service-group.search	String		
name (*)	Service group name	String		

Note: (*) parameter is required in the API.

Response Fields

The response XML is a service group if exists, and with a collection of the group members.

service_group	XML tag for the service group
name	service group name.
protocol	service group type, either TCP(2) or UDP(3).
lb_method	desired load-balancing algorithm on the service group with:
0	RoundRobin
1	WeightedRoundRobin
2	LeastConnection
3	WeightedLeastConnection
4	LeastConnectionOnServicePort
5	WeightedLeastConnectionOnServicePort
6	FastResponseTime
7	LeastRequest
8	StrictRoundRobin
9	StateLessSourceIPHash
10	StateLessSourceIPHashOnly
11	StateLessDestinationIPHash
12	StateLessSourceDestinationIPHash
13	StateLessPerPackageRoundRobin
health_monitor	health monitor method used at the service group member.
min_active_member	min active members.
status	status of min active members, enabled(1) or disabled(0)
number	number of min active members.
priority_set	priority option of min active members:
0	Do nothing
1	Skip Priority Set
2	Dynamic PriorityRequired, please fill in a valid number. (1 - 63)
client_reset	send client reset when server selection fail enabled(1) or disabled(0)
stats_data	stats data, either enabled(1) or disabled(0)
extended_stats	extended stats, either enabled(1) or disabled(0)

member_list	XML tag for service group members
member	XML tag for a single service group member, and contains the following properties:
server	server name (or IPv4, IPv6 address) of this member
port	server port number
template	server port template name
priority	member priority
stats_data	member stats data, either enabled(1) or disabled(0)

Example Response

Request as:

- **URL:**
https://_AX_IP_Address_/services/rest/V2/
- **HTTP Action:**
GET

CLI configuration before aXAPI call:

```
!
slb service-group http tcp
method least-request
min-active-member 63 dynamic-priority
health-check ping7
reset-on-server-selection-fail
extended-stats
member _s_124.254.3.3:80
member _s_124.254.3.1:80
!
```

1. HTTP Request in URL Format:

session_id=123456789&method=slb.service_group.search&name=http&format=url

Response as the HTTP body (URL Format):

```
<response status="ok">
<service_group>
  <name>http</name>
  <protocol>2</protocol>
  <lb_method>7</lb_method>
  <health_monitor>ping7</health_monitor>
  <min_active_member>
    <status>1</status>
    <number>63</number>
    <priority_set>1</priority_set>
  </min_active_member>
  <client_reset>1</client_reset>
  <stats_data>1</stats_data>
  <extended_stats>1</extended_stats>
  <member_list>
    <member>
      <server>_s_124.254.3.3</server>
      <port>80</port>
      <template>default</template>
      <priority>1</priority>
      <stats_data>1</stats_data>
    </member>
    <member>
      <server>_s_124.254.3.1</server>
      <port>80</port>
      <template>default</template>
      <priority>1</priority>
      <stats_data>1</stats_data>
    </member>
  </member_list>
</service_group>

```

```

</member_list>
</service_group>
</response>

```

2. HTTP Request in JSON Format:

session_id=123456789&method=slb.service_group.search&name=http&format=json

Response as the HTTP body (JSON Format):

```

{
    "service_group":
    {
        "name": "http",
        "protocol": 2,
        "lb_method": 7,
        "health_monitor": "ping7",
        "min_active_member":
        {
            "status": 1,
            "number": 63,
            "priority_set": 2
        },
        "backup_server_event_log_enable": 0,
        "client_reset": 1,
        "stats_data": 1,
        "extended_stats": 1,
        "member_list": [
            {
                "server": "_s_124.254.3.1",
                "port": 80,
                "template": "default",
                "priority": 1,
                "status": 1,
                "stats_data": 1
            },
            {
                "server": "_s_124.254.3.3",
                "port": 80,
                "template": "default",
                "priority": 1,
                "status": 1,
                "stats_data": 1
            }
        ]
    }
}

```

Menus Privilege

Config Mode >> Service >> SLB >> Service Group

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.1.3 “slb.service_group.fetchAllStatistics” Method

This method is used to return the statistical data for all service groups and their members that have been configured on the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
----------------	-------------	-----------	-------	---------

session_id (*)	user authenticated session id.	String		
method (*)	slb.service_group.fetchAllStatistics	String		

Note: (*) parameter is required in the API.

Response Fields

The response XML is the statistical information for all service groups, including the statistics for each service group members.

service_group_stat_list XML tag for the collection of the service group statistics.

service_group_stat	XML tag for the service group statistics
name	service group name.
protocol	L4 protocol of this service group, TCP(2) or UDP(3)
status	member status, disabled(0), all members up (1), partition up(2), functional up (3), down (4), unknow (5)
cur_conns	total number of current connections
tot_conns	total number of connections, ulong64
req_pkts	total number of request packets received, ulong64
resp_pkts	total number of response packets sent, ulong64
req_bytes	total number of request bytes received, ulong64
resp_bytes	total number of response bytes sent, ulong64
cur_reqs	total number of current requests
tot_reqs	total number of requests, ulong64
tot_succ_reqs	total number of successful requests, ulong64
member_stat_list	XML tag for service group members
member_stat	XML tag for a single service group member, and contains the following properties:
server	server IPv4 or IPv6 address
port	server port number
status	member status, up(1), down (4), unknown(5)
cur_conns	total number of current connections
tot_conns	total number of connections, ulong64
req_pkts	total number of request packets received, ulong64
resp_pkts	total number of response packets sent, ulong64
req_bytes	total number of request bytes received, ulong64
resp_bytes	total number of response bytes sent, ulong64
cur_reqs	total number of current requests
tot_reqs	total number of requests, ulong64
tot_succ_reqs	total number of successful requests, ulong64

Example Response

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
slb service-group http_grp_01 tcp
  member _s_124.254.3.3:80
  member _s_124.254.3.1:80
!
```

1. HTTP Request in URL Format:

session_id=123456789&method=slb.service_group.fetchAllStatistics&format=url

Response as the HTTP body (URL Format):

```
<response status="ok">
  <service_group_stat_list>
    <service_group_stat>
      <name>http_grp_01</name>
      <protocol>2</protocol>
      <status>4</status>
      <cur_conns>0</cur_conns>
      <tot_conns>0</tot_conns>
      <req_pkts>0</req_pkts>
      <resp_pkts>0</resp_pkts>
      <req_bytes>0</req_bytes>
      <resp_bytes>0</resp_bytes>
      <cur_reqs>0</cur_reqs>
      <total_req>0</total_req>
      <total_req_succ>0</total_req_succ>
      <member_stat_list>
        <member_stat>
          <server>s_124.254.3.3</server>
          <port>80</port>
          <status>4</status>
          <cur_conns>0</cur_conns>
          <tot_conns>0</tot_conns>
          <req_pkts>0</req_pkts>
          <resp_pkts>0</resp_pkts>
          <req_bytes>0</req_bytes>
          <resp_bytes>0</resp_bytes>
          <cur_reqs>0</cur_reqs>
          <total_req>0</total_req>
          <total_req_succ>0</total_req_succ>
        </member_stat>
        <member_stat>
          <server>s_124.254.3.1</server>
          <port>80</port>
          <status>4</status>
          <cur_conns>0</cur_conns>
          <tot_conns>0</tot_conns>
          <req_pkts>0</req_pkts>
          <resp_pkts>0</resp_pkts>
          <req_bytes>0</req_bytes>
          <resp_bytes>0</resp_bytes>
          <cur_reqs>0</cur_reqs>
          <total_req>0</total_req>
          <total_req_succ>0</total_req_succ>
        </member_stat>
      </member_stat_list>
    </service_group_stat>
  </service_group_stat_list>
</response>
```

2. HTTP Request in JSON Format:

```
session_id=123456789&method=slb.service_group.fetchAllStatistics&format=json
```

Response as the HTTP body (JSON Format):

```
{
  "service_group_stat_list": [
    {
      "name": "http_grp_01",
      "protocol": 2,
      "status": 4,
      "cur_conns": 0,
```

```

"tot_conns":0,
"req_pkts":0,
"resp_pkts":0,
"req_bytes":0,
"resp_bytes":0,
"cur_reqs":0,
"total_reqs":0,
"total_reqs_succ":0,
"member_stat_list": [
{
    "server": "_s_124.254.3.3",
    "port":80,
    "status":4,
    "cur_conns":0,
    "tot_conns":0,
    "req_pkts":0,
    "resp_pkts":0,
    "req_bytes":0,
    "resp_bytes":0,
    "cur_reqs":0,
    "total_reqs":0,
    "total_reqs_succ":0
},
{
    "server": "_s_124.254.3.1",
    "port":80,
    "status":4,
    "cur_conns":0,
    "tot_conns":0,
    "req_pkts":0,
    "resp_pkts":0,
    "req_bytes":0,
    "resp_bytes":0,
    "cur_reqs":0,
    "total_reqs":0,
    "total_reqs_succ":0
}
]
}
]
}

```

Menus Privilege

Monitor Mode >> Service >> SLB >> Service Group

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.1.4 “slb.service_group.fetchStatistics” Method

This method is used to return the statistical data for the service group and associated members for a given service group name.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.service_group.fetchStatistics	String		
name (*)	Service group name	String		

Note: (*) parameter is required in the API.

Response Fields

The response XML is the statistical information for the service group if exists, and with a collection of the group member's statistics.

service_group_stat	XML tag for the service group statistics
name	service group name.
protocol	L4 protocol of this service group, TCP(2) or UDP(3)
status	member status, disabled(0), all members up (1), partition up(2), functional up (3), down (4), unknown (5)
cur_conns	total number of current connections
tot_conns	total number of connections, ulong64
req_pkts	total number of request packets received, ulong64
resp_pkts	total number of response packets sent, ulong64
req_bytes	total number of request bytes received, ulong64
resp_bytes	total number of response bytes sent, ulong64
cur_reqs	total number of current requests
tot_reqs	total number of requests, ulong64
tot_succ_reqs	total number of successful requests, ulong64
member_stat_list	XML tag for service group members
member_stat	XML tag for a single service group member, and contains the following properties:
server	server IPv4 or IPv6 address
port	server port number
status	member status, up(1), down (4), unknown(5)
cur_conns	total number of current connections
tot_conns	total number of connections, ulong64
req_pkts	total number of request packets received, ulong64
resp_pkts	total number of response packets sent, ulong64
req_bytes	total number of request bytes received, ulong64
resp_bytes	total number of response bytes sent, ulong64
cur_reqs	total number of current requests
tot_reqs	total number of requests, ulong64
tot_succ_reqs	total number of successful requests, ulong64

Example Response

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
slb service-group http_grp_01 tcp
  member _s_124.254.3.3:80
  member _s_124.254.3.1:80
!
```

1. HTTP Request in URL Format:

session_id=123456789&method=slb.service_group.fetchStatistics&name=http_srv_grp1&format=url

Response as the HTTP body (URL Format):

```
<responsestatus="ok">
  <service_group_stat>
    <name>http</name>
```

```

<protocol>2</protocol>
<status>4</status>
<cur_conns>0</cur_conns>
<tot_conns>0</tot_conns>
<req_pkts>0</req_pkts>
<resp_pkts>0</resp_pkts>
<req_bytes>0</req_bytes>
<resp_bytes>0</resp_bytes>
<cur_reqs>0</cur_reqs>
<total_req>0</total_req>
<total_req_succ>0</total_req_succ>
<member_stat_list>
    <member_stat>
        <server>_s_124.254.3.3</server>
        <port>80</port>
        <status>4</status>
        <cur_conns>0</cur_conns>
        <tot_conns>0</tot_conns>
        <req_pkts>0</req_pkts>
        <resp_pkts>0</resp_pkts>
        <req_bytes>0</req_bytes>
        <resp_bytes>0</resp_bytes>
        <cur_reqs>0</cur_reqs>
        <total_req>0</total_req>
        <total_req_succ>0</total_req_succ>
    </member_stat>
    <member_stat>
        <server>_s_124.254.3.1</server>
        <port>80</port>
        <status>4</status>
        <cur_conns>0</cur_conns>
        <tot_conns>0</tot_conns>
        <req_pkts>0</req_pkts>
        <resp_pkts>0</resp_pkts>
        <req_bytes>0</req_bytes>
        <resp_bytes>0</resp_bytes>
        <cur_reqs>0</cur_reqs>
        <total_req>0</total_req>
        <total_req_succ>0</total_req_succ>
    </member_stat>
</member_stat_list>
</service_group_stat>
</response>

```

2. HTTP Request in JSON Format:

session_id=123456789&method=slb.service_group.fetchStatistics&name=http_srv_grp1&format=json

Response as the HTTP body (JSON Format):

```
{
    "service_group_stat": {
        "name": "http_srv_grp1",
        "protocol": 2,
        "status": 4,
        "cur_conns": 0,
        "tot_conns": 0,
        "req_pkts": 0,
        "resp_pkts": 0,
        "req_bytes": 0,
        "resp_bytes": 0,
        "cur_reqs": 0,
        "total_reqs": 0,
        "total_reqs_succ": 0,
    }
}
```

```

"member_stat_list": [
{
    "server": "_s_124.254.3.3",
    "port": 80,
    "status": 4,
    "cur_conns": 0,
    "tot_conns": 0,
    "req_pkts": 0,
    "resp_pkts": 0,
    "req_bytes": 0,
    "resp_bytes": 0,
    "cur_reqs": 0,
    "total_reqs": 0,
    "total_reqs_succ": 0
},
{
    "server": "_s_124.254.3.1",
    "port": 80,
    "status": 4,
    "cur_conns": 0,
    "tot_conns": 0,
    "req_pkts": 0,
    "resp_pkts": 0,
    "req_bytes": 0,
    "resp_bytes": 0,
    "cur_reqs": 0,
    "total_reqs": 0,
    "total_reqs_succ": 0
}
]
}
}

```

Menus Privilege

Monitor Mode >> Service >> SLB >> Service Group

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.1.5 “slb.service_group.create” Method

This method is used to create a new service group with members.

This method is required to use the HTTP POST method.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.service-group.create	String		
name (*)	Service group name	String		
protocol(*)	Service group protocol type, TCP(2) or UDP(3)	Int	2 or 3	
lb_method	Desired load-balancing algorithm for the service group 0 RoundRobin 1 WeightedRoundRobin 2 LeastConnection 3 WeightedLeastConnection 4 LeastConnectionOnServicePort 5 WeightedLeastConnectionOnServicePort 6 FastResponseTime 7 LeastRequest 8 StrictRoundRobin 9 StateLessSourceIPHash 10 StateLessSourceIPHashOnly	Int	0 - 13	0

	11 StateLessDestinationIPHash 12 StateLessSourceDestinationIPHash 13 StateLessPerPackageRoundRobin			
min_active_member status	Min active members The status of min active member configuration, enabled (1) or disabled (0)	Int	0 - 1	0
number priority_set	number of min active members 0 do nothing 1 Skip Priority Set 2 Dynamic priority	Int Int	1 .. 63 0 - 2	0 0
client_reset	Send client reset when server selection fail, enabled(1) or disabled(0), TCP only.	Int	1 or 0	0
stats_data	Stats data, enabled(1) or disabled(0)	Int	1 or 0	0
extended_stats	Extended stats, enabled(1) or disabled(0)	Int	1 or 0	0
health_monitor	Health monitor method used at the service group member	String		(default)
member_list(*)	Service group member list to be added member1^Bmember2^B ... memberN ^B: ASCII Code 0x02, URL-encode %02	String		
member<n> server (*) port (*) template priority stats_data	Member at element <n> IPv4/IPv6 address, server name or server's DNS name Port number Server port template Member priority Member stats data Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03	String Int String Int Int	0-65534 1 .. 10 1 or 0	default 1 1

Note: (*) parameter is required in the API.

Note: The ‘server’ parameter of members can be an IPv6 address, IPv4 address, DNS name or preconfigured real server name in AX. When the real server name is used, the service group member will use this real server. Otherwise, this method will create a real server for you, and name it with _s_ prefix.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
slb server _s_server02 server2
    port 8080 udp
!
slb server _s_1.1.1.10 1.1.1.10
    port 8080 udp
!
slb server _s_1.1.1.11 1.1.1.11
    port 80 udp
!
slb server _s_3ff5_8888 3ff5::8888
    port 8099 udp
!
```

1. HTTP Request in URL Format:

```
session_id=xxxxxxxxxxxxxxxxxxxxx&method=slb.service_group.create&service_group=name
%03test9%02protocol%033%02lb_method%033%02health_monitor%03ping%02min_active_member%02sta
ts_data%030%02extended_stats%030%02member_list&min_active_member=status%030%02number%035%
02priority_set%031&member_list=m1%02m2%02m3%02m4&m1=server%03server02%02port%038080&m2=se
rver%031.1.1.10%02port%038080&m3=server%031.1.1.11%02port%0380&m4=server%033ff5::8888%02p
ort%038099%02template%03default%02stats_data%031%02priority%0316&format=url
```

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in JSON Format:

```
session_id=xxxxxxxxxxxxxxxxxxxxx&method=slb.service_group.create&format=json
```

Request POST data (JSON Format):

```
{
    "service_group": {
        "name": "test9",
        "protocol": 3,
        "lb_method": 3,
        "health_monitor": "ping",
        "min_active_member": {
            "status": 0,
            "priority_set": 0
        },
        "backup_server_event_log_enable": 0,
        "stats_data": 1,
        "extended_stats": 0,
        "member_list": [
            {
                "server": "_s_server02",
                "port": 8080,
                "template": "default",
                "priority": 1,
                "status": 1,
                "stats_data": 1
            },
            {
                "server": "_s_3ff5_8888",
                "port": 8099,
                "template": "default",
                "priority": 16,
                "status": 1,
                "stats_data": 1
            },
            {
                "server": "_s_1.1.1.11",
                "port": 80,
                "template": "default",
                "priority": 1,
                "status": 1,
                "stats_data": 1
            },
            {
                "server": "_s_1.1.1.10",
                "port": 8080,
                "template": "default",
                "priority": 1,
                "status": 1
            }
        ]
    }
}
```

```

        "stats_data":1
    }
]
}

```

Response as the HTTP body (JSON Format):

```

{
  "response":
  {
    "status": "OK"
  }
}

```

CLI configuration after aXAPI call:

```

!
slb service-group test9 udp
  method weighted-least-connection
  health-check ping
  member _s_server02:8080
  member _s_1.1.1.10:8080
  member _s_1.1.1.11:80
  member _s_3ff5_8888:8099 priority 16
!
```

Menus Privilege

Config Mode >> Service >> SLB >> Service Group

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.1.6 “slb.service_group.update” Method

This method is used to update an SLB service group.

This method is required to use the HTTP POST method.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.service-group.update	String		
name (*)	service group name	String		
protocol(*)	service group protocol type, TCP(2) or UDP(3)	Int	2 or 3	
lb_method	Desired load-balancing algorithm for the service group 0 RoundRobin 1 WeightedRoundRobin 2 LeastConnection 3 WeightedLeastConnection 4 LeastConnectionOnServicePort 5 WeightedLeastConnectionOnServicePort 6 FastResponseTime 7 LeastRequest 8 StrictRoundRobin 9 StateLessSourceIPHash 10 StateLessSourceIPHashOnly	Int	0 - 13	0

	11 StateLessDestinationIPHash 12 StateLessSourceDestinationIPHash 13 StateLessPerPackageRoundRobin			
min_active_member	Min active members	Int	0 - 1	0
status	The status of min active member configuration, enabled (1) or disabled (0)	Int	1 .. 63	0
number	number of min active members	Int	0 - 2	0
priority_set	0 Do nothing 1 Skip Priority Set 2 Dynamic priority	Int		
client_reset	Send client reset when server selection fail, enabled(1) or disabled(0), TCP only.	Int	1 or 0	0
stats_data	Stats data, enabled(1) or disabled(0)	Int	1 or 0	0
extended_stats	Extended stats, enabled(1) or disabled(0)	Int	1 or 0	0
health_monitor	Health monitor method used at the service group member	String		(default)
member_list	Service group member list to be added member1^Bmember2^B ... memberN ^B: ASCII Code 0x02, URL-encode %02	String		
member<n>	Member at element <n>	String		
server (*)	IPv4/IPv6 address, server name or server's DNS name			
port (*)	Port number	Int	0-65534	
template	Server port template	String		
priority	Member priority	Int	1 .. 10	1
stats_data	Member stats data	Int	1 or 0	1
	Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03			

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Example

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
slb service-group test9 udp
  method round-robin
  health-check ping
  member _s_server02:8080
  member _s_1.1.1.10:8080
!
!
slb server _s_server02 server2
  port 8080  udp
!
slb server _s_1.1.1.10 1.1.1.10
  port 8080  udp
!
slb server _s_1.1.1.11 1.1.1.11
  port 80  udp
!
slb server _s_3ff5_8888 3ff5::8888
  port 8099  udp
!
```

1. HTTP Request in URL Format:

```
session_id=123456789&method=slb.service_group.update&service_group=name%03test9%02proto
1%033%02lb_method%033%02health_monitor%03ping%02min_active_member%02stats_data%030%02exte
nded_stats%030%02member_list&min_active_member=status%030%02number%035%02priority_set%031
&member_list=m1%02m2%02m3%02m4&m1=server%03server02%02port%038080&m2=server%031.1.1.10%02
port%038080&m3=server%031.1.1.11%02port%0380&m4=server%033ff5::8888%02port%038099%02templ
ate%03default%02stats_data%031%02priority%0316&format=url
```

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in JSON Format:

```
session_id=123456789&method=slb.service_group.update&format=json
```

Request POST data (JSON Format):

```
{
  "service_group": {
    "name": "test9",
    "protocol": 3,
    "lb_method": 3,
    "health_monitor": "ping",
    "min_active_member": {
      "status": 0,
      "priority_set": 0
    },
    "backup_server_event_log_enable": 0,
    "stats_data": 1,
    "extended_stats": 0,
    "member_list": [
      {
        "server": "_s_server02",
        "port": 8080,
        "template": "default",
        "priority": 1,
        "status": 1,
        "stats_data": 1
      },
      {
        "server": "_s_3ff5_8888",
        "port": 8099,
        "template": "default",
        "priority": 16,
        "status": 1,
        "stats_data": 1
      },
      {
        "server": "_s_1.1.1.11",
        "port": 80,
        "template": "default",
        "priority": 1,
        "status": 1,
        "stats_data": 1
      },
      {
        "server": "_s_1.1.1.10",
        "port": 8080,
```

```

        "template":"default",
        "priority":1,
        "status":1,
        "stats_data":1
    }
]
}
}

```

Response as the HTTP body (JSON Format):

```
{
  "response":
  {
    "status": "OK"
  }
}
```

CLI configuration after aXAPI call:

```
!
slb service-group test9 udp
  method weighted-least-connection
  health-check ping
  member _s_server02:8080
  member _s_1.1.1.10:8080
  member _s_1.1.1.11:80
  member _s_3ff5_8888:8099 priority 16
!
```

Menus Privilege

Config Mode >> Service >> SLB >> Service Group

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.1.7 “slb.service_group.delete” Method

This method is used to get delete a service group.

This method is required in order to use the HTTP POST method.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.service-group.delete	String		
name (*)	service group name	String		

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:

POST

CLI configuration before aXAPI call:

```
!  
slb service-group http_srv_grp1 udp  
method weighted-least-connection  
health-check ping  
member _s_server02:8080  
member _s_1.1.1.10:8080  
member _s_1.1.1.11:80  
member _s_3ff5_8888:8099 priority 16  
!
```

1. HTTP Request in URL Format:

```
session_id=123456789&method=slb.service_group.delete&name=http_srv_grp1&format=url
```

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>  
<response status="ok">  
</response>
```

2. HTTP Request in JSON Format:

```
session_id=123456789&method=slb.service_group.delete&name=http_srv_grp1&format=json
```

Request POST data (JSON Format)::

```
{  
    "name": "http_srv_grp1"  
}
```

Response as the HTTP body (JSON Format):

```
{  
    "response":  
    {  
        "status": "OK"  
    }  
}
```

CLI configuration after aXAPI call:

```
!  
!  
!
```

Menus Privilege

Config Mode >> Service >> SLB >> Service Group

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.1.8 “slb.service_group.member.create” Method

This method is used to add a new member to a service group.

This method is required to use the HTTP POST method.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.service_group.member.create	String		
name (*)	the name of the service group.	String		
member	member object			
server (*)	IPv4/IPv6 address, server name or server's DNS name	String		
port (*)	Port number	Int	0 .. 65334	
template	Server port template	String		
priority	Member priority	Int	1 .. 10	default
stats_data	Member stats data	Int	1 or 0	1

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
slb service-group http tcp
method least-request
min-active-member 63 dynamic-priority
health-check ping7
reset-on-server-selection-fail
extended-stats
member _s_124.254.3.3:80
member _s_124.254.3.1:80
!
!
slb server _s_1.1.1.3 1.1.1.3
  port 8080  tcp
!
```

1. HTTP Request in URL Format:

```
session_id=123456789&method=slb.service_group.member.create&name=http&member=serve
r%031.1.1.3%02port%038080&format=url
```

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in JSON Format:

```
session_id=123456789&method=slb.service_group.member.create&format=json
```

Request POST data (JSON Format)::

```
{  
    "name": "http",  
    "member":  
    {  
        "server": "_s_1.1.1.3",  
        "port": 8080  
    }  
}
```

Response as the HTTP body (JSON Format):

```
{  
    "response":  
    {  
        "status": "OK"  
    }  
}
```

CLI configuration after aXAPI call:

```
!  
slb service-group http tcp  
method least-request  
min-active-member 63 dynamic-priority  
health-check ping7  
reset-on-server-selection-fail  
extended-stats  
member _s_124.254.3.3:80  
member _s_124.254.3.1:80  
member _s_1.1.1.3:8080  
!
```

Menus Privilege

Config Mode >> Service >> SLB >> Service Group

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.1.9 “`slb.service_group.member.update`” Method

This method is used to update one or more parameters, such as server name, port number, or IPv4/v6 address, for an SLB service group.

This method is required to use the HTTP POST method.

Parameters

Parameter Name	Description	Data Type	Range	Default
<code>session_id (*)</code>	user authenticated session id.	String		
<code>method (*)</code>	<code>slb.service_group.member.update</code>	String		
<code>name (*)</code>	the name of the service group.	String		
<code>member</code>	member object			
<code>server (*)</code>	IPv4/IPv6 address, server name or server's DNS name	String		
<code>port (*)</code>	Port number	Int	0 .. 65534	
<code>template</code>	Server port template	String		
<code>priority</code>	Member priority	Int	1 .. 10	default 1

stats_data	Member stats data	Int	1 or 0	1
-------------------	-------------------	-----	--------	---

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
slb service-group http tcp
method least-request
min-active-member 63 dynamic-priority
health-check ping7
reset-on-server-selection-fail
extended-stats
member _s_124.254.3.3:80
member _s_124.254.3.1:80
member _s_1.1.1.3:8080
!
```

1. HTTP Request in URL Format:

```
session_id=123456789&method=slb.service_group.member.update&name=http&member=serve
r%031.1.1.3%02port%038080%02priority%0316&format=url
```

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in JSON Format:

```
session_id=123456789&method=slb.service_group.member.update&format=url
```

Request POST data (JSON Format)::

```
{
  "name": "http",
  "member": [
    {
      "server": "_s_1.1.1.3",
      "port": 8080,
      "priority": 16
    }
  ]
}
```

Response as the HTTP body (JSON Format):

```
{
```

```

"response":
{
    "status" : "OK"
}

```

CLI configuration after aXAPI call:

```

!
slb service-group http tcp
method least-request
min-active-member 63 dynamic-priority
health-check ping7
reset-on-server-selection-fail
extended-stats
member _s_124.254.3.3:80
member _s_124.254.3.1:80
member _s_1.1.1.3:8080 priority 16
!
```

Menus Privilege

Config Mode >> Service >> SLB >> Service Group

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.1.10 “**slb.service_group.member.delete**” Method

This method is used to get delete a member of a service group.

This method is required to use the HTTP POST method.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.service_group.member.delete	String		
name (*)	the name of the service group.	String		
member server (*) port (*)	member object IPv4/IPv6 address, server name or server's DNS name Port number	String Int	0-65534	

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```

!
slb service-group http tcp
method least-request

```

```

min-active-member 63 dynamic-priority
health-check ping7
reset-on-server-selection-fail
extended-stats
member _s_124.254.3.3:80
member _s_124.254.3.1:80
member _s_1.1.1.3:8080 priority 16
!
```

1. HTTP Request in URL Format:

```
session_id=123456789&method=slb.service_group.member.delete&name=http&member=serve
r%031.1.1.3%02port%038080&format=url
```

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in JSON Format:

```
session_id=123456789&method=slb.service_group.member.delete&format=json
```

Request POST data (JSON Format)::

```
{
  "name": "http",
  "member": {
    {
      "server": "_s_1.1.1.3",
      "port": 8080
    }
  }
}
```

Response as the HTTP body (JSON Format):

```
{
  "response": {
    {
      "status": "OK"
    }
  }
}
```

CLI configuration after aXAPI call:

```
!
slb service-group http tcp
  method least-request
  min-active-member 63 dynamic-priority
  health-check ping7
  reset-on-server-selection-fail
  extended-stats
  member _s_124.254.3.3:80
  member _s_124.254.3.1:80
!
```

Menus Privilege

Config Mode >> Service >> SLB >> Service Group

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.2 Server

7.2.1 “slb.server.getAll” Method

This method is used to get configuration information for all servers configured on the AX.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.server.getAll	String		

Note: (*) parameter is required in the API.

Response Fields

The response XML is the collection of servers, and each server will have a collection of the ports.

server_list	XML tag for the collection of servers
server	XML tag for the server
name	Server name
host	Server IP address or dns name
gslb_external_address	GSLB external IP address
health_monitor	Server health-monitor name, empty means the default
status	Server status, enabled(1) or disabled(0)
weight	Server weight
conn_limit	Server connection limit
conn_limit_log	Server connection limit logging, enabled(1) or disabled(0)
conn_resume	Server connection resume
stats_data	Server stat data option, enabled(1) or disabled(0)
extended_stats	Server extended stats, enabled(1) or disabled(0)
slow_start	Server slow start option, enabled(1) or disabled(0)
spoofing_cache	Server spoofing cache option, enabled(1) or disabled(0)
template	Server template name, empty means the default
port_list	XML tag for the server ports
port	XML tag for a server port, and contains the following properties:
port_num	server port number
protocol	either TCP(2) or UDP(3) protocol type
weight	server port weight
no_ssl	server port no ssl, enabled(1) or disabled(0)
conn_limit	server port connection limit
conn_limit_log	server port connection limit logging, enabled(1) or disabled(0)
conn_resume	server port connection resume
template	server port template name, empty means the default
stats_data	server port stat data option, enabled(1) or disabled(0)
extended_stats	server port extended stats, enabled(1) or disabled(0)
health_monitor	server port health monitor name, empty means the default. Only when follow port is not set
follow_port	XML tag of follow port, only when health monitor is not set
follow_port_num	follow port number
follow_port_type	follow port type

Example Response

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
slb server server01 124.254.1.133
  external-ip 124.254.6.12
  template server servert
  health-check ping7
  conn-resume 100
    weight 100
  slow-start
  spoofing-cache
  extended-stats
  port 21  tcp
    template port portt
    no-ssl
    health-check follow-port 80 tcp
    weight 100
    conn-resume 100
    extended-stats
  port 80  tcp
    extended-stats
slb server server02 3ff2::9933
  port 8080  tcp
  port 21  tcp
slb server server03 ABCDEFG
  external-ip 12.12.33.33
  conn-resume 1000000
    weight 100
  slow-start
  extended-stats
  port 20  tcp
  port 80  tcp
!
```

1. HTTP Request in URL Format:

session_id=123456789&method=slb.server.getAll&format=url

Response as the HTTP body (URL Format):

```
<response status="ok">
<server_list>
<server>
  <name>server01</name>
  <host>124.254.1.133</host>
  <gslb_external_address>124.254.6.12</gslb_external_address>
  <weight>100</weight>
  <health_monitor>ping7</health_monitor>
  <status>1</status>
  <conn_limit>8000000</conn_limit>
  <conn_limit_log>1</conn_limit_log>
  <conn_resume>100</conn_resume>
  <stats_data>1</stats_data>
  <extended_stats>1</extended_stats>
  <slow_start>1</slow_start>
  <spoofing_cache>1</spoofing_cache>
  <template>servert</template>
<port_list>
<port>
  <port_num>21</port_num>
```

```

<protocol>2</protocol>
<weight>100</weight>
<no_ssl>1</no_ssl>
<conn_limit>8000000</conn_limit>
<conn_limit_log>1</conn_limit_log>
<conn_resume>100</conn_resume>
<template>portt</template>
<stats_data>1</stats_data>
<follow_port>
<follow_port_num>80</follow_port_num>
<follow_port_type>2</follow_port_type>
</follow_port>
<extended_stats>1</extended_stats>
</port>
<port>
<port_num>80</port_num>
<protocol>2</protocol>
<weight>1</weight>
<no_ssl>0</no_ssl>
<conn_limit>8000000</conn_limit>
<conn_limit_log>1</conn_limit_log>
<conn_resume>0</conn_resume>
<template>default</template>
<stats_data>1</stats_data>
<extended_stats>0</extended_stats>
</port>
</port_list>
</server>
<server>
<name>server02</name>
<host>3ff2::9933</host>
<gslb_external_address>0.0.0.0</gslb_external_address>
<weight>1</weight>
<health_monitor>(default)</health_monitor>
<status>1</status>
<conn_limit>8000000</conn_limit>
<conn_limit_log>1</conn_limit_log>
<conn_resume>0</conn_resume>
<stats_data>0</stats_data>
<extended_stats>0</extended_stats>
<slow_start>0</slow_start>
<spoofing_cache>0</spoofing_cache>
<template>default</template>
<port_list>
<port>
<port_num>8080</port_num>
<protocol>3</protocol>
<weight>0</weight>
<no_ssl>0</no_ssl>
<conn_limit>8000000</conn_limit>
<conn_limit_log>0</conn_limit_log>
<conn_resume>0</conn_resume>
<template>default</template>
<stats_data>1</stats_data>
<extended_stats>0</extended_stats>
</port>
<port>
<port_num>21</port_num>
<protocol>2</protocol>
<weight>1</weight>
<no_ssl>0</no_ssl>
<conn_limit>8000000</conn_limit>

```

```

<conn_limit_log>1</conn_limit_log>
<conn_resume>0</conn_resume>
<template>default</template>
<stats_data>1</stats_data>
<extended_stats>0</extended_stats>
</port>
</port_list>
</server>
<server>
<name>server03</name>
<host>ABCDEFG</host>
<gslb_external_address>12.12.33.33</gslb_external_address>
<weight>100</weight>
<health_monitor>(default)</health_monitor>
<status>1</status>
<conn_limit>8000000</conn_limit>
<conn_limit_log>1</conn_limit_log>
<conn_resume>1000000</conn_resume>
<stats_data>1</stats_data>
<extended_stats>0</extended_stats>
<slow_start>1</slow_start>
<spoofing_cache>1</spoofing_cache>
<template>default</template>
<port_list>
<port>
<port_num>20</port_num>
<protocol>3</protocol>
<weight>1</weight>
<no_ssl>1</no_ssl>
<conn_limit>8000000</conn_limit>
<conn_limit_log>0</conn_limit_log>
<conn_resume>100</conn_resume>
<template>default</template>
<stats_data>1</stats_data>
<follow_port>
<follow_port_num>80</follow_port_num>
<follow_port_type>2</follow_port_type>
</follow_port>
<extended_stats>1</extended_stats>
</port>
<port>
<port_num>80</port_num>
<protocol>2</protocol>
<weight>1</weight>
<no_ssl>0</no_ssl>
<conn_limit>8000000</conn_limit>
<conn_limit_log>1</conn_limit_log>
<conn_resume>0</conn_resume>
<template>default</template>
<stats_data>1</stats_data>
<extended_stats>0</extended_stats>
</port>
</port_list>
</server>
</server_list>
</response>

```

2. HTTP Request in JSON Format:

session_id=123456789&method=slb.server.getAll&format=json

Response as the HTTP body (JSON Format):

```
{  
  "server_list": [  
    {  
      "name": "server01",  
      "host": "124.254.1.133",  
      "gslb_external_address": "124.254.6.12",  
      "weight": 100,  
      "health_monitor": "ping7",  
      "status": 1,  
      "conn_limit": 8000000,  
      "conn_limit_log": 1,  
      "conn_resume": 100,  
      "stats_data": 1,  
      "extended_stats": 1,  
      "slow_start": 1,  
      "spoofing_cache": 1,  
      "template": "servert",  
      "port_list": [  
        {  
          "port_num": 80,  
          "protocol": 2,  
          "status": 1,  
          "weight": 1,  
          "no_ssl": 0,  
          "conn_limit": 8000000,  
          "conn_limit_log": 1,  
          "conn_resume": 0,  
          "template": "default",  
          "stats_data": 1,  
          "health_monitor": "",  
          "extended_stats": 1  
        },  
        {  
          "port_num": 21,  
          "protocol": 2,  
          "status": 1,  
          "weight": 100,  
          "no_ssl": 1,  
          "conn_limit": 8000000,  
          "conn_limit_log": 1,  
          "conn_resume": 100,  
          "template": "portt",  
          "stats_data": 1,  
          "follow_port":  
            {  
              "follow_port_num": 80,  
              "follow_port_type": 2  
            },  
          "extended_stats": 1  
        }  
      ]  
    },  
    {  
      "name": "server02",  
      "host": "3ff2::9933",  
      "gslb_external_address": "0.0.0.0",  
      "weight": 1,  
      "health_monitor": "(default)",  
      "status": 1,  
      "conn_limit": 8000000,  
      "conn_limit_log": 1,  
      "conn_resume": 0,  
      "stats_data": 1,  
      "extended_stats": 1  
    }  
  ]  
}
```

```

"extended_stats":0,
"slow_start":0,
"spoofing_cache":0,
"template":"default",
"port_list":[
{
    "port_num":21,
    "protocol":2,
    "status":1,
    "weight":1,
    "no_ssl":0,
    "conn_limit":8000000,
    "conn_limit_log":1,
    "conn_resume":0,
    "template":"default",
    "stats_data":1,
    "health_monitor":"",
    "extended_stats":0
},
{
    "port_num":8080,
    "protocol":2,
    "status":1,
    "weight":1,
    "no_ssl":0,
    "conn_limit":8000000,
    "conn_limit_log":1,
    "conn_resume":0,
    "template":"default",
    "stats_data":1,
    "health_monitor":"",
    "extended_stats":0
}
]
},
{
    "name":"server03",
    "host":"ABCDEFG",
    "gslb_external_address":"12.12.33.33",
    "weight":100,
    "health_monitor":"(default)",
    "status":1,
    "conn_limit":8000000,
    "conn_limit_log":1,
    "conn_resume":1000000,
    "stats_data":1,
    "extended_stats":1,
    "slow_start":1,
    "spoofing_cache":0,
    "template":"default",
    "port_list":[
{
    "port_num":80,
    "protocol":2,
    "status":1,
    "weight":1,
    "no_ssl":0,
    "conn_limit":8000000,
    "conn_limit_log":1,
    "conn_resume":0,
    "template":"default",
    "stats_data":1,
    "health_monitor":"",
    "extended_stats":0
}
]
}

```

```

    "port_num":20,
    "protocol":2,
    "status":1,
    "weight":1,
    "no_ssl":0,
    "conn_limit":8000000,
    "conn_limit_log":1,
    "conn_resume":0,
    "template":"default",
    "stats_data":1,
    "health_monitor":"",
    "extended_stats":0
  }
]
}
]
}

```

Menus Privilege

Config Mode >> Service >> SLB >> Server

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.2.2 “slb.server.search” Method

This method is used to get server information (with ports) for a given server by server name, IP address, or DNS name.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.server.search	String		
name (**)	Server name	String		
host (**)	Server identifier, ip address or dns name	String		

Note: (*) parameter is required in the API.

(**) Either one of parameters is required.

Response Fields

The response XML is a server, and server will have a collection of the ports.

server	XML tag for the server
name	Server name
host	Server IP address or dns name
gslb_external_address	GSLB external IP address
health_monitor	Server health-monitor name, empty means the default
status	Server status, enabled(1) or disabled(0)
weight	Server weight
conn_limit	Server connection limit
conn_limit_log	Server connection limit logging, enabled(1) or disabled(0)
conn_resume	Server connection resume
stats_data	Server stat data option, enabled(1) or disabled(0)
extended_stats	Server extended stats, enabled(1) or disabled(0)
slow_start	Server slow start option, enabled(1) or disabled(0)
spoofing_cache	Server spoofing cache option, enabled(1) or disabled(0)

template	Server template name, empty means the default
port_list	XML tag for the server ports
port	XML tag for a server port, and contains the following properties:
port_num	server port number
protocol	either TCP(2) or UDP(3) protocol type
weight	server port weight
no_ssl	server port no ssl, enabled(1) or disabled(0)
conn_limit	server port connection limit
conn_limit_log	server port connection limit logging, enabled(1) or disabled(0)
conn_resume	server port connection resume
template	server port template name, empty means the default
stats_data	server port stat data option, enabled(1) or disabled(0)
extended_stats	server port extended stats, enabled(1) or disabled(0)
health_monitor	server port health monitor name, empty means the default. Only when follow port is not set
follow_port	XML tag of follow port, only when health monitor is not set
follow_port_num	follow port number
follow_port_type	follow port type

Example Response

Example1: Search the server by given server address.

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
slb server server01 124.254.1.133
  external-ip 124.254.6.12
  template server servert
  health-check ping7
  conn-resume 100
    weight 100
  slow-start
  spoofing-cache
  extended-stats
  port 21 tcp
    template port portt
    no-ssl
    health-check follow-port 80 tcp
    weight 100
    conn-resume 100
    extended-stats
  port 80 tcp
    extended-stats
!
!
```

1. HTTP Request in URL Format:

session_id=123456789&method=slb.server.search&name=server01&format=url

Response as the HTTP body (URL Format):

```
<response status="ok">
<server>
  <name>server01</name>
  <host>124.254.1.133</host>
  <gslb_external_address>124.254.6.12</gslb_external_address>
  <weight>100</weight>
```

```

<health_monitor>ping7</health_monitor>
<status>1</status>
<conn_limit>8000000</conn_limit>
<conn_limit_log>1</conn_limit_log>
<conn_resume>100</conn_resume>
<stats_data>1</stats_data>
<extended_stats>1</extended_stats>
<slow_start>1</slow_start>
<spoofing_cache>1</spoofing_cache>
<template>servert</template>
<port_list>
<port>
<port_num>21</port_num>
<protocol>2</protocol>
<weight>100</weight>
<no_ssl>1</no_ssl>
<conn_limit>8000000</conn_limit>
<conn_limit_log>1</conn_limit_log>
<conn_resume>100</conn_resume>
<template>portt</template>
<stats_data>1</stats_data>
<follow_port>
<follow_port_num>80</follow_port_num>
<follow_port_type>2</follow_port_type>
</follow_port>
<extended_stats>1</extended_stats>
</port>
<port>
<port_num>80</port_num>
<protocol>2</protocol>
<weight>1</weight>
<no_ssl>0</no_ssl>
<conn_limit>8000000</conn_limit>
<conn_limit_log>1</conn_limit_log>
<conn_resume>0</conn_resume>
<template>default</template>
<stats_data>1</stats_data>
<extended_stats>0</extended_stats>
</port>
</port_list>
</server>
</response>

```

2. HTTP Request in JSON Format:

session_id=123456789&method=slb.server.search&name=server01&format=json

Response as the HTTP body (JSON Format):

```
{
  "server": {
    "name": "server01",
    "host": "124.254.1.133",
    "gslb_external_address": "124.254.6.12",
    "weight": 100,
    "health_monitor": "ping7",
    "status": 1,
    "conn_limit": 8000000,
    "conn_limit_log": 1,
  }
}
```

```

"conn_resume":100,
"stats_data":1,
"extended_stats":1,
"slow_start":1,
"spoofing_cache":1,
"template":"server",
"port_list":[
{
  "port_num":80,
  "protocol":2,
  "status":1,
  "weight":1,
  "no_ssl":0,
  "conn_limit":8000000,
  "conn_limit_log":1,
  "conn_resume":0,
  "template":"default",
  "stats_data":1,
  "health_monitor":"(default)",
  "extended_stats":1
},
{
  "port_num":21,
  "protocol":2,
  "status":1,
  "weight":100,
  "no_ssl":1,
  "conn_limit":8000000,
  "conn_limit_log":1,
  "conn_resume":100,
  "template":"portt",
  "stats_data":1,
  "follow_port":
  {
    "follow_port_num":80,
    "follow_port_type":2
  },
  "extended_stats":1
}
]
}
}

```

Example2: Search the server by given hostname.

session_id=123456789&method=slb.real_server.search&host=sj-http-in01

Example3: Search the server by given IPv6 address.

session_id=123456789&method=slb.real_server.search&host=sj-3ff1::2222

Example4: Search the server by given IPv4 address.

session_id=123456789&method=slb.real_server.search&host=10.0.0.126

Menus Privilege

Config Mode >> Service >> SLB >> Server

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.2.3 “slb.server.fetchAllStatistics” Method

This method is used to get all server statistics information (with their ports) that have been configured on the AX.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.server.fetchAllStatistics	String		

Note: (*) parameter is required in the API.

Response Fields

The response XML is the statistical data for all servers, including the statistics for each server port.

Server_stat_list	XML tag for the collection of server statistics
Server_stat	XML tag for the server statistics
name	Server name.
host	Server IP address (IPv4 or IPv6) or dns name
status	Server status, disabled (0), running (1), down (2)
cur_conns	total number of current connections
tot_conns	total number of connections, ulong64
req_pkts	total number of request packets received, ulong64
resp_pkts	total number of response packets sent, ulong64
req_bytes	total number of request bytes received, ulong64
resp_bytes	total number of response bytes sent, ulong64
cur_reqs	total number of current requests
total_reqs	total number of requests, ulong64
total_reqs_succ	total number of successful requests, ulong64
port_stat_list	XML tag for the server ports
port_stat	XML tag for a single server port, and contains the following properties:
port_num	server port number
protocol	L3 protocol. TCP(2) or UDP(3)
status	member status disabled (0), running (1), down (2)
cur_conns	total number of current connections
tot_conns	total number of connections, ulong64
req_pkts	total number of request packets received, ulong64
resp_pkts	total number of response packets sent, ulong64
req_bytes	total number of request bytes received, ulong64
resp_bytes	total number of response bytes sent, ulong64
cur_reqs	total number of current requests
total_reqs	total number of requests, ulong64
total_reqs_succ	total number of successful requests, ulong64

Example Response

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

!

```

slb server server01 124.254.1.133
  external-ip 124.254.6.12
  template server servert
  health-check ping7
  conn-resume 100
    weight 100
  slow-start
  spoofing-cache
  extended-stats
  port 21  tcp
    template port portt
    no-ssl
    health-check follow-port 80 tcp
    weight 100
    conn-resume 100
    extended-stats
  port 80  tcp
    extended-stats
slb server server02 3ff2::9933
  port 8080  tcp
  port 21  tcp
slb server server3 ABCDEFG
  external-ip 12.12.33.33
  conn-resume 1000000
    weight 100
  slow-start
  extended-stats
  port 20  tcp
  port 80  tcp
!

```

1. HTTP Request in URL Format:

session_id=123456789&method=slb.server.fetchAllStatistics&format=url

Response as the HTTP body (URL Format):

```

<response status="ok">
  <server_stat_list>
    <server_stat>
      <name>server01</name>
      <host>124.254.1.133</host>
      <status>2</status>
      <cur_conns>0</cur_conns>
      <tot_conns>0</tot_conns>
      <req_pkts>0</req_pkts>
      <resp_pkts>0</resp_pkts>
      <req_bytes>0</req_bytes>
      <resp_bytes>0</resp_bytes>
      <cur_reqs>0</cur_reqs>
      <total_req>0</total_req>
      <total_req_succ>0</total_req_succ>
    <port_stat_list>
      <port_stat>
        <port_num>21</port_num>
        <protocol>2</protocol>
        <status>2</status>
        <cur_conns>0</cur_conns>
        <tot_conns>0</tot_conns>
        <req_pkts>0</req_pkts>
        <resp_pkts>0</resp_pkts>
        <req_bytes>0</req_bytes>
        <resp_bytes>0</resp_bytes>
        <cur_reqs>0</cur_reqs>
        <total_req>0</total_req>

```

```

        <total_req_succ>0</total_req_succ>
    </port_stat>
    <port_stat>
        <port_num>80</port_num>
        <protocol>2</protocol>
        <status>2</status>
        <cur_conns>0</cur_conns>
        <tot_conns>0</tot_conns>
        <req_pkts>0</req_pkts>
        <resp_pkts>0</resp_pkts>
        <req_bytes>0</req_bytes>
        <resp_bytes>0</resp_bytes>
        <cur_reqs>0</cur_reqs>
        <total_req>0</total_req>
        <total_req_succ>0</total_req_succ>
    </port_stat>
</port_stat_list>
</server_stat>
<server_stat>
    <name>server02</name>
    <host>3ff2::9933</host>
    <status>2</status>
    <cur_conns>0</cur_conns>
    <tot_conns>0</tot_conns>
    <req_pkts>0</req_pkts>
    <resp_pkts>0</resp_pkts>
    <req_bytes>0</req_bytes>
    <resp_bytes>0</resp_bytes>
    <cur_reqs>0</cur_reqs>
    <total_req>0</total_req>
    <total_req_succ>0</total_req_succ>
<port_stat_list>
    <port_stat>
        <port_num>8080</port_num>
        <protocol>3</protocol>
        <status>2</status>
        <cur_conns>0</cur_conns>
        <tot_conns>0</tot_conns>
        <req_pkts>0</req_pkts>
        <resp_pkts>0</resp_pkts>
        <req_bytes>0</req_bytes>
        <resp_bytes>0</resp_bytes>
        <cur_reqs>0</cur_reqs>
        <total_req>0</total_req>
        <total_req_succ>0</total_req_succ>
    </port_stat>
    <port_stat>
        <port_num>21</port_num>
        <protocol>2</protocol>
        <status>2</status>
        <cur_conns>0</cur_conns>
        <tot_conns>0</tot_conns>
        <req_pkts>0</req_pkts>
        <resp_pkts>0</resp_pkts>
        <req_bytes>0</req_bytes>
        <resp_bytes>0</resp_bytes>
        <cur_reqs>0</cur_reqs>
        <total_req>0</total_req>
        <total_req_succ>0</total_req_succ>
    </port_stat>
</port_stat_list>
</server_stat>

```

```

<server_stat>
  <name>server03</name>
  <host>ABCDEFG</host>
  <status>1</status>
  <cur_conns>0</cur_conns>
  <tot_conns>0</tot_conns>
  <req_pkts>0</req_pkts>
  <resp_pkts>0</resp_pkts>
  <req_bytes>0</req_bytes>
  <resp_bytes>0</resp_bytes>
  <cur_reqs>0</cur_reqs>
  <total_req>0</total_req>
  <total_req_succ>0</total_req_succ>
  <port_stat_list>
    <port_stat>
      <port_num>20</port_num>
      <protocol>3</protocol>
      <status>2</status>
      <cur_conns>0</cur_conns>
      <tot_conns>0</tot_conns>
      <req_pkts>0</req_pkts>
      <resp_pkts>0</resp_pkts>
      <req_bytes>0</req_bytes>
      <resp_bytes>0</resp_bytes>
      <cur_reqs>0</cur_reqs>
      <total_req>0</total_req>
      <total_req_succ>0</total_req_succ>
    </port_stat>
    <port_stat>
      <port_num>80</port_num>
      <protocol>2</protocol>
      <status>2</status>
      <cur_conns>0</cur_conns>
      <tot_conns>0</tot_conns>
      <req_pkts>0</req_pkts>
      <resp_pkts>0</resp_pkts>
      <req_bytes>0</req_bytes>
      <resp_bytes>0</resp_bytes>
      <cur_reqs>0</cur_reqs>
      <total_req>0</total_req>
      <total_req_succ>0</total_req_succ>
    </port_stat>
    </port_stat_list>
  </server_stat>
</server_stat_list>
</response>

```

2. HTTP Request in JSON Format:

session_id=123456789&method=slb.server.fetchAllStatistics&format=json

Response as the HTTP body (JSON Format):

```
{
  "server_stat_list": [
    {
      "name": "server01",
      "host": "124.254.1.133",
      "status": 2,
      "cur_conns": 0,
      "port_stat_list": [
        {
          "port_num": 20,
          "protocol": 3,
          "status": 2,
          "cur_conns": 0,
          "tot_conns": 0,
          "req_pkts": 0,
          "resp_pkts": 0,
          "req_bytes": 0,
          "resp_bytes": 0,
          "cur_reqs": 0,
          "total_req": 0,
          "total_req_succ": 0
        },
        {
          "port_num": 80,
          "protocol": 2,
          "status": 2,
          "cur_conns": 0,
          "tot_conns": 0,
          "req_pkts": 0,
          "resp_pkts": 0,
          "req_bytes": 0,
          "resp_bytes": 0,
          "cur_reqs": 0,
          "total_req": 0,
          "total_req_succ": 0
        }
      ]
    }
  ]
}
```

```
"tot_conns":0,
"req_pkts":0,
"resp_pkts":0,
"req_bytes":0,
"resp_bytes":0,
"cur_reqs":0,
"total_reqs":0,
"total_reqs_succ":0,
"port_stat_list":[
{
    "port_num":80,
    "protocol":2,
    "status":2,
    "cur_conns":0,
    "tot_conns":0,
    "req_pkts":0,
    "resp_pkts":0,
    "req_bytes":0,
    "resp_bytes":0,
    "cur_reqs":0,
    "total_reqs":0,
    "total_reqs_succ":0
},
{
    "port_num":21,
    "protocol":2,
    "status":2,
    "cur_conns":0,
    "tot_conns":0,
    "req_pkts":0,
    "resp_pkts":0,
    "req_bytes":0,
    "resp_bytes":0,
    "cur_reqs":0,
    "total_reqs":0,
    "total_reqs_succ":0
}
],
},
{
    "name":"server02",
    "host":"3ff2::9933",
    "status":2,
    "cur_conns":0,
    "tot_conns":0,
    "req_pkts":0,
    "resp_pkts":0,
    "req_bytes":0,
    "resp_bytes":0,
    "cur_reqs":0,
    "total_reqs":0,
    "total_reqs_succ":0,
    "port_stat_list":[
{
    "port_num":21,
    "protocol":2,
    "status":2,
    "cur_conns":0,
    "tot_conns":0,
    "req_pkts":0,
    "resp_pkts":0,
    "req_bytes":0,
    "resp_bytes":0,
    "cur_reqs":0,
    "total_reqs":0,
    "total_reqs_succ":0
}
]
```

```
        },
        {
            "port_num":8080,
            "protocol":2,
            "status":2,
            "cur_conns":0,
            "tot_conns":0,
            "req_pkts":0,
            "resp_pkts":0,
            "req_bytes":0,
            "resp_bytes":0,
            "cur_reqs":0,
            "total_reqs":0,
            "total_reqs_succ":0
        }
    ]
},
{
    "name":"server03",
    "host":"ABCDEFG",
    "status":2,
    "cur_conns":0,
    "tot_conns":0,
    "req_pkts":0,
    "resp_pkts":0,
    "req_bytes":0,
    "resp_bytes":0,
    "cur_reqs":0,
    "total_reqs":0,
    "total_reqs_succ":0,
    "port_stat_list":[
        {
            "port_num":80,
            "protocol":2,
            "status":2,
            "cur_conns":0,
            "tot_conns":0,
            "req_pkts":0,
            "resp_pkts":0,
            "req_bytes":0,
            "resp_bytes":0,
            "cur_reqs":0,
            "total_reqs":0,
            "total_reqs_succ":0
        },
        {
            "port_num":20,
            "protocol":2,
            "status":2,
            "cur_conns":0,
            "tot_conns":0,
            "req_pkts":0,
            "resp_pkts":0,
            "req_bytes":0,
            "resp_bytes":0,
            "cur_reqs":0,
            "total_reqs":0,
            "total_reqs_succ":0
        }
    ]
}
]
```

Menus Privilege

Monitor Mode >> Service >> SLB >> Server

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.2.4 “slb.server.fetchStatistics” Method

This method is used to get the server statistics information (with ports) by entering a given server name or host identifier.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.server.fetchStatistics	String		
name (**)	server name	String		
host (**)	host identifier, IPv4, IPv6 or DNS name	String		

Note: (*) parameter is required in the API.

(**) Either one of parameters is required.

Response Fields

Server_stat	XML tag for the server statistics
name	Server name.
host	Server IP address (IPv4 or IPv6) or dns name
status	Server status, disabled(0), running(1), down(2)
cur_conns	total number of current connections
tot_conns	total number of connections, ulong64
req_pkts	total number of request packets received, ulong64
resp_pkts	total number of response packets sent, ulong64
req_bytes	total number of request bytes received, ulong64
resp_bytes	total number of response bytes sent, ulong64
cur_reqs	total number of current requests
total_reqs	total number of requests, ulong64
total_reqs_succ	total number of successful requests, ulong64
port_stat_list	XML tag for the server ports
port_stat	XML tag for a single server port, and contains the following properties:
port_num	server port number
protocol	L3 protocol. TCP(2) or UDP(3)
status	member status disabled(0), running(1), down(2)
cur_conns	total number of current connections
tot_conns	total number of connections, ulong64
req_pkts	total number of request packets received, ulong64
resp_pkts	total number of response packets sent, ulong64
req_bytes	total number of request bytes received, ulong64
resp_bytes	total number of response bytes sent, ulong64
cur_reqs	total number of current requests
total_reqs	total number of requests, ulong64
total_reqs_succ	total number of successful requests, ulong64

Example Response

Example:

Request as:

- URL:
https:// AX_IP_Address /services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
slb server server01 124.254.1.133
  external-ip 124.254.6.12
  template server servert
  health-check ping7
  conn-resume 100
    weight 100
  slow-start
  spoofing-cache
  extended-stats
  port 21  tcp
    template port portt
    no-ssl
    health-check follow-port 80  tcp
    weight 100
    conn-resume 100
    extended-stats
  port 80  tcp
    extended-stats
!
```

1. HTTP Request in URL Format:

```
session_id=123456789&method=slb.server.fetchStatistics&name=server01&format=url
```

Response as the HTTP body (URL Format):

```
<response status="ok">
  <server_stat>
    <name>server01</name>
    <host>124.254.1.133</host>
    <status>2</status>
    <cur_conns>0</cur_conns>
    <tot_conns>0</tot_conns>
    <req_pkts>0</req_pkts>
    <resp_pkts>0</resp_pkts>
    <req_bytes>0</req_bytes>
    <resp_bytes>0</resp_bytes>
    <cur_reqs>0</cur_reqs>
    <total_req>0</total_req>
    <total_req_succ>0</total_req_succ>
  <port_stat_list>
    <port_stat>
      <port_num>21</port_num>
      <protocol>2</protocol>
      <status>2</status>
      <cur_conns>0</cur_conns>
      <tot_conns>0</tot_conns>
      <req_pkts>0</req_pkts>
      <resp_pkts>0</resp_pkts>
      <req_bytes>0</req_bytes>
      <resp_bytes>0</resp_bytes>
      <cur_reqs>0</cur_reqs>
      <total_req>0</total_req>
```

```

        <total_req_succ>0</total_req_succ>
    </port_stat>
    <port_stat>
        <port_num>80</port_num>
        <protocol>2</protocol>
        <status>2</status>
        <cur_conns>0</cur_conns>
        <tot_conns>0</tot_conns>
        <req_pkts>0</req_pkts>
        <resp_pkts>0</resp_pkts>
        <req_bytes>0</req_bytes>
        <resp_bytes>0</resp_bytes>
        <cur_reqs>0</cur_reqs>
        <total_req>0</total_req>
        <total_req_succ>0</total_req_succ>
    </port_stat>
</port_stat_list>
</server_stat>
</response>

```

2. HTTP Request in JSON Format:

session_id=123456789&method=slb.server.fetchStatistics&name=server01&format=json

Response as the HTTP body (JSON Format):

```
{
  "server_stat": [
    {
      "name": "server01",
      "host": "124.254.1.133",
      "status": 2,
      "cur_conns": 0,
      "tot_conns": 0,
      "req_pkts": 0,
      "resp_pkts": 0,
      "req_bytes": 0,
      "resp_bytes": 0,
      "cur_reqs": 0,
      "total_reqs": 0,
      "total_reqs_succ": 0,
      "port_stat_list": [
        {
          "port_num": 80,
          "protocol": 2,
          "status": 2,
          "cur_conns": 0,
          "tot_conns": 0,
          "req_pkts": 0,
          "resp_pkts": 0,
          "req_bytes": 0,
          "resp_bytes": 0,
          "cur_reqs": 0,
          "total_reqs": 0,
          "total_reqs_succ": 0
        },
        {
          "port_num": 21,
          "protocol": 2,
          "status": 2,
          "cur_conns": 0,
          "tot_conns": 0,
          "req_pkts": 0,
          "resp_pkts": 0,
          "req_bytes": 0,
          "resp_bytes": 0,
          "cur_reqs": 0,
          "total_reqs": 0,
          "total_reqs_succ": 0
        }
      ]
    }
  ]
}
```

```

        "req_pkts":0,
        "resp_pkts":0,
        "req_bytes":0,
        "resp_bytes":0,
        "cur_reqs":0,
        "total_reqs":0,
        "total_reqs_succ":0
    }
]
}
}

```

Example2: Fetch the serverstatistics by given hostname.

```
session_id=123456789&method=slb.real_server.fetchStatistics&host=sj-http-in01
```

Example3: Fetch the serverstatistics by given IPv6 address.

```
session_id=123456789&method=slb.real_server.fetchStatistics&host=sj-3ff1::2222
```

Example4: Fetch the serverstatistics by given ipv4 address.

```
session_id=123456789&method=slb.real_server.fetchStatistics&host=10.0.0.126
```

Menus Privilege

Monitor Mode >> Service >> SLB >> Server

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.2.5 “slb.server.create” Method

This method is used to create a new server and ports.

This method is required to use the HTTP POST method.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.server.create	String		
name (*)	server name	String		
host (*)	server identifier, ip address or dns name			
gslb_external_address	server GSLB external IP address			
status	server status, either enabled(1) or disabled(0)	Int	1 or 0	1
health_monitor	server health monitor method name	String		ping
weight	server weight	Int	1 or 0	1
conn_limit	server connection limit	Int	1 to 8000000	8000000
conn_limit_log	server connection limit logging	Int	1 or 0	0
conn_resume	server connection resume	Int	0 to 1000000	0
slow_start	server slow start option	Int	0 to 1	0
spoofing_cache	server spoofing cache option	Int	0 to 1	0
stats_data	server stat data option	Int	0 to 1	0
extended_stats	server extended stats option	Int	0 to 1	0
template	server template name	String		default
port_list	port list to be added: port1^Bport2^B ... portN ^B: ASCII Code 0x02, URL-encode %02	String	1 to 100	1

port<n>	port at element <n>				
port_num: (*)	Port number	Int	0-65534		
protocol: (*)	Protocol type, either TCP(2) or UDP(3)	Int	2 to 3		
status:	Port status, either enabled(1) or disabled(0)	Int	1 or 0	1	
conn_limit:	Connection limit of the virtual service	Int	1 to	8000000	
conn_limit_log:	Connection limit logging	Int	8000000	0	
conn_resume:	Connection resume	Int	1 or 0	0	
weight:	Port weight	Int	0 to	1	
no_ssl:	No ssl option	Int	1000000	0	
stats_data:	Stat dada option	Int	1 to 100	0	
extended_stat:	Extended stats option	Int	1 or 0	0	
health_monitor	Health monitor method name	String	1 or 0	0	
follow_port	Follow port	String	1 or 0	0	
 follow_port_num	Follow port number	Int	0 – 65534		
 follow_type	Follow port type, TCP(2) or UDP (3)	int	2 - 3		
template	Server port template name	String			
Note:	In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03				

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V1/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
!
!
```

1. HTTP Request in URL Format:

```
session_id=123456789&method=slb.server.create&server=name%03test02%02host%039.5.5.9%02status%031%02health_monitor%03(default)%02weight%031%02conn_limit%03100000%02conn_resume%03200%02template%03default%02slow_start%031%02spoofing_cache%031%02stats_data%031%02extended_stats%031%02port_list&port_list=p1%02p2&p1=port_num%038080%02protocol%032%02stat_us%031%02weight%03100%02no_ssl%031%02conn_limit%037000000%02conn_limit_log%031%02template%03default%02health_monitor%03ping%02stats_data%031%02extended_stats%031&p2=port_num%03800%02protocol%033%02status%030%02weight%0350%02no_ssl%031%02conn_limit%037000000%02conn_limit_log%031%02template%03default%02follow_port&follow_port=follow_port_num%038080%02follow_port_type%032&format=url
```

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in JSON Format:

```
session_id=123456789&method=slb.server.create&format=json
```

Request POST data (JSON Format)::

```
{  
    "server":  
    {  
        "name":"test02",  
        "host":"9.5.5.9",  
        "gslb_external_address":"0.0.0.0",  
        "weight":1,  
        "health_monitor":"(default)",  
        "status":1,  
        "conn_limit":1000000,  
        "conn_limit_log":0,  
        "conn_resume":200,  
        "stats_data":1,  
        "extended_stats":1,  
        "slow_start":1,  
        "spoofing_cache":1,  
        "template":"default",  
        "port_list": [  
            {  
                "port_num":8000,  
                "protocol":3,  
                "status":1,  
                "weight":50,  
                "no_ssl":1,  
                "conn_limit":7000000,  
                "conn_limit_log":1,  
                "conn_resume":0,  
                "template":"default",  
                "stats_data":1,  
                "follow_port":  
                    {  
                        "follow_port_num":8080,  
                        "follow_port_type":2  
                    },  
                "extended_stats":0  
            },  
            {  
                "port_num":8080,  
                "protocol":2,  
                "status":1,  
                "weight":100,  
                "no_ssl":1,  
                "conn_limit":7000000,  
                "conn_limit_log":1,  
                "conn_resume":0,  
                "template":"default",  
                "stats_data":1,  
                "health_monitor":"ping",  
                "extended_stats":1  
            }  
        ]  
    }  
}
```

CLI configuration after aXAPI call:

```
!  
slb server test02 9.5.5.9  
    conn-limit 1000000 no-logging  
    conn-resume 200  
    slow-start  
    spoofing-cache  
    extended-stats
```

```

port 8080  tcp
no-ssl
health-check ping
weight 100
conn-limit 7000000
extended-stats
port 8000  udp
no-ssl
health-check follow-port 8080  tcp
weight 50
conn-limit 7000000
!

```

Response as the HTTP body (JSON Format):

```
{
  "response": {
    "status" : "OK"
  }
}
```

Menus Privilege

Config Mode >> Service >> SLB >> Server

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.2.6 “slb.server.update” Method

This method is used to update parameters for an SLB server, such as ports, server weight, or connection limiting.

This method is required to use the HTTP POST method.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.server.update	String		
name (*)	Server name	String		
host (*)	Server identifier, ip address or dns name			
gslb_external_address	Server GSLB external IP address			
status	Server status, either enabled(1) or disabled(0)	Int	1 or 0	1
health_monitor	Server health monitor method name	String		ping
weight	Server weight	Int	1 or 0	1
conn_limit	Server connection limit	Int	1 to 1000000	1000000
conn_limit_log	Server connection limit logging	Int	1 or 0	0
conn_resume	Server connection resume	Int	0 to 1000000	0
slow_start	Server slow start option	Int	0 to 1	0
spoofing_cache	Server spoofing cache option	Int	0 to 1	0
stats_data	Server stat data option	Int	0 to 1	0
extended_stats	Server extended stats option	Int	0 to 1	0
template	Server template name	String		default
port_list	Port list to be added: port1^Bport2^B ... portN ^B: ASCII Code 0x02, URL-encode %02	String	1 to 100	1
port<n> port_num: (*) protocol: (*)	Port at element <n> Port number Protocol type, either TCP(2) or UDP(3)	Int Int	0-65534 2 to 3	

status:	Port status, either enabled(1) or disabled(0)	Int	1 or 0	1
conn_limit:	Connection limit of the virtual service	Int	1 to 1000000	1000000
conn_limit_log:	Connection limit logging	Int	0	0
conn_resume:	Connection resume	Int	1 or 0	0
weight:	Port weight	Int	0 to 1	1
no_ssl:	No ssl option	Int	1000000	0
stats_data:	Stat dada option	Int	1 to 100	0
extended_stat:	Extended stats option	Int	1 or 0	0
health_monitor	Health monitor method name	String	1 or 0	0
follow_port	Follow port	Int	1 or 0	0
 follow_port_num	Follow port number	int	0 - 65534	
 follow_type	Follow port type, TCP(2) or UDP (3)	String	2 - 3	
template	Server port template name			
Note:				
In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option.				
^B: ASCII Code 0x02, URL-encode %02				
^C: ASCII Code 0x03, URL-encode %03				

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- **URL:**
https://_AX_IP_Address_/services/rest/V2/
- **HTTP Action:**
POST

CLI configuration before aXAPI call:

```
!
slb server test02 9.5.5.9
  port 8080  tcp
    no-ssl
    health-check ping
!
```

1. HTTP Request in URL Format:

```
session_id=123456789&method=slb.server.update&server=name%03test02%02host%039.5.5.9%02status%03%02health_monitor%03(default)%02weight%031%02conn_limit%031000000%02conn_resume%03200%02template%03default%02slow_start%031%02spoofing_cache%031%02stats_data%031%02extended_stats%031%02port_list&port_list=p1%02p2&p1=port_num%038080%02protocol%032%02status%031%02weight%03100%02no_ssl%031%02conn_limit%037000000%02conn_limit_log%031%02template%03default%02health_monitor%03ping%02stats_data%031%02extended_stats%031&p2=port_num%03800%02protocol%033%02status%030%02weight%0350%02no_ssl%031%02conn_limit%037000000%02conn_limit_log%031%02template%03default%02follow_port&follow_port=follow_port_num%038080%02follow_port_type%032&format=url
```

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in JSON Format:

```
session_id=123456789&method=slb.server.update&format=json
```

Request POST data (JSON Format)::

```
{
  "server": [
    {
      "name": "test02",
      "host": "9.5.5.9",
      "gslb_external_address": "0.0.0.0",
      "weight": 1,
      "health_monitor": "(default)",
      "status": 1,
      "conn_limit": 1000000,
      "conn_limit_log": 0,
      "conn_resume": 200,
      "stats_data": 1,
      "extended_stats": 1,
      "slow_start": 1,
      "spoofing_cache": 1,
      "template": "default",
      "port_list": [
        {
          "port_num": 8000,
          "protocol": 3,
          "status": 1,
          "weight": 50,
          "no_ssl": 1,
          "conn_limit": 7000000,
          "conn_limit_log": 1,
          "conn_resume": 0,
          "template": "default",
          "stats_data": 1,
          "follow_port": [
            {
              "follow_port_num": 8080,
              "follow_port_type": 2
            }
          ],
          "extended_stats": 0
        },
        {
          "port_num": 8080,
          "protocol": 2,
          "status": 1,
          "weight": 100,
          "no_ssl": 1,
          "conn_limit": 7000000,
          "conn_limit_log": 1,
          "conn_resume": 0,
          "template": "default",
          "stats_data": 1,
          "health_monitor": "ping",
          "extended_stats": 1
        }
      ]
    }
  ]
}
```

Response as the HTTP body (JSON Format):

```
{
  "response": [
    {
      "status": "OK"
    }
  ]
}
```

CLI configuration after aXAPI call:

```
!
slb server test02 9.5.5.9
  conn-limit 1000000 no-logging
  conn-resume 200
  slow-start
  spoofing-cache
  extended-stats
  port 8080  tcp
    no-ssl
    health-check ping
    weight 100
    conn-limit 7000000
    extended-stats
  port 8000  udp
    no-ssl
    health-check follow-port 8080  tcp
    weight 50
    conn-limit 7000000
!
```

Menus Privilege

Config Mode >> Service >> SLB >> Server

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.2.7 “slb.server.delete” Method

This method is used to delete an SLB server and ports by specifying the particular server id, server name, server ip address, or dns name for the server.

This method is required in order to use the HTTP POST method.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.server.delete	String		
name (**)	server name	String		
host (**)	ip address or dns name	String		

Note: (*) parameter is required in the API.

(**) Either one of parameters is required.

Response Fields

General XML status response

Example Response

Example1: Delete “server2” by the server name.

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
slb server server2 9.5.5.9
  port 8080  tcp
    no-ssl
```

```
health-check ping
```

```
!
```

1. HTTP Request in URL Format:

```
session_id=123456789&method=slb.server.delete&name=server2&format=url
```

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in JSON Format:

```
session_id=123456789&method=slb.server.delete&format=json
```

Request POST data (JSON Format)::

```
{  
    "name": "test02"  
}
```

Response as the HTTP body (JSON Format):

```
{  
    "response":  
    {  
        "status": "OK"  
    }  
}
```

CLI configuration after aXAPI call:

```
!
```

```
!
```

Menus Privilege

Config Mode >> Service >> SLB >> Server

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.2.8 “slb.server.port.create” Method

This method is used to create a new port for a given server.

This method is required to use the HTTP POST method.

Parameters

Parameter Name	Description	Data Type	Range	Default
----------------	-------------	-----------	-------	---------

session_id (*)	user authenticated session id.	String		
method (*)	slb.server.port.create	String		
name (**)	server name	String		
host (**)	server identifier, ip address or dns name			
port port_num: (*) protocol: (*) status: conn_limit: conn_limit_log: conn_resume: weight: no_ssl: stats_data: extended_stat: health_monitor follow_port follow_port_num follow_type template	port object Port number Protocol type, either TCP(2) or UDP(3) Port status, either enabled(1) or disabled(0) Connection limit of the virtual service Connection limit logging Connection resume Port weight No ssl option Stat dada option Extended stats option Health monitor method name Follow port Follow port number Follow port type, TCP(2) or UDP (3) Server port template name	Int Int Int Int Int Int Int Int Int Int String Int int String	1 to 65535 2 to 3 1 or 0 1 to 1000000 1 or 0 0 to 1000000 1 to 100 1 or 0 1 or 0 1 or 0 0 - 65534 2 - 3	1 1000000 0 0 0 1 0 0 0 0

Note: (*) parameter is required in the API.

(**) Either one of parameters is required.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
slb server test04 2067:2345:5678:6::9001
  port 151  tcp
!
```

1. HTTP Request in URL Format:

```
session_id=123456789&method=slb.server.port.create&name=test04&port=port_num%03202
%02protocol%032%02template%03portt&format=url
```

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in JSON Format:

```
session_id=123456789&method=slb.server.port.create&name=test04&format=json
```

Request POST data (JSON Format):

```
{
  "name": "test04",
  "port":
```

```

        "port_num":202,
        "protocol":2,
        "template":"portt"
    }
}

```

Response as the HTTP body (JSON Format):

```

{
    "response":
    {
        "status" : "OK"
    }
}

```

CLI configuration after aXAPI call:

```

!
slb server test04 2067:2345:5678:6::9001
    port 151  tcp
    port 202  tcp
        template port portt
!

```

Menus Privilege

Config Mode >> Service >> SLB >> Server

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.2.9 “slb.server.port.update” Method

This method is used to update parameters for a server port, such as port status or connection limiting.

This method is required to use the HTTP POST method.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.server.port.update	String		
name (**)	server name	String		
host (**)	server identifier, ip address or dns name			
port				
port_num: (*)	Port number	Int	1 to 65535	
protocol: (*)	Protocol type, either TCP(2) or UDP(3)	Int	2 to 3	
status:	Port status, either enabled(1) or disabled(0)	Int	1 or 0	1
conn_limit:	Connection limit of the virtual service	Int	1 to	1000000
conn_limit_log:	Connection limit logging	Int	1000000	0
conn_resume:	Connection resume	Int	1 or 0	0
weight:	Port weight	Int	0 to	1
no_ssl:	No ssl option	Int	1000000	0
stats_data:	Stat data option	Int	1 to 100	0
extended_stat:	Extended stats option	Int	1 or 0	0
health_monitor	Health monitor method name	String	1 or 0	
follow_port	Follow port		1 or 0	
follow_port_num	Follow port number	Int		
follow_type	Follow port type, TCP(2) or UDP (3)	int		
	Server port template name	String	0 – 65534	

template	Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03	2 - 3	
-----------------	--	-------	--

Note: (*) parameter is required in the API.

(**) Either one of parameters is required.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
slb server test04 2067:2345:5678:6::9001
  port 151  tcp
  port 202  tcp
    template port portt
!
```

1. HTTP Request in URL Format:

```
session_id=123456789&method=slb.server.port.update&name=test04&port=port_num%03202%02proto
ocol%032%02template%03portt&format=url
```

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in JSON Format:

```
session_id=123456789&method=slb.server.port.update&name=test04&format=json
```

Request POST data (JSON Format):

```
{
  "name": "test04",
  "port": {
    "port_num": 202,
    "protocol": 2,
    "template": "default"
  }
}
```

CLI configuration after aXAPI call:

```
!
slb server test04 2067:2345:5678:6::9001
  port 151  tcp
  port 202  tcp
!
```

Menus Privilege

Config Mode >> Service >> SLB >> Server

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.2.10 “slb.server.port.delete” Method

This method is used to delete the port on a specific server by specifying the server name, server id, server name, server IP address, or DNS name for the server, and the port number and protocol type.

This method is required to use the HTTP POST method.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.server.port.delete	String		
name (**)	server name	String		
host (**)	server identifier, ip address or dns name			
port port_num: (*) protocol: (*)	port object Port number Protocol type, either TCP(2) or UDP(3)	Int Int	0 to 65534 2 to 3	

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- **URL:**
https://_AX_IP_Address_/services/rest/V2/
- **HTTP Action:**
POST
- **HTTP Body**

Example1:

CLI configuration before aXAPI call:

```
!
slb server test04 2067:2345:5678:6::9001
  port 151  tcp
  port 202  tcp
    template port portt
!
```

1. **HTTP Request in URL Format:**

```
session_id=123456789&method=slb.server.port.delete&name=test04&port=port_num%03202
%02protocol%032&format=url
```

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in JSON Format:

```
session_id=123456789&method=slb.server.port.delete&format=json
```

Request POST data (JSON Format):

```
{  
    "name": "test04",  
    "port":  
    {  
        "port_num": 202,  
        "protocol": 2,  
    }  
}
```

Response as the HTTP body (JSON Format):

```
{  
    "response":  
    {  
        "status": "OK"  
    }  
}
```

CLI configuration after aXAPI call:

```
!  
slb server test04 2067:2345:5678:6::9001  
    port 151  tcp  
!
```

Example 2: session_id=123456789&method=slb.server.port.delete

Request POST data (JSON Format):

```
{  
    "host": "3ff6::6745",  
    "port":  
    {  
        "port_num": 202,  
        "protocol": 2,  
    }  
}
```

Example 3: session_id=123456789&method=slb.server.port.delete

Request POST data (JSON Format):

```
{  
    "host": "dns_sj_http01",  
    "port":  
    {  
        "port_num": 202,  
        "protocol": 2,  
    }  
}
```

Menus Privilege

Config Mode >> Service >> SLB >> Server

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.3 Virtual Server

7.3.1 “slb.virtual_server.getAll” Method

This method is used to get configuration information for all virtual servers with information about the virtual services configured for each virtual server.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.virtual_server.getAll	String		

Note: (*) parameter is required in the API.

Response Fields

The response XML is the collection of virtual servers, and each virtual server will have a collection of the virtual services.

virtual_server_list	XML tag for the collection of virtual servers
virtual_server	XML tag for the virtual server
name	virtual server name.
[]	One of the following token (address, subnet, acl_id, acl_name):
address	virtual server address, either IPv4 or IPv6
subnet	IPv4 subnet
address	address of subnet
mask_len	the length of this subnet
acl_id	IPv4 ACL id
acl_name	IPv6 ACL name
status	virtual server status, either enabled(1) or disabled(0)
arp_status	ARP disabled option, either disabled(0) or enabled(1)
stata_data	stats data option, either disabled(0) or enabled(1)
extended_stat	extended stat option, either disabled(0) or enabled(1)
disable_vserver_on_condition	disable vserver on conditions. Do not disable virtual server in any case(0), disable virtual server when any port down (1), disable virtual server when all ports down(2)
redistribution_flagged	redistribution flagged option, either disabled(0) or enabled(1)
ha_group	configured HA group
group	status of HA config
ha_group_id	ha group id
dynamic_server_weight	dynamic server weight
vip_template	virtual server template name
pbslb_template	pbslb policy template name

vport_list	XML tag for virtual services
vport	XML tag for a virtual service, and contains the following properties:
protocol	virtual service protocol with: TCP 2 UDP 3 Other 4 RTSP 5 FTP 6 MMS 7 SIP 8 FAST-HTTP 9 TCP-PROXY 10 HTTP 11 HTTPS 12 SSL-PROXY 13 SMTP 14 SIP-TCP 15 SIP-STL 16 DIAMETER 17 DNS-UDP 18 TFTP 19
port	virtual service port number
service_group	service group assigned to the virtual service
status	virtual service status either enabled(1) or disabled(0)
connection_limit	connection limit of the virtual service
 status	connection limit status of the virtual service either enabled(1) or disabled(0)
 connection_limit	connection limit value
 connection_limit_action	connection limit action, either reset(1) or drop(0)
 connection_limit_log	connection limit logging, either enabled(1) or disabled(0)
default_selection	use default server selection when preferred method fails, either enabled(1) or disabled(0)
received_hop	use received hop for response either enabled(1) or disabled(0)
stats_data	stats data, either enabled(1) or disabled(0)
extended_stats	extended stats, either enabled(1) or disabled(0)
snat_against_vip	source nat traffic against vip either enabled(1) or disabled(0)
vport_template	virtual server port template name
port_acl_id	assigned access-list, ipv4 only
pbslb_template	pbslb template
aflex_list	aflex list
 aflex	name of aflex
acl_natpool_binding_list	acl – natpool binding list
 acl_natpool_binding	acl – natpool binding
 []	one of the following properties:
 Acl_id	ipv4 acl id
 Acl name	IPv6 acl name
 nat_pool	source nat pool name

http vport has the following additional properties:

send_reset	send client reset when server selection fails, either enabled(1) or disabled(0)
sync_cookie	syn cookie, either enabled(1) or disabled(0)
source_nat	source NAT pool name
http_template	HTTP template name
ram_cache_template	RAM caching template name

tcp_proxy_template	TCP-Proxy template name
server_ssl_template	Server-SSL template name
conn_reuse_template	Connection-Reuse template name
[]	one of the following properties:
source_ip_persistence_template	Source-IP-Persistent template name
cookie_persistence_template	cookie persistence template name

https vport has the following additional properties:

send_reset	send client reset when server selection fails, either enabled(1) or disabled(0)
sync_cookie	syn cookie, either enabled(1) or disabled(0)
source_nat	source NAT pool name
http_template	HTTP template name
ram_cache_template	RAM caching template name
tcp_proxy_template	TCP-Proxy template name
client_ssl_template	client ssl template
server_ssl_template	Server-SSL template name
conn_reuse_template	Connection-Reuse template name
[]	one of the following properties:
source_ip_persistence_template	Source-IP-Persistent template name
cookie_persistence_template	cookie persistence template name

fast-http vport has the following additional properties:

send_reset	send client reset when server selection fails, either enabled(1) or disabled(0)
sync_cookie	syn cookie, either enabled(1) or disabled(0)
source_nat	source NAT pool name
http_template	HTTP template name
tcp_template	TCP template name
conn_reuse_template	Connection-Reuse template name
[]	one of the following properties:
source_ip_persistence_template	Source-IP-Persistent template name
cookie_persistence_template	cookie persistence template name

tcp vport has the following additional properties:

send_reset	send client reset when server selection fails, either enabled(1) or disabled(0)
ha_connection_mirror	HA connection mirror enabled(1) or disabled (0)
direct_server_return	direct server return (no destination NAT) enabled (1) or disabled (0)
sync_cookie	syn cookie, either enabled(1) or disabled(0)
source_nat	source NAT pool name

tcp_template	TCP template name
[]	one of the following properties:
source_ip_persistence_template	Source-IP-Persistent template name
cookie_persistence_template	cookie persistence template name

udp vport has the following additional properties:

ha_connection_mirror	HA connection mirror enabled(1) or disabled (0)
direct_server_return	direct server return (no destination NAT) enabled (1) or disabled (0)
source_nat	source NAT pool name
udp_template	UDP template name
dns_template	UDP template name
source_ip_persistence_template	Source-IP-Persistent template name

rtsp vport has the following additional properties:

send_reset	send client reset when server selection fails, either enabled(1) or disabled(0)
ha_connection_mirror	HA connection mirror enabled(1) or disabled (0)
direct_server_return	direct server return (no destination NAT) enabled (1) or disabled (0)
sync_cookie	syn cookie, either enabled(1) or disabled(0)
tcp_template	TCP template name
rtsp_template	RTSP template name
source_ip_persistence_template	Source-IP-Persistent template name

ftp vport has the following additional properties:

send_reset	send client reset when server selection fails, either enabled(1) or disabled(0)
ha_connection_mirror	HA connection mirror enabled(1) or disabled (0)
direct_server_return	direct server return (no destination NAT) enabled (1) or disabled (0)
sync_cookie	syn cookie, either enabled(1) or disabled(0)
source_nat	source NAT pool name
tcp_template	TCP template name
source_ip_persistence_template	Source-IP-Persistent template name

mms vport has the following additional properties:

send_reset	send client reset when server selection fails, either enabled(1) or disabled(0)
ha_connection_mirror	HA connection mirror enabled(1) or disabled (0)

sync_cookie	syn cookie, either enabled(1) or disabled(0)
tcp_template	TCP template name
source_ip_persistence_template	Source-IP-Persistent template name

ssl proxy vport has the following additional properties:

send_reset	send client reset when server selection fails, either enabled(1) or disabled(0)
sync_cookie	syn cookie, either enabled(1) or disabled(0)
source_nat	source NAT pool name
tcp_proxy_template	TCP-Proxy template name
client_ssl_template	client ssl template
server_ssl_template	Server-SSL template name
conn_reuse_template	Connection-Reuse template name
source_ip_persistence_template	Source-IP-Persistent template name

smtp proxy vport has the following additional properties:

send_reset	send client reset when server selection fails, either enabled(1) or disabled(0)
sync_cookie	syn cookie, either enabled(1) or disabled(0)
source_nat	source NAT pool name
tcp_proxy_template	TCP-Proxy template name
client_ssl_template	client ssl template
smtp_template	Server-SSL template name
source_ip_persistence_template	Source-IP-Persistent template name

sip vport has the following additional properties:

send_reset	send client reset when server selection fails, either enabled(1) or disabled(0)
ha_connection_mirror	HA connection mirror enabled(1) or disabled (0)
udp_template	UDP template name
source_ip_persistence_template	Source-IP-Persistent template name
sip_template	UDP template name
dns_template	SIP template name

sip-tcp vport has the following additional properties:

send_reset	send client reset when server selection fails, either enabled(1) or disabled(0)
sync_cookie	syn cookie, either enabled(1) or disabled(0)
source_nat	source NAT pool name

server_ssl_template	Server-SSL template name
conn_reuse_template	Connection-Reuse template name
tcp_proxy_template	TCP-Proxy template name
source_ip_persistence_template	Source-IP-Persistent template name
sip_template	SIP template name

sip-tls vport has the following additional properties:

send_reset	send client reset when server selection fails, either enabled(1) or disabled(0)
sync_cookie	syn cookie, either enabled(1) or disabled(0)
source_nat	source NAT pool name
client_ssl_template	client ssl template
server_ssl_template	Server-SSL template name
conn_reuse_template	Connection-Reuse template name
tcp_proxy_template	TCP-Proxy template name
source_ip_persistence_template	Source-IP-Persistent template name
sip_template	SIP template name

tcp-proxy vport has the following additional properties:

send_reset	send client reset when server selection fails, either enabled(1) or disabled(0)
sync_cookie	syn cookie, either enabled(1) or disabled(0)
source_nat	source NAT pool name
tcp_proxy_template	TCP-Proxy template name
source_ip_persistence_template	Source-IP-Persistent template name

dns-udp vport has the following additional properties:

send_reset	send client reset when server selection fails, either enabled(1) or disabled(0)
source_nat	source NAT pool name\
udp_template	UDP template name
source_ip_persistence_template	Source-IP-Persistent template name
dns_template	UDP template name

diameter-udp vport has the following additional properties:

send_reset	send client reset when server selection fails, either enabled(1) or disabled(0)
source_nat	source NAT pool name
udp_template	UDP template name
source_ip_persistence_template	Source-IP-Persistent template name

dns_template	UDP template name
---------------------	-------------------

tftp vport has the following additional properties:

ha_connection_mirror	HA connection mirror enabled(1) or disabled (0)
direct_server_return	direct server return (no destination NAT) enabled (1)
source_nat	source NAT pool name
udp_template	UDP template name

'other' vport has the following additional properties:

direct_server_return	direct server return (no destination NAT) enabled (1)
source_nat	source NAT pool name
udp_template	UDP template name
l4_template_type	TCP (2) or UDP(3)
[]	one of the following properties:
udp_template	UDP template name
tcp_template	TCP template name

Example Response

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
slb virtual-server AAA 8.2.3.18
  disable when-all-ports-down
  redistribution-flagged
  template policy 111
  ha-group 1
  ha-dynamic 11
  extended-stats
  port 80 http
    name _8.2.3.18_HTTP_80
    access-list 1 source-nat-pool test
    service-group http
    syn-cookie sack
    template tcp-proxy tcppt
  port 389 sip-tcp
    name _8.2.3.18_SIP-TCP_389
    service-group http
  port 8088 fast-http
    name _8.2.3.18_Fast-HTTP_8088
    syn-cookie sack
!
```

1. HTTP Request in URL Format:

session_id=123456789&method=slb.virtual_server.getAll&format=url

Response as the HTTP body (URL Format):

```
<response status="ok">
<virtual_server_list>
```

```

<virtual_server>
  <name>AAA</name>
  <address>8.2.3.18</address>
  <status>1</status>
  <arp_status>1</arp_status>
  <stats_data>1</stats_data>
  <extended_stats>1</extended_stats>
  <disable_vserver_on_condition>0</ disable_vserver_on_condition >
  <redistribution_flagged>1</redistribution_flagged>
  <ha_group>
    <status>1</status>
    <ha_group_id>1</ha_group_id>
    <dynamic_server_weight>11</dynamic_server_weight>
  </ha_group>
  <vip_template>default</vip_template>
  <pbslb_template>111</pbslb_template>
  <vport_list>
    <vport>
      <protocol>15</protocol>
      <port>389</port>
      <service_group>http</service_group>
      <connection_limit>
        <status>0</status>
        <connection_limit>8000000</connection_limit>
        <connection_limit_action>0</connection_limit_action>
        <connection_limit_log>1</connection_limit_log>
      </connection_limit>
      <default_selection>1</default_selection>
      <received_hop>0</received_hop>
      <status>1</status>
      <stats_data>1</stats_data>
      <extended_stats>0</extended_stats>
      <snat_against_vip>0</snat_against_vip>
      <vport_template/>
      <port_acl_id>0</port_acl_id>
      <aflex_list/>
      <send_reset>0</send_reset>
      <sync_cookie>0</sync_cookie>
      <source_nat/>
      <server_ssl_template/>
      <conn_reuse_template/>
      <tcp_proxy_template/>
      <source_ip_persistence_template/>
      <sip_template/>
      <pbslb_template/>
      <acl_natpool_binding_list/>
    </vport>
    <vport>
      <protocol>11</protocol>
      <port>80</port>
      <service_group>http</service_group>
      <connection_limit>
        <status>0</status>
        <connection_limit>8000000</connection_limit>
        <connection_limit_action>0</connection_limit_action>
        <connection_limit_log>1</connection_limit_log>
      </connection_limit>
      <default_selection>1</default_selection>
      <received_hop>0</received_hop>
      <status>1</status>
      <stats_data>1</stats_data>
      <extended_stats>0</extended_stats>
      <snat_against_vip>0</snat_against_vip>
      <vport_template/>
      <acl_id>0</acl_id>
      <aflex_list/>
    </vport>
  </vport_list>
</virtual_server>

```

```

<send_reset>0</send_reset>
<sync_cookie>0</sync_cookie>
<source_nat/>
<http_template/>
<ram_cache_template/>
<tcp_proxy_template>tcppt</tcp_proxy_template>
<server_ssl_template/>
<conn_reuse_template/>
<source_ip_persistence_template/>
<pbslb_template/>
<acl_natpool_binding_list>
  <acl_natpool_binding>
    <port_acl_id>1</port_acl_id>
    <nat_pool>test</nat_pool>
  </acl_natpool_binding>
</acl_natpool_binding_list>
</vport>
<vport>
  <protocol>3</protocol>
  <port>12347</port>
  <service_group>udp</service_group>
  <connection_limit>
    <status>0</status>
    <connection_limit>8000000</connection_limit>
    <connection_limit_action>0</connection_limit_action>
    <connection_limit_log>1</connection_limit_log>
  </connection_limit>
  <default_selection>0</default_selection>
  <received_hop>0</received_hop>
  <status>0</status>
  <stats_data>0</stats_data>
  <extended_stats>0</extended_stats>
  <snat_against_vip>0</snat_against_vip>
  <vport_template/>
  <acl_id>0</acl_id>
  <aflex_list>
    <aflex>DownloadFile1</aflex>
    <aflex>Test01</aflex>
  </aflex_list>
  <ha_connection_mirror>0</ha_connection_mirror>
  <direct_server_return>0</direct_server_return>
  <source_nat/>
  <udp_template/>
  <source_ip_persistence_template/>
  <dns_template/>
  <pbslb_template/>
  <acl_natpool_binding_list/>
</vport>
</vport_list>
</virtual_server>
</virtual_server_list>
</response>

```

2. HTTP Request in JSON Format:

session_id=123456789&method=slb.virtual_server.getAll&format=json

Request POST data (JSON Format):

```
{
  "virtual_server_list": [
    {
      "name": "VirtualServer1"
    }
  ]
}
```

```

"name":"AAA",
"address":"8.2.3.18",
"status":1,
"arp_status":1,
"stats_data":1,
"extended_stats":1,
"disable_vserver_on_condition":2,
"redistribution_flagged":1,
"ha_group":
{
    "status":1,
    "ha_group_id":1,
    "dynamic_server_weight":11
},
"vip_template":"default",
"pbslb_template":"111",
"vport_list":[
{
    "protocol":9,
    "port":8088,
    "service_group":"",
    "connection_limit":
    {
        "status":0,
        "connection_limit":8000000,
        "connection_limit_action":0,
        "connection_limit_log":0
    },
    "default_selection":1,
    "received_hop":0,
    "status":1,
    "stats_data":1,
    "extended_stats":0,
    "snat_against_vip":0,
    "vport_template":"default",
    "aflex_list":[],
    "send_reset":0,
    "sack":0,
    "source_nat":"",
    "http_template":"",
    "tcp_template":"",
    "conn_reuse_template":"",
    "source_ip_persistence_template":"",
    "pbslb_template":"",
    "acl_natpool_binding_list":[]
},
{
    "protocol":15,
    "port":389,
    "service_group":"http",
    "connection_limit":
    {
        "status":0,
        "connection_limit":8000000,
        "connection_limit_action":0,
        "connection_limit_log":0
    },
    "default_selection":1,
    "received_hop":0,
    "status":1,
    "stats_data":1,
    "extended_stats":0,
    "snat_against_vip":0,
    "vport_template":"default",
    "aflex_list":[],
    "send_reset":0,
}
]
}

```

```
"sync_cookie":  
{  
    "sync_cookie":0,  
    "sack":0  
},  
    "source_nat": "",  
    "server_ssl_template": "",  
    "conn_reuse_template": "",  
    "tcp_proxy_template": "",  
    "source_ip_persistence_template": "",  
    "sip_template": "",  
    "pbslb_template": "",  
    "acl_natpool_binding_list": []  
},  
{  
    "protocol":11,  
    "port":80,  
    "service_group":"http",  
    "connection_limit":  
    {  
        "status":0,  
        "connection_limit":8000000,  
        "connection_limit_action":0,  
        "connection_limit_log":0  
    },  
    "default_selection":1,  
    "received_hop":0,  
    "status":1,  
    "stats_data":1,  
    "extended_stats":0,  
    "snat_against_vip":0,  
    "vport_template":"default",  
    "aflex_list": [],  
    "send_reset":0,  
    "sync_cookie":  
    {  
        "sync_cookie":1,  
        "sack":0  
    },  
    "source_nat": "",  
    "http_template": "",  
    "ram_cache_template": "",  
    "tcp_proxy_template": "tcppt",  
    "server_ssl_template": "",  
    "conn_reuse_template": "",  
    "source_ip_persistence_template": "",  
    "pbslb_template": "",  
    "acl_natpool_binding_list": [  
    {  
        "acl_id":1,  
        "nat_pool":"test"  
    }  
]  
}  
}  
]  
}
```

Menus Privilege

Config Mode >> Service >> SLB >> Virtual Server

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.3.2 “slb.virtual_server.search” Method

This method is used to get virtual server configuration information by given virtual server name or IP address (IPv4 or IPv6).

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.virtual_server.search	String		
One of the following parameters (address, subnet, acl_id, acl_name)				
address	The address of this virtual server (fetch the virtual server by address, either IPv4 or IPv6)	IP		
subnet address mask_len	The subnet of this virtual server (fetch the virtual server by subnet address and mask length, only for IPv4)	IPv4 Int	1 – 32	
acl_id	The acl id of this virtual server (fetch the virtual server by acl id, for IPv4 wildcard virtual server only)	Int	1 – 199	
acl_name	The acl name of this virtual server (fetch the virtual server by acl name, for IPv6 wildcard virtual server only)	String		

Note: (*) parameter is required in the API.

(**) Either one of parameters is required.

Response Fields

virtual_server_list	XML tag for the collection of virtual servers
virtual_server	XML tag for the virtual server
name	virtual server name.
[]	One of the following token (address, subnet, acl_id, acl_name):
address	virtual server address, either IPv4 or IPv6
subnet	IPv4 subnet
address	address of subnet
mask_len	the length of this subnet
acl_id	IPv4 acl id
acl_name	IPv6 acl name
status	virtual server status, either enabled(1) or disabled(0)
arp_status	ARP disabled option, either disabled(0) or enabled(1)
stata_data	stats data option, either disabled(0) or enabled(1)
extended_stat	extended stat option, either disabled(0) or enabled(1)
disable_vserver_on_condition	disable vserver on conditions. Do not disable virtual server in any case(0), disable virtual server when any port down (1), disable virtual server when all ports down(2)
redistribution_flagged	redistribution flagged option, either disabled(0) or enabled(1)
ha_group	configured HA group

group	status of HA config
ha_group_id	ha group id
dynamic_server_weight	dynamic server weight
vip_template	virtual server template name
pbslb_template	pbslb policy template name
vport_list	XML tag for virtual services
vport	XML tag for a virtual service, and contains the following properties:
protocol	virtual service protocol with:
	TCP 2
	UDP 3
	Other 4
	RTSP 5
	FTP 6
	MMS 7
	SIP 8
	FAST-HTTP 9
	TCP-PROXY 10
	HTTP 11
	HTTPS 12
	SSL-PROXY 13
	SMTP 14
	SIP-TCP 15
	SIP-STL 16
	DIAMETER 17
	DNS-UDP 18
	TFTP 19
port	virtual service port number
service_group	service group assigned to the virtual service
status	virtual service status either enabled(1) or disabled(0)
connection_limit	connection limit of the virtual service
status	connection limit status of the virtual service either enabled(1) or disabled(0)
connection_limit	connection limit value
connection_limit_action	connection limit action, either reset(1) or drop(0)
connection_limit_log	connection limit logging, either enabled(1) or disabled(0)
default_selection	use default server selection when preferred method fails, either enabled(1) or disabled(0)
received_hop	use received hop for response either enabled(1) or disabled(0)
stats_data	stats data, either enabled(1) or disabled(0)
extended_stats	extended stats, either enabled(1) or disabled(0)
snat_against_vip	source nat traffic against vip either enabled(1) or disabled(0)
vport_template	virtual server port template name
port_acl_id	assigned access-list, IPv4 only
pbslb_template	pbslb template
aflex_list	aflex list
aflex	name of aflex
acl_natpool_binding_list	acl – natpool binding list
acl_natpool_binding	acl – natpool binding
[]	one of the following properties:
	Acl_id IPv4 acl id
	Acl name IPv6 acl name
nat_pool	source nat pool name

http vport has the following additional properties:

send_reset send client reset when server selection fails,

either enabled(1) or disabled(0)	
sync_cookie	syn cookie, either enabled(1) or disabled(0)
source_nat	source NAT pool name
http_template	HTTP template name
ram_cache_template	RAM caching template name
tcp_proxy_template	TCP-Proxy template name
server_ssl_template	Server-SSL template name
conn_reuse_template	Connection-Reuse template name
[] one of the following properties:	
source_ip_persistence_template	Source-IP-Persistent template name
cookie_persistence_template	cookie persistence template name

https vport has the following additional properties:

send_reset	send client reset when server selection fails, either enabled(1) or disabled(0)
sync_cookie	syn cookie, either enabled(1) or disabled(0)
source_nat	source NAT pool name
http_template	HTTP template name
ram_cache_template	RAM caching template name
tcp_proxy_template	TCP-Proxy template name
client_ssl_template	client ssl template
server_ssl_template	Server-SSL template name
conn_reuse_template	Connection-Reuse template name
[] one of the following properties:	
source_ip_persistence_template	Source-IP-Persistent template name
cookie_persistence_template	cookie persistence template name

fast-http vport has the following additional properties:

send_reset	send client reset when server selection fails, either enabled(1) or disabled(0)
sync_cookie	syn cookie, either enabled(1) or disabled(0)
source_nat	source NAT pool name
http_template	HTTP template name
tcp_template	TCP template name
conn_reuse_template	Connection-Reuse template name
[] one of the following properties:	
source_ip_persistence_template	Source-IP-Persistent template name
cookie_persistence_template	cookie persistence template name

tcp vport has the following additional properties:

send_reset	send client reset when server selection fails, either enabled(1) or disabled(0)
-------------------	--

ha_connection_mirror HA connection mirror enabled(1) or disabled (0)
direct_server_return direct server return (no destination NAT) enabled (1) or disabled
(0)

sync_cookie	syn cookie, either enabled(1) or disabled(0)
source_nat	source NAT pool name
tcp_template	TCP template name
[]	one of the following properties:
source_ip_persistence_template	Source-IP-Persistent template name
cookie_persistence_template	cookie persistence template name

udp vport has the following additional properties:

ha_connection_mirror HA connection mirror enabled(1) or disabled (0)
direct_server_return direct server return (no destination NAT) enabled (1) or disabled
(0)

source_nat	source NAT pool name
udp_template	UDP template name
dns_template	UDP template name
source_ip_persistence_template	Source-IP-Persistent template name

rtsp vport has the following additional properties:

send_reset send client reset when server selection fails,
either enabled(1) or disabled(0)
ha_connection_mirror HA connection mirror enabled(1) or disabled (0)
direct_server_return direct server return (no destination NAT) enabled (1) or disabled
(0)

sync_cookie	syn cookie, either enabled(1) or disabled(0)
tcp_template	TCP template name
rtsp_template	RTSP template name
source_ip_persistence_template	Source-IP-Persistent template name

ftp vport has the following additional properties:

send_reset send client reset when server selection fails,
either enabled(1) or disabled(0)
ha_connection_mirror HA connection mirror enabled(1) or disabled (0)
direct_server_return direct server return (no destination NAT) enabled (1) or disabled
(0)

sync_cookie	syn cookie, either enabled(1) or disabled(0)
source_nat	source NAT pool name
tcp_template	TCP template name
source_ip_persistence_template	Source-IP-Persistent template name

mms vport has the following additional properties:

send_reset	send client reset when server selection fails, either enabled(1) or disabled(0)
ha_connection_mirror	HA connection mirror enabled(1) or disabled (0)
sync_cookie	syn cookie, either enabled(1) or disabled(0)
tcp_template	TCP template name

source_ip_persistence_template Source-IP-Persistent template name

ssl proxy vport has the following additional properties:

send_reset	send client reset when server selection fails, either enabled(1) or disabled(0)
sync_cookie	syn cookie, either enabled(1) or disabled(0)
source_nat	source NAT pool name
tcp_proxy_template	TCP-Proxy template name
client_ssl_template	client ssl template
server_ssl_template	Server-SSL template name
conn_reuse_template	Connection-Reuse template name
source_ip_persistence_template	Source-IP-Persistent template name

smtp proxy vport has the following additional properties:

send_reset	send client reset when server selection fails, either enabled(1) or disabled(0)
sync_cookie	syn cookie, either enabled(1) or disabled(0)
source_nat	source NAT pool name
tcp_proxy_template	TCP-Proxy template name
client_ssl_template	client ssl template
smtp_template	Server-SSL template name
source_ip_persistence_template	Source-IP-Persistent template name

sip vport has the following additional properties:

send_reset	send client reset when server selection fails, either enabled(1) or disabled(0)
ha_connection_mirror	HA connection mirror enabled(1) or disabled (0)
udp_template	UDP template name
source_ip_persistence_template	Source-IP-Persistent template name
sip_template	UDP template name
dns_template	SIP template name

sip-tcp vport has the following additional properties:

send_reset	send client reset when server selection fails, either enabled(1) or disabled(0)
sync_cookie	syn cookie, either enabled(1) or disabled(0)
source_nat	source NAT pool name
server_ssl_template	Server-SSL template name
conn_reuse_template	Connection-Reuse template name
tcp_proxy_template	TCP-Proxy template name
source_ip_persistence_template	Source-IP-Persistent template name
sip_template	SIP template name

sip-tls vport has the following additional properties:

send_reset	send client reset when server selection fails, either enabled(1) or disabled(0)
sync_cookie	syn cookie, either enabled(1) or disabled(0)
source_nat	source NAT pool name
client_ssl_template	client ssl template
server_ssl_template	Server-SSL template name
conn_reuse_template	Connection-Reuse template name
tcp_proxy_template	TCP-Proxy template name
source_ip_persistence_template	Source-IP-Persistent template name
sip_template	SIP template name

tcp-proxy vport has the following additional properties:

send_reset	send client reset when server selection fails, either enabled(1) or disabled(0)
sync_cookie	syn cookie, either enabled(1) or disabled(0)
source_nat	source NAT pool name
tcp_proxy_template	TCP-Proxy template name
source_ip_persistence_template	Source-IP-Persistent template name

dns-udp vport has the following additional properties:

send_reset	send client reset when server selection fails, either enabled(1) or disabled(0)
source_nat	source NAT pool name\
udp_template	UDP template name
source_ip_persistence_template	Source-IP-Persistent template name
dns_template	UDP template name

diameter-udp vport has the following additional properties:

send_reset	send client reset when server selection fails, either enabled(1) or disabled(0)
source_nat	source NAT pool name
udp_template	UDP template name
source_ip_persistence_template	Source-IP-Persistent template name
dns_template	UDP template name

tftp vport has the following additional properties:

ha_connection_mirror	HA connection mirror enabled(1) or disabled (0)
direct_server_return	direct server return (no destination NAT) enabled (1)
source_nat	source NAT pool name
udp_template	UDP template name

‘other’ vport has the following additional properties:

direct_server_return	direct server return (no destination NAT) enabled (1)
source_nat	source NAT pool name
udp_template	UDP template name
l4_template_type	TCP (2) or UDP(3)
[]	one of the following properties: udp_template UDP template name tcp_template TCP template name

Example Response

Example1: search the virtual server by the name.

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
slb virtual-server http-vip1 8.2.3.18
  redistribution-flagged
  template policy 111
  ha-group 1
  ha-dynamic 11
  extended-stats
  port 389 diameter
    name _8.2.3.18_Diameter_389
    service-group http
    syn-cookie sack
  port 80 ssl-proxy
    name _8.2.3.18_SSL_Proxy_80
    service-group http
  port 12347 udp
    name _8.2.3.18_UDP_12347
    service-group udp
    aflex DownloadFile1
    aflex Test01
!
```

1. HTTP Request in URL Format:

```
session_id=123456789&method=slb.virtual_server.search&name=http_vip1&format=url
```

Response as the HTTP body (URL Format):

```
<response status="ok">
  <virtual_server>
    <name>http_vip1</name>
    <address>8.2.3.18</address>
    <status>1</status>
    <arp_status>1</arp_status>
    <stats_data>1</stats_data>
    <extended_stats>1</extended_stats>
    <disable_vserver_on_condition>0</disable_vserver_on_condition>
    <redistribution_flagged>1</redistribution_flagged>
    <ha_group>
      <status>1</status>
      <ha_group_id>1</ha_group_id>
      <dynamic_server_weight>11</dynamic_server_weight>
    </ha_group>
    <vip_template>default</vip_template>
    <pbslb_template>111</pbslb_template>
    <vport_list>
      <vport>
        <protocol>15</protocol>
        <port>389</port>
        <service_group>http</service_group>
        <connection_limit>
          <status>0</status>
          <connection_limit>8000000</connection_limit>
          <connection_limit_action>0</connection_limit_action>
          <connection_limit_log>1</connection_limit_log>
        </connection_limit>
        <default_selection>1</default_selection>
        <received_hop>0</received_hop>
        <status>1</status>
        <stats_data>1</stats_data>
        <extended_stats>0</extended_stats>
        <snat_against_vip>0</snat_against_vip>
        <vport_template/>
        <port_acl_id>0</port_acl_id>
        <aflex_list/>
        <send_reset>0</send_reset>
        <sync_cookie>0</sync_cookie>
        <source_nat/>
        <server_ssl_template/>
        <conn_reuse_template/>
        <tcp_proxy_template/>
        <source_ip_persistence_template/>
        <sip_template/>
        <pbslb_template/>
        <acl_natpool_binding_list/>
      </vport>
    <vport>
      <protocol>11</protocol>
      <port>80</port>
      <service_group>http</service_group>
      <connection_limit>
        <status>0</status>
        <connection_limit>8000000</connection_limit>
        <connection_limit_action>0</connection_limit_action>
        <connection_limit_log>1</connection_limit_log>
      </connection_limit>
      <default_selection>1</default_selection>
      <received_hop>0</received_hop>
      <status>1</status>
    </vport>
  </virtual_server>
</response>
```

```

<stats_data>1</stats_data>
<extended_stats>0</extended_stats>
<snat_against_vip>0</snat_against_vip>
<vport_template/>
<port_acl_id>0</port_acl_id>
<aflex_list/>
<send_reset>0</send_reset>
<sync_cookie>0</sync_cookie>
<source_nat/>
<http_template/>
<ram_cache_template/>
<tcp_proxy_template>tcppt</tcp_proxy_template>
<server_ssl_template/>
<conn_reuse_template/>
<source_ip_persistence_template/>
<pbslb_template/>
<acl_natpool_binding_list>
    <acl_natpool_binding>
        <acl_id>1</acl_id>
        <nat_pool>test</nat_pool>
    </acl_natpool_binding>
</acl_natpool_binding_list>
</vport>
<vport>
    <protocol>3</protocol>
    <port>12347</port>
    <service_group>udp</service_group>
    <connection_limit>
        <status>0</status>
        <connection_limit>8000000</connection_limit>
        <connection_limit_action>0</connection_limit_action>
        <connection_limit_log>1</connection_limit_log>
    </connection_limit>
    <default_selection>0</default_selection>
    <received_hop>0</received_hop>
    <status>0</status>
    <stats_data>0</stats_data>
    <extended_stats>0</extended_stats>
    <snat_against_vip>0</snat_against_vip>
    <vport_template/>
    <port_acl_id>0</port_acl_id>
    <aflux_list>
        <aflux>DownloadFile1</aflux>
        <aflux>Test01</aflux>
    </aflux_list>
    <ha_connection_mirror>0</ha_connection_mirror>
    <direct_server_return>0</direct_server_return>
    <source_nat/>
    <udp_template/>
    <source_ip_persistence_template/>
    <dns_template/>
    <pbslb_template/>
    <acl_natpool_binding_list/>
</vport>
</vport_list>
</virtual_server>
</response>

```

2. HTTP Request in JSON Format:

session_id=123456789&method=slb.virtual_server.search&name=http_vip1&format=json

Response as the HTTP Body (JSON Format):

```
{  
    "virtual_server":  
    {  
        "name": "http-vip1",  
        "address": "8.2.3.18",  
        "status": 1,  
        "arp_status": 1,  
        "stats_data": 1,  
        "extended_stats": 1,  
        "disable_vserver_on_condition": 0,  
        "redistribution_flagged": 1,  
        "ha_group":  
        {  
            "status": 1,  
            "ha_group_id": 1,  
            "dynamic_server_weight": 11  
        },  
        "vip_template": "default",  
        "pbslb_template": "111",  
        "vport_list": [  
            {  
                "protocol": 3,  
                "port": 12347,  
                "service_group": "udp",  
                "connection_limit":  
                {  
                    "status": 0,  
                    "connection_limit": 8000000,  
                    "connection_limit_action": 0,  
                    "connection_limit_log": 0  
                },  
                "default_selection": 1,  
                "received_hop": 0,  
                "status": 1,  
                "stats_data": 1,  
                "extended_stats": 0,  
                "snat_against_vip": 0,  
                "vport_template": "default",  
                "aflex_list": ["DownloadFile1",  
                             "Test01"],  
                "ha_connection_mirror": 0,  
                "direct_server_return": 0,  
                "source_nat": "",  
                "udp_template": "",  
                "source_ip_persistence_template": "",  
                "dns_template": "",  
                "pbslb_template": "",  
                "acl_natpool_binding_list": []  
            },  
            {  
                "protocol": 13,  
                "port": 80,  
                "service_group": "http",  
                "connection_limit":  
                {  
                    "status": 0,  
                    "connection_limit": 8000000,  
                    "connection_limit_action": 0,  
                    "connection_limit_log": 0  
                },  
                "default_selection": 1,  
                "received_hop": 0,  
                "status": 1,  
                "stats_data": 1,  
            }  
        ]  
    }  
}
```

```

"extended_stats":0,
"snat_against_vip":0,
"vport_template":"default",
"aflex_list":[],
"send_reset":0,
"sync_cookie":
{
    "sync_cookie":0,
    "sack":0
},
"source_nat":"",
"tcp_proxy_template":"",
"client_ssl_template":"",
"server_ssl_template":"",
"source_ip_persistence_template":"",
"pbslb_template":"",
"acl_natpool_binding_list":[]

},
{
    "protocol":17,
    "port":389,
    "service_group":"http",
    "connection_limit":
    {
        "status":0,
        "connection_limit":8000000,
        "connection_limit_action":0,
        "connection_limit_log":0
    },
    "default_selection":1,
    "received_hop":0,
    "status":1,
    "stats_data":1,
    "extended_stats":0,
    "snat_against_vip":0,
    "vport_template":"default",
    "aflex_list":[],
    "send_reset":0,
    "ha_connection_mirror":0,
    "sync_cookie":
    {
        "sync_cookie":1,
        "sack":0
    },
    "source_nat":"",
    "tcp_proxy_template":"",
    "source_ip_persistence_template":"",
    "diameter_template":"",
    "pbslb_template":"",
    "acl_natpool_binding_list":[]

}
]
}
}

```

Example2: search the virtual server by the address.

Request as:

session_id=123456789&method=slb.virtual_server.search&address=8.2.3.18

Response as the HTTP Body (JSON Format):

```
{
    "virtual_server":
    {
        "name":"http-vip1",

```

```

"address":"8.2.3.18",
"status":1,
"arp_status":1,
"stats_data":1,
"extended_stats":1,
"disable_vserver_on_condition":0,
"redistribution_flagged":1,
"ha_group":
{
    "status":1,
    "ha_group_id":1,
    "dynamic_server_weight":11
},
"vip_template":"default",
"pbslb_template":"111",
"vport_list":[
{
    "protocol":3,
    "port":12347,
    "service_group":"udp",
    "connection_limit":
    {
        "status":0,
        "connection_limit":8000000,
        "connection_limit_action":0,
        "connection_limit_log":0
    },
    "default_selection":1,
    "received_hop":0,
    "status":1,
    "stats_data":1,
    "extended_stats":0,
    "snat_against_vip":0,
    "vport_template":"default",
    "aflex_list":["DownloadFile1",
    "Test01"],
    "ha_connection_mirror":0,
    "direct_server_return":0,
    "source_nat":"",
    "udp_template":"",
    "source_ip_persistence_template":"",
    "dns_template":"",
    "pbslb_template":"",
    "acl_natpool_binding_list":[]
},
{
    "protocol":13,
    "port":80,
    "service_group":"http",
    "connection_limit":
    {
        "status":0,
        "connection_limit":8000000,
        "connection_limit_action":0,
        "connection_limit_log":0
    },
    "default_selection":1,
    "received_hop":0,
    "status":1,
    "stats_data":1,
    "extended_stats":0,
    "snat_against_vip":0,
    "vport_template":"default",
    "aflex_list":[],
    "send_reset":0,
    "sync_cookie":"
}
]
}

```

```
{
    "sync_cookie":0,
    "sack":0
},
"source_nat":"",
"tcp_proxy_template":"",
"client_ssl_template":"",
"server_ssl_template":"",
"source_ip_persistence_template":"",
"pbslb_template":"",
"acl_natpool_binding_list":[]
},
{
    "protocol":17,
    "port":389,
    "service_group":"http",
    "connection_limit":
    {
        "status":0,
        "connection_limit":8000000,
        "connection_limit_action":0,
        "connection_limit_log":0
    },
    "default_selection":1,
    "received_hop":0,
    "status":1,
    "stats_data":1,
    "extended_stats":0,
    "snat_against_vip":0,
    "vport_template":"default",
    "aflex_list":[],
    "send_reset":0,
    "ha_connection_mirror":0,
    "sync_cookie":
    {
        "sync_cookie":1,
        "sack":0
    },
    "source_nat":"",
    "tcp_proxy_template":"",
    "source_ip_persistence_template":"",
    "diameter_template":"",
    "pbslb_template":"",
    "acl_natpool_binding_list":[]
}
}
}
```

Example 3: Search the virtual server by ACL ID.

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
GET
- HTTP Body
session_id=123456789&method=slb.virtual_server.search&acl_id=100

Response as the HTTP body:

See “search virtual server by name”.

Example 4: search the virtual server by subnet.

Request as:

- **URL:**
https://_AX_IP_Address_/services/rest/V2/
- **HTTP Action:**
GET
- **HTTP Body**
session_id=123456789&method=slb.virtual_server.search&subnet=address%03192.168.3.2
14&mask_len=24

Response as the HTTP body:

See “search virtual server by name”.

Example 5: search the virtual server by acl name.

Request as:

- **URL:**
https://_AX_IP_Address_/services/rest/V2/
- **HTTP Action:**
GET
- **HTTP Body**
session_id=123456789&method=slb.virtual_server.search&acl_name=v6acl_01

Response as the HTTP body:

See “search virtual server by name”.

Menus Privilege

Config Mode >> Service >> SLB >> Virtual Server

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.3.3 “slb.virtual_server.fetchAllStatistics” Method

This method is used to retrieve all virtual server statistics data and virtual service statistics data that have been configured on the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.virtual_server.fetchAllStatistics	String		

Note: (*) parameter is required in the API.

Response Fields

The response XML is the statistical data for all virtual servers, including the statistics for each virtual services.

virtual_server_stat_list	XML tag for the collection of virtual servers statistic
virtual_server_stat	XML tag for the virtual server statistics
name	Virtual server name.
[]	One of the following token (address, subnet, acl_id, acl_name):
address	virtual server address, either IPv4 or IPv6
subnet	IPv4 subnet
address	address of subnet
mask_len	the length of this subnet
acl_id	IPv4 ACL ID
acl name	IPv6 ACL name

status	Virtual server status 0: disabled 1: all up 2: partition up 3: function up 4: down 5: unknown																																				
cur_conn	total number of current connections																																				
tot_conn	total number of connections, ulong64																																				
req_pkts	total number of request packets received, ulong64																																				
resp_pkts	total number of response packets sent, ulong64																																				
req_bytes	total number of request bytes received, ulong64																																				
resp_bytes	total number of response bytes sent, ulong64																																				
cur_reqs	total number of current requests																																				
total_reqs	total number of requests, ulong64																																				
total_reqs_succ	total number of successful requests, ulong64																																				
vport_stat_list	XML tag for the virtual server ports																																				
vport_stat	XML tag for a single virtual server port, and contains the following properties: port virtual server port number protocol virtual port protocol <table border="0"> <tr><td>TCP</td><td>2</td></tr> <tr><td>UDP</td><td>3</td></tr> <tr><td>Other</td><td>4</td></tr> <tr><td>RTSP</td><td>5</td></tr> <tr><td>FTP</td><td>6</td></tr> <tr><td>MMS</td><td>7</td></tr> <tr><td>SIP</td><td>8</td></tr> <tr><td>FAST-HTTP</td><td>9</td></tr> <tr><td>TCP-PROXY</td><td>10</td></tr> <tr><td>HTTP</td><td>11</td></tr> <tr><td>HTTPS</td><td>12</td></tr> <tr><td>SSL-PROXY</td><td>13</td></tr> <tr><td>SMTP</td><td>14</td></tr> <tr><td>SIP-TCP</td><td>15</td></tr> <tr><td>SIP-STL</td><td>16</td></tr> <tr><td>DIAMETER</td><td>17</td></tr> <tr><td>DNS-UDP</td><td>18</td></tr> <tr><td>TFTP</td><td>19</td></tr> </table> status virtual port status. 0: disabled 1: all up 2: partition up 3: function up 4: down 5: unknown	TCP	2	UDP	3	Other	4	RTSP	5	FTP	6	MMS	7	SIP	8	FAST-HTTP	9	TCP-PROXY	10	HTTP	11	HTTPS	12	SSL-PROXY	13	SMTP	14	SIP-TCP	15	SIP-STL	16	DIAMETER	17	DNS-UDP	18	TFTP	19
TCP	2																																				
UDP	3																																				
Other	4																																				
RTSP	5																																				
FTP	6																																				
MMS	7																																				
SIP	8																																				
FAST-HTTP	9																																				
TCP-PROXY	10																																				
HTTP	11																																				
HTTPS	12																																				
SSL-PROXY	13																																				
SMTP	14																																				
SIP-TCP	15																																				
SIP-STL	16																																				
DIAMETER	17																																				
DNS-UDP	18																																				
TFTP	19																																				
cur_conn	total number of current connections																																				
tot_conn	total number of connections, ulong64																																				
req_pkts	total number of request packets received, ulong64																																				
resp_pkts	total number of response packets sent, ulong64																																				
req_bytes	total number of request bytes received, ulong64																																				
resp_bytes	total number of response bytes sent, ulong64																																				
cur_reqs	total number of current requests																																				
total_reqs	total number of requests, ulong64																																				
total_reqs_succ	total number of successful requests, ulong64																																				

Example Response

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
slb virtual-server AAA 25.25.25.65
  port 80  tcp
    name _25.25.25.65_TCP_80
    service-group http
!
```

1. HTTP Request in URL Format:

```
session_id=123456789&method=slb.virtual_server.fetchAllStatistics&format=url
```

Response as the HTTP body (URL Format):

```
<response status="ok">
<virtual_server_stat_list>
  <virtual_server_stat>
    <name>AAA</name>
    <address>8.2.3.18</address>
    <status>1</status>
    <cur_conns>0</cur_conns>
    <tot_conns>0</tot_conns>
    <req_pkts>0</req_pkts>
    <resp_pkts>0</resp_pkts>
    <req_bytes>0</req_bytes>
    <resp_bytes>0</resp_bytes>
    <cur_reqs>0</cur_reqs>
    <total_reqs>0</total_reqs>
    <total_reqs_succ>0</total_reqs_succ>
    <vport_stat_list>
      <vport_stat>
        <port>389</port>
        <protocol>15</protocol>
        <status>0</status>
        <cur_conns>0</cur_conns>
        <tot_conns>0</tot_conns>
        <req_pkts>0</req_pkts>
        <resp_pkts>0</resp_pkts>
        <req_bytes>0</req_bytes>
        <resp_bytes>0</resp_bytes>
        <cur_reqs>0</cur_reqs>
        <total_reqs>0</total_reqs>
        <total_reqs_succ>0</total_reqs_succ>
      </vport_stat>
      <vport_stat>
        <port>80</port>
        <protocol>11</protocol>
        <status>0</status>
        <cur_conns>0</cur_conns>
        <tot_conns>0</tot_conns>
        <req_pkts>0</req_pkts>
        <resp_pkts>0</resp_pkts>
        <req_bytes>0</req_bytes>
        <resp_bytes>0</resp_bytes>
        <cur_reqs>0</cur_reqs>
        <total_reqs>0</total_reqs>
        <total_reqs_succ>0</total_reqs_succ>
      </vport_stat>
    </vport_stat_list>
  </virtual_server_stat>
</virtual_server_stat_list>
</response>
```

```

<port>12347</port>
<protocol>3</protocol>
<status>0</status>
<cur_conns>0</cur_conns>
<tot_conns>0</tot_conns>
<req_pkts>0</req_pkts>
<resp_pkts>0</resp_pkts>
<req_bytes>0</req_bytes>
<resp_bytes>0</resp_bytes>
<cur_reqs>0</cur_reqs>
<total_reqs>0</total_reqs>
<total_reqs_succ>0</total_reqs_succ>
</vport_stat>
</vport_stat_list>
</virtual_server_stat>
</virtual_server_stat_list>
</response>

```

2. HTTP Request in JSON Format:

```
session_id=123456789&method=slb.virtual_server.fetchAllStatistics&format=json
```

Response as the HTTP body (JSON Format):

```
{
  "virtual_server_stat_list": [
    {
      "name": "AAA",
      "address": "25.25.25.65",
      "status": 1,
      "cur_conns": 0,
      "tot_conns": 0,
      "req_pkts": 0,
      "resp_pkts": 0,
      "req_bytes": 0,
      "resp_bytes": 0,
      "cur_reqs": 0,
      "total_reqs": 0,
      "total_reqs_succ": 0,
      "vport_stat_list": [
        {
          "port": 80,
          "protocol": 2,
          "status": 0,
          "cur_conns": 0,
          "tot_conns": 0,
          "req_pkts": 0,
          "resp_pkts": 0,
          "req_bytes": 0,
          "resp_bytes": 0,
          "cur_reqs": 0,
          "total_reqs": 0,
          "total_reqs_succ": 0
        }
      ]
    }
  ]
}
```

Menus Privilege

Monitor Mode >> Service >> SLB >> Virtual Server

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.3.4 “slb.virtual_server.fetchStatistics” Method

This method is used to retrieve virtual server statistics data by entering a given IP address (IPv4 or IPv6) or virtual server name.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.virtual_server.fetchStatistics	String		
name (**)	Virtual server name	String		
One of the following parameters (address, subnet, acl_id, acl_name)(**)				
address	The address of this virtual server (fetch the virtual server by address, either IPv4 or IPv6)	Ip		
subnet address mask_len	The subnet of this virtual server (fetch the virtual server by subnet address and mask length, only for IPv4)	IPv4 Int	1 – 32	
acl_id	The acl id of this virtual server (fetch the virtual server by acl id, for IPv4 wildcard virtual server only)	Int	1 - 199	
acl_name	The acl name of this virtual server (fetch the virtual server by acl name, for IPv6 wildcard virtual server only)	String		

Note: (*) parameter is required in the API.

(**) Either one of parameters is required.

Response Fields

virtual_server_stat	XML tag for the virtual server statistics
name	Virtual server name.
[]	One of the following token (address, subnet, acl_id, acl_name):
address	virtual server address, either IPv4 or IPv6
subnet	IPv4 subnet
address	address of subnet
mask_len	the length of this subnet
acl_id	IPv4 ACL id
acl name	IPv6 ACL name
status	Virtual server status 0: disabled 1: all up 2: partition up 3: function up 4: down 5: unknown
cur_conn	total number of current connections
tot_conn	total number of connections, ulong64
req_pkts	total number of request packets received, ulong64
resp_pkts	total number of response packets sent, ulong64

req_bytes	total number of request bytes received, ulong64
resp_bytes	total number of response bytes sent, ulong64
cur_reqs	total number of current requests
total_reqs	total number of requests, ulong64
total_reqs_succ	total number of successful requests, ulong64
vport_stat_list	XML tag for the virtual server ports
vport_stat	XML tag for a single virtual server port, and contains the following properties:
port	virtual server port number
protocol	virtual port protocol
TCP	2
UDP	3
Other	4
RTSP	5
FTP	6
MMS	7
SIP	8
FAST-HTTP	9
TCP-PROXY	10
HTTP	11
HTTPS	12
SSL-PROXY	13
SMTP	14
SIP-TCP	15
SIP-STL	16
DIAMETER	17
DNS-UDP	18
TFTP	19
status	virtual port status.
0:	disabled
1:	all up
2:	partition up
3:	function up
4:	down
5:	unknown
cur_conns	total number of current connections
tot_conns	total number of connections, ulong64
req_pkts	total number of request packets received, ulong64
resp_pkts	total number of response packets sent, ulong64
req_bytes	total number of request bytes received, ulong64
resp_bytes	total number of response bytes sent, ulong64
cur_reqs	total number of current requests
total_reqs	total number of requests, ulong64
total_reqs_succ	total number of successful requests, ulong64

Example Response

Example1: fetch the virtual server statistical data by the address.

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
slb virtual-server AAA 25.25.25.65
    port 80  tcp
```

```
name _25.25.25.65_TCP_80
service-group http
```

```
!
```

1. HTTP Request in URL Format:

```
session_id=123456789&method=slb.virtual_server.fetchStatistics&address=25.25.25.65&format
=url
```

Response as the HTTP body (URL Format):

```
<response status="ok">
  <virtual_server_stat>
    <name>AAA</name>
    <address>8.2.3.18</address>
    <status>1</status>
    <cur_conns>0</cur_conns>
    <tot_conns>0</tot_conns>
    <req_pkts>0</req_pkts>
    <resp_pkts>0</resp_pkts>
    <req_bytes>0</req_bytes>
    <resp_bytes>0</resp_bytes>
    <cur_reqs>0</cur_reqs>
    <total_reqs>0</total_reqs>
    <total_reqs_succ>0</total_reqs_succ>
    <vport_stat_list>
      <vport_stat>
        <port>389</port>
        <protocol>15</protocol>
        <status>0</status>
        <cur_conns>0</cur_conns>
        <tot_conns>0</tot_conns>
        <req_pkts>0</req_pkts>
        <resp_pkts>0</resp_pkts>
        <req_bytes>0</req_bytes>
        <resp_bytes>0</resp_bytes>
        <cur_reqs>0</cur_reqs>
        <total_reqs>0</total_reqs>
        <total_reqs_succ>0</total_reqs_succ>
      </vport_stat>
      <vport_stat>
        <port>80</port>
        <protocol>11</protocol>
        <status>0</status>
        <cur_conns>0</cur_conns>
        <tot_conns>0</tot_conns>
        <req_pkts>0</req_pkts>
        <resp_pkts>0</resp_pkts>
        <req_bytes>0</req_bytes>
        <resp_bytes>0</resp_bytes>
        <cur_reqs>0</cur_reqs>
        <total_reqs>0</total_reqs>
        <total_reqs_succ>0</total_reqs_succ>
      </vport_stat>
      <vport_stat>
        <port>12347</port>
        <protocol>3</protocol>
        <status>0</status>
        <cur_conns>0</cur_conns>
        <tot_conns>0</tot_conns>
        <req_pkts>0</req_pkts>
        <resp_pkts>0</resp_pkts>
        <req_bytes>0</req_bytes>
```

```

<resp_bytes>0</resp_bytes>
<cur_reqs>0</cur_reqs>
<total_reqs>0</total_reqs>
<total_reqs_succ>0</total_reqs_succ>
</vport_stat>
</vport_stat_list>
</virtual_server_stat>
</response>

```

2. HTTP Request in JSON Format:

session_id=123456789&method=slb.virtual_server.fetchStatistics&address=25.25.25.65&format=json

Response as the HTTP body (JSON Format):

```
{
  "virtual_server_stat": [
    {
      "name": "AAA",
      "address": "25.25.25.65",
      "status": 1,
      "cur_conns": 0,
      "tot_conns": 0,
      "req_pkts": 0,
      "resp_pkts": 0,
      "req_bytes": 0,
      "resp_bytes": 0,
      "cur_reqs": 0,
      "total_reqs": 0,
      "total_reqs_succ": 0,
      "vport_stat_list": [
        {
          "port": 80,
          "protocol": 2,
          "status": 0,
          "cur_conns": 0,
          "tot_conns": 0,
          "req_pkts": 0,
          "resp_pkts": 0,
          "req_bytes": 0,
          "resp_bytes": 0,
          "cur_reqs": 0,
          "total_reqs": 0,
          "total_reqs_succ": 0
        }
      ]
    }
  ]
}
```

Example 2: fetch the virtual server statistical data by the name.

Request as:

session_id=123456789&method=slb.virtual_server.fetchStatistics&name=AAA

Response as the HTTP body (JSON Format):

```
{
  "virtual_server_stat": [
    {
      "name": "AAA",
      "address": "25.25.25.65",
      "status": 1,
```

```

"cur_conns":0,
"tot_conns":0,
"req_pkts":0,
"resp_pkts":0,
"req_bytes":0,
"resp_bytes":0,
"cur_reqs":0,
"total_reqs":0,
"total_reqs_succ":0,
"vport_stat_list":[
{
    "port":80,
    "protocol":2,
    "status":0,
    "cur_conns":0,
    "tot_conns":0,
    "req_pkts":0,
    "resp_pkts":0,
    "req_bytes":0,
    "resp_bytes":0,
    "cur_reqs":0,
    "total_reqs":0,
    "total_reqs_succ":0
}
]
}
}

```

Example 3: Fetch the virtual server statistical data by the subnet.

Request as (URL Format):

```
session_id=123456789&method=slb.virtual_server.fetchStatistics&subnet=address%03192.168.3
.214%02mask_len%0324
```

Response as the HTTP body:

See 'fetch virtual server statistics by address'.

Example 4: fetch the virtual server statistical data by the acl id.

Request as:

```
session_id=123456789&method=slb.virtual_server.fetchStatistics&acl_id=100
```

Response as the HTTP body:

See 'fetch virtual server statistics by address'.

Example 5: fetch the virtual server statistical data by the acl name.

Request as:

```
session_id=123456789&method=slb.virtual_server.fetchStatistics&acl_name=vip_test_01
```

Response as the HTTP body:

See 'fetch virtual server statistics by address'.

Menus Privilege

Monitor Mode >> Service >> SLB >> Virtual Server

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.3.5 “slb.virtual_server.create” Method

This method is used to create a virtual server and its virtual services.

This method is required to use the HTTP POST method.

arp_status	ARP disabled option, either disabled(0) or enabled(1)
stat_data	stats data option, either disabled(0) or enabled(1)
ext_stat	extended stat option, either disabled(0) or enabled(1)
when_all_down	when all ports down option, either disabled(0) or enabled(1)
redis_flag	redistribution flagged option, either disabled(0) or enabled(1)
ha_group	configured HA group
vip_templ	virtual server template name
pbslb_templ	pbslb policy template name

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.vip.create	String		
name (*)	Virtual server name	String		
[] One of the following token (address, subnet, acl_id, acl_name): address(*) subnet address mask_len acl_id acl_name				
status	Virtual server status, either enabled(1) or disabled(0)	Int	1 or 0	1
arp_status	Virtual server arp status, either enabled(1) or disabled(0)	Int	1 or 0	1
stats_data	Virtual server stats data, either enabled(1) or disabled(0)	Int	1 or 0	1
extended_stats	Virtual server extended stats, enabled(1) or disabled(0)	Int	1 or 0	1
disable_vserver_on_condition	Disable virtual server on conditions. Do not disable virtual server in any case(0), disable virtual server when any port down (1), disable virtual server when all ports down (2). The corresponding CLI command is: enable when-any-port-down and enable when-all-ports-down under slb virtual server sub mode.	Int	0 - 3	0
redistribution_flagged	Virtual server redistribution flagged option, either enabled(1) or disabled(0)	Int	1 or 0	1
ha_group status ha_group_id dynamic_server_weight	Configure HA group Ha status ha group id dynamic server weight	Int Int Int	0 or 1 1 – 31 1 - 255	
vip_template	Virtual server template name	String		default
pbslb_template	Virtual server pbslb policy template name	String		
vport_list	Virtual service list to be added: service1^Cservice1_config^B service2^Cservice2_config^B ... serviceN^CserviceN_config ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-	String		

	<p>encode %03</p> <p>Note: serviceN_config is defined by the virtual type of element N. And, it can be empty.</p>																																						
vport <n> protocol: (*) port: (*) service_group: (*) status: connection_limit: status connection_limit connection_limit_action connection_limit_log default_selection received_hop stats_data extended_stats snat_against_vip vport_template port_acl_id: aflex_list: aflex pbslb_template: acl_natpool_binding_list acl_natpool_binding [] acl_id acl_name nat_pool	<p>Virtual service at element <n></p> <p>Virtual service type:</p> <table> <tbody> <tr><td>TCP</td><td>2</td></tr> <tr><td>UDP</td><td>3</td></tr> <tr><td>Other</td><td>4</td></tr> <tr><td>RTSP</td><td>5</td></tr> <tr><td>FTP</td><td>6</td></tr> <tr><td>MMS</td><td>7</td></tr> <tr><td>SIP</td><td>8</td></tr> <tr><td>FAST-HTTP</td><td>9</td></tr> <tr><td>TCP-PROXY</td><td>10</td></tr> <tr><td>HTTP</td><td>11</td></tr> <tr><td>HTTPS</td><td>12</td></tr> <tr><td>SSL-PROXY</td><td>13</td></tr> <tr><td>SMTP</td><td>14</td></tr> <tr><td>SIP-TCP</td><td>15</td></tr> <tr><td>SIP-STL</td><td>16</td></tr> <tr><td>DIAMETER</td><td>17</td></tr> <tr><td>DNS-UDP</td><td>18</td></tr> <tr><td>TFTP</td><td>19</td></tr> </tbody> </table> <p>Virtual service port number</p> <p>Service group assigned to the virtual service</p> <p>Virtual service status, either enabled(1) or disabled(0)</p> <p>Connection limit of the virtual service</p> <p>connection limit value</p> <p>connection limit action, either reset(1) or drop(0)</p> <p>connection limit logging, either enabled(1) or disabled(0)</p> <p>use default server selection when preferred method fails, either enabled(1) or disabled(0)</p> <p>use received hop for response either enabled(1) or disabled(0)</p> <p>stats data, either enabled(1) or disabled(0)</p> <p>extended stats, either enabled(1) or disabled(0)</p> <p>source nat traffic against vip either enabled(1) or disabled(0)</p> <p>virtual server port template name</p> <p>Assigned access-list number</p> <p>AFlex list</p> <p>Aflex name</p> <p>Assigned PBSLB template name</p> <p>acl – natpool binding list</p> <p>acl – natpool binding</p> <p>one of the following properties: IPv4 ACL id IPv6 ACL name source nat pool name</p> <p>Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03</p>	TCP	2	UDP	3	Other	4	RTSP	5	FTP	6	MMS	7	SIP	8	FAST-HTTP	9	TCP-PROXY	10	HTTP	11	HTTPS	12	SSL-PROXY	13	SMTP	14	SIP-TCP	15	SIP-STL	16	DIAMETER	17	DNS-UDP	18	TFTP	19	<p>Int</p> <p>Int</p> <p>String</p> <p>Int</p> <p>Structure</p> <p>Int</p> <p>Int</p> <p>Int</p> <p>Int</p> <p>Int</p> <p>Int</p> <p>Int</p> <p>Int</p> <p>Int</p> <p>String</p> <p>Int</p> <p>String</p> <p>String</p> <p>String</p> <p>Int</p> <p>String</p> <p>String</p>	<p>0 to 65534</p> <p>1 or 0</p> <p>0 or 1</p> <p>1 to 8000000</p> <p>0 or 1</p> <p>1 – 199</p> <p>1 – 199</p>
TCP	2																																						
UDP	3																																						
Other	4																																						
RTSP	5																																						
FTP	6																																						
MMS	7																																						
SIP	8																																						
FAST-HTTP	9																																						
TCP-PROXY	10																																						
HTTP	11																																						
HTTPS	12																																						
SSL-PROXY	13																																						
SMTP	14																																						
SIP-TCP	15																																						
SIP-STL	16																																						
DIAMETER	17																																						
DNS-UDP	18																																						
TFTP	19																																						

http vport has the following additional properties:	send_reset sync_cookie source_nat http_template ram_cache_template tcp_proxy_template server_ssl_template conn_reuse_template [] source_ip_persistence_template cookie_persistence_template <p>Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03</p>	String String Int	0 – 1	
https vport has the following additional properties:	send_reset sync_cookie source_nat http_template ram_cache_template tcp_proxy_template client_ssl_template server_ssl_template conn_reuse_template [] source_ip_persistence_template cookie_persistence_template <p>Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03</p>	String String Int		
Fast http vport has the following additional properties:	send_reset sync_cookie source_nat http_template tcp_template conn_reuse_template [] source_ip_persistence_template <p>send client reset when server selection fails, either enabled(1) or disabled(0) syn cookie, either enabled(1) or disabled(0) source NAT pool name HTTP template name TCP template name Connection-Reuse template name one of the following properties:</p>	String String Int		

cookie_persistence_template	Source-IP-Persistent template name cookie persistence template name Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03	String		
TCP vport has the following additional properties: send_reset ha_connection_mirror direct_server_return sync_cookie source_nat tcp_template conn_reuse_template [] source_ip_persistence_template cookie_persistence_template	send client reset when server selection fails, either enabled(1) or disabled(0) HA connection mirror enabled(1) or disabled (0) direct server return (no destination NAT) enabled (1) or disabled (0) syn cookie, either enabled(1) or disabled(0) source NAT pool name TCP template name Connection-Reuse template name one of the following properties: Source-IP-Persistent template name cookie persistence template name Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03	String String Int Int Int Int String String String String String		
UDP vport has the following additional properties: ha_connection_mirror direct_server_return source_nat udp_template dns_template source_ip_persistence_template	HA connection mirror enabled(1) or disabled (0) direct server return (no destination NAT) enabled (1) or disabled (0) source NAT pool name UDP template name UDP template name one of the following properties: Source-IP-Persistent template name Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03	Int Int String String String String		
RTSP vport has the following additional properties: send_reset ha_connection_mirror direct_server_return	send client reset when server selection fails, either enabled(1) or disabled(0) HA connection mirror enabled(1) or	Int Int Int		

	sync_cookie tcp_template source_ip_persistence_template	disabled (0) direct server return (no destination NAT) enabled (1) or disabled (0) syn cookie, either enabled(1) or disabled(0) TCP template name Source-IP-Persistent template name Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03	Int String String		
FTP vport has the following additional properties:	send_reset ha_connection_mirror direct_server_return sync_cookie source_nat tcp_template source_ip_persistence_template	send client reset when server selection fails, either enabled(1) or disabled(0) HA connection mirror enabled(1) or disabled (0) direct server return (no destination NAT) enabled (1) or disabled (0) syn cookie, either enabled(1) or disabled(0) sourc nat pool name TCP template name Source-IP-Persistent template name Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03	Int Int Int Int String String String		
mms vport has the following additional properties:	send_reset ha_connection_mirror sync_cookie tcp_temp source_ip_p_temp	send client reset when server selection fails, either enabled(1) or disabled(0) HA connection mirror enabled(1) or disabled (0) Syn Cookie flag, either enabled(1) or disabled(0) TCP template name Source IP persistent template name Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03	Int Int String String	1 or 0	0
SSL-Proxy vport has the following additional properties:	send_reset sync_cookie source_nat tcp_proxy_template client_ssl_template server_ssl_template conn_reuse_template	send client reset when server selection fails, either enabled(1) or disabled(0) syn cookie, either enabled(1) or disabled(0) source NAT pool name TCP-Proxy template name client ssl template Server-SSL template name	Int String String String String String		

	source_ip_persistence_template SMPT vport has the following additional properties: send_reset sync_cookie source_nat tcp_proxy_template client_ssl_template smtp_template source_ip_persistence_template	Connection-Reuse template name Source-IP-Persistent template name Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03	String		
	SIP vport has the following additional properties: send_reset ha_connection_mirror udp_template source_ip_persistence_template sip_template dns_template	send client reset when server selection fails, either enabled(1) or disabled(0) syn cookie, either enabled(1) or disabled(0) source NAT pool name TCP-Proxy template name client ssl template smtp template Source-IP-Persistent template name Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03	Int Int String String String String String		
	SIP-TCP vport has the following additional properties: send_reset sync_cookie source_nat server_ssl_template conn_reuse_template tcp_proxy_template source_ip_persistence_template sip_template	send client reset when server selection fails, either enabled(1) or disabled(0)\ syn cookie, either enabled(1) or disabled(0) source NAT pool name Server-SSL template name Connection-Reuse template name TCP-Proxy template name Source-IP-Persistent template name SIP template name	Int Int String String String String String		

	<p>Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03</p>			
sip-tls vport has the following additional properties: send_reset sync_cookie source_nat client_ssl_template server_ssl_template conn_reuse_template tcp_proxy_template source_ip_persistence_template sip_template	send client reset when server selection fails, either enabled(1) or disabled(0) syn cookie, either enabled(1) or disabled(0) source NAT pool name client ssl template Server-SSL template name Connection-Reuse template name TCP-Proxy template name Source-IP-Persistent template name Sip template name <p>Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03</p>	Int Int String String String String String String String String		
tcp-proxy vport has the following additional properties: send_reset sync_cookie source_nat tcp_proxy_template source_ip_persistence_template	send client reset when server selection fails, either enabled(1) or disabled(0) syn cookie, either enabled(1) or disabled(0) source NAT pool name TCP-Proxy template name Source-IP-Persistent template name <p>Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03</p>			
dns-udp vport has the following additional properties: send_reset source_nat	send client reset when server selection fails,			

udp_template source_ip_persistence_template dns_template	<p>either enabled(1) or disabled(0) source NAT pool name UDP template name Source-IP-Persistent template name Dns template name</p> <p>Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03</p>			
diameter-udp vport has the following additional properties:	<p>send_reset</p> <p>source_nat</p> <p>udp_template</p> <p>source_ip_persistence_template</p> <p>dns_template</p> <p>send client reset when server selection fails, either enabled(1) or disabled(0) source NAT pool name</p> <p>UDP template name</p> <p>Source-IP-Persistent template name</p> <p>UDP template name</p> <p>Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03</p>			
tftp vport has the following additional properties:	<p>ha_connection_mirror</p> <p>direct_server_return</p> <p>source_nat</p> <p>udp_template</p> <p>HA connection mirror enabled(1) or disabled (0)</p> <p>direct server return (no destination NAT) enabled (1)</p> <p>source NAT pool name</p> <p>UDP template name</p> <p>Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03</p>			
'other' vport has the following additional properties:				

direct_server_return	direct server return (no destination NAT) enabled (1)			
source_nat	source NAT pool name			
udp_template	UDP template name			
l4_template_type	TCP (2) or UDP(3)			
[] udp_template tcp_template	one of the following properties: UDP template name TCP template name			

Note:

In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option.
 ^B: ASCII Code 0x02, URL-encode %02
 ^C: ASCII Code 0x03, URL-encode %03

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Example 1: Create a virtual server with subnet.

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
access-list 1 permit any
access-list 111 permit any
!
!
slb template policy 111
!
!
ip nat pool test 7.7.7.7 7.7.7.7 netmask /24
!
```

1. HTTP Request in URL Format:

```
method=slb.virtual_server.create&virtual_server=name%03pltest7%02subnet%02vport_li
st%02status%030%02arp_status%030%02stats_data%030%02extended_stat%031%02disable_vserver_o
n_condition%030%02redistribution_flagged%031%02ha_group%031%02pbslb_template%03111&subnet
=address%0315.22.55.5%02mask_len%0324&vport_list=11%0212&11=protocol%032%02port%0380&l2=p
rotocol%033%02port%0320%02port_acl_id%03100%02aflex_list%02acl_natpool_binding_list&aflex
_list=f1%02f2&f1=DownloadFile1&acl_natpool_binding_list=h1%02h2&h1=acl_id%031%02nat_pool%
03test&h2=acl_id%03100%02nat_pool%03123&format=url
```

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in JSON Format:

method=slb.virtual_server.create&format=json

Request POST data (JSON Format):

```
{  
    "virtual_server":  
    {  
        "name":"pltest7",  
        "subnet":  
        {  
            "address":"15.22.55.5",  
            "mask_len":24  
        },  
        "status":0,  
        "arp_status":0,  
        "stats_data":0,  
        "extended_stats":0,  
        "disable_vserver_on_condition":0,  
        "redistribution_flagged":1,  
        "ha_group":  
        {  
            "status":0  
        },  
        "vip_template":"default",  
        "pbslb_template":"111",  
        "vport_list":  
        [  
            {  
                "protocol":3,  
                "port":20,  
                "service_group": "",  
                "connection_limit":  
                {  
                    "status":0,  
                    "connection_limit":8000000,  
                    "connection_limit_action":0,  
                    "connection_limit_log":0  
                },  
                "default_selection":1,  
                "received_hop":0,  
                "status":1,  
                "stats_data":1,  
                "extended_stats":0,  
                "snat_against_vip":0,  
                "vport_template":"default",  
                "port_acl_id":3,  
                "aflex_list":["DownloadFile1"],  
                "ha_connection_mirror":0,  
                "direct_server_return":0,  
                "source_nat": "",  
                "udp_template": "",  
                "source_ip_persistence_template": "",  
                "dns_template": "",  
                "pbslb_template": "",  
                "acl_natpool_binding_list":  
                [  
                    {  
                        "acl_id":1,  
                        "nat_pool":"test"  
                    }  
                ]  
            },  
        ]  
    },  
}
```

```
{
    "protocol":2,
    "port":80,
    "service_group":"",
    "connection_limit":
    {
        "status":0,
        "connection_limit":8000000,
        "connection_limit_action":0,
        "connection_limit_log":0
    },
    "default_selection":1,
    "received_hop":0,
    "status":1,
    "stats_data":1,
    "extended_stats":0,
    "snat_against_vip":0,
    "vport_template":"default",
    "aflex_list":[],
    "send_reset":0,
    "ha_connection_mirror":0,
    "direct_server_return":0,
    "sync_cookie":
    {
        "sync_cookie":0,
        "sack":0
    },
    "source_nat":"",
    "tcp_template":"",
    "source_ip_persistence_template":"",
    "pbslb_template":"",
    "acl_natpool_binding_list":[]
}
}
```

Response as the HTTP body (JSON Format):

```
{
    "response":
    {
        "status": "OK"
    }
}
```

CLI configuration after aXAPI call:

```
!
slb virtual-server pltest7 15.22.55.5 /24
    disable
    redistribution-flagged
    arp-disable
    template policy 111
    stats-data-disable
    port 80 tcp
        name _15.22.55.5(24)_TCP_80
    port 20 udp
        name _15.22.55.5(24)_UDP_20
        access-list 1 source-nat-pool test
        access-list 100
        aflex DownloadFile1
!
```

Example 2: Create a virtual server with IPv6 address.

```
method=slb.virtual_server.create&virtual_server=name%03pltest7%02address%033ff5::1
234%02vport_list%02status%030%02arp_status%030%02stats_data%030%02extended_stat%03
1%02disable_vserver_on_condition%030%02redistribution_flagged%031%02ha_group%031%0
2pbslb_template%0311&vport_list=l1%0212&l1=protocol%032%02port%0380&l2=protocol%03
3%02port%0320%02port_acl_id%03100%02aflex_list%02acl_natpool_binding_list&aflex_li
st=f1%02f2&f1=DownloadFile1&f2=aflex2&acl_natpool_binding_list=h1%02h2&h1=acl_id%0
31%02nat_pool%03test&h2=acl_id%03100%02nat_pool%03123&format=url
```

Request POST data (JSON Format):

```
{
  "virtual_server": {
    "name": "pltest7",
    "subnet": {
      "address": "3ff5::1234"
    },
    "status": 0,
    "arp_status": 0,
    "stats_data": 0,
    "extended_stats": 0,
    "disable_vserver_on_condition": 0,
    "redistribution_flagged": 1,
    "ha_group": {
      "status": 0
    },
    "vip_template": "default",
    "pbslb_template": "111",
    "vport_list": [
      {
        "protocol": 3,
        "port": 20,
        "service_group": "",
        "connection_limit": {
          "status": 0,
          "connection_limit": 8000000,
          "connection_limit_action": 0,
          "connection_limit_log": 0
        },
        "default_selection": 1,
        "received_hop": 0,
        "status": 1,
        "stats_data": 1,
        "extended_stats": 0,
        "snat_against_vip": 0,
        "vport_template": "default",
        "port_acl_id": 3,
        "aflex_list": ["tt"],
        "ha_connection_mirror": 0,
        "direct_server_return": 0,
        "source_nat": "",
        "udp_template": "",
        "source_ip_persistence_template": "",
        "dns_template": "",
        "pbslb_template": "",
        "acl_natpool_binding_list": [
          {
            "acl_id": 1,
            "nat_pool": "test"
          }
        ]
      }
    ]
  }
}
```

```

"protocol":2,
"port":80,
"service_group":"",
"connection_limit":
{
    "status":0,
    "connection_limit":8000000,
    "connection_limit_action":0,
    "connection_limit_log":0
},
"default_selection":1,
"received_hop":0,
"status":1,
"stats_data":1,
"extended_stats":0,
"snat_against_vip":0,
"vport_template":"default",
"aflex_list":[],
"send_reset":0,
"ha_connection_mirror":0,
"direct_server_return":0,
"sync_cookie":
{
    "sync_cookie":0,
    "sack":0
},
"source_nat":"",
"tcp_template":"",
"source_ip_persistence_template":"",
"pbslb_template":"",
"acl_natpool_binding_list":[]
}
]
}
}

```

Menus Privilege

Config Mode >> Service >> SLB >> Virtual Server

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.3.6 “slb.virtual_server.update” Method

This method is used to update parameters for an existing virtual server, such as ARP status, or HA status, or connection limiting. **This method is required to use the HTTP POST method.**

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.vip.update	String		
name (*)	Virtual server name	String		
[] One of the following token (address, subnet, acl_id, acl_name): address(*) subnet address mask_len	Virtual server IP address, either IPv4 or IPv6 IPv4 subnet address of subnet	IPv4/v6 IPv4/v6 Int	1 – 32	

acl_id	the length of this subnet	Int	1 – 199	
acl_name	IPv4 ACL id IPv6 ACL name	String		
status	Virtual server status, either enabled(1) or disabled(0)	Int	1 or 0	1
arp_status	Virtual server arp status, either enabled(1) or disabled(0)	Int	1 or 0	1
stats_data	Virtual server stats data, either enabled(1) or disabled(0)	Int	1 or 0	1
extended_stats	Virtual server extended stats, enabled(1) or disabled(0)	Int	1 or 0	1
disable_vserver_on_condition	Disable virtual server on conditions. Do not disable virtual server in any case(0), disable virtual server when any port down (1), disable virtual server when all ports down (2). The corresponding CLI command is: disable when-any-port-down and disable when-all-ports-down under slb virtual server sub mode.	Int	0 - 2	0
redistribution_flagged	Virtual server redistribution flagged option, either enabled(1) or disabled(0)	Int	1 or 0	1
ha_group	Configure HA group	Int	0 or 1	
status	Ha status	Int	1 – 31	
ha_group_id	ha group id	Int	1 - 255	
dynamic_server_weight	dynamic server weight	Int		
vip_template	Virtual server template name	String		default
pbslb_template	Virtual server pbslb policy template name	String		
vport_list	Virtual service list to be added: service1^Cservice1_config^B service2^Cservice2_config^B ... serviceN^CserviceN_config ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03 Note: serviceN_config is defined by the virtual type of element N. And, it can be empty.	String		
vport <n> protocol: (*)	Virtual service at element <n> Virtual service type: TCP 2 UDP 3 Other 4 RTSP 5 FTP 6 MMS 7 SIP 8 FAST-HTTP 9 TCP-PROXY 10 HTTP 11 HTTPS 12 SSL-PROXY 13 SMTP 14 SIP-TCP 15 SIP-STL 16 DIAMETER 17 DNS-UDP 18 TFTP 19	Int		
port: (*) service_group: (*) status: connection_limit:	Virtual service port number Service group assigned to the virtual service Virtual service status, either enabled(1) or disabled(0) Connection limit of the virtual service connection limit value	Int String Int Int Int	0-65534 1 or 0 1 to 8000000 0 or 1	

default_selection received_hop stats_data extended_stats snat_against_vip vport_template port_acl_id: aflex_list: aflex pbslb_template: acl_natpool_binding_list acl_natpool_binding [] acl_id acl_name nat_pool	connection limit action, either reset(1) or drop(0) connection limit logging, either enabled(1) or disabled(0) use default server selection when preferred method fails, either enabled(1) or disabled(0) use received hop for response either enabled(1) or disabled(0) stats data, either enabled(1) or disabled(0) extended stats, either enabled(1) or disabled(0) source nat traffic against vip either enabled(1) or disabled(0) virtual server port template name Assigned access-list number AFlex list Aflex name Assigned PBSLB template name acl – natpool binding list acl – natpool binding one of the following properties: IPv4 acl id IPv6 acl name source nat pool name	Int Int Int Int Int Int String Int	0 or 1 0 or 1 0 or 1 0 or 1 0 or 1 0 or 1 1 – 199	
http vport has the following additional properties: send_reset sync_cookie source_nat http_template ram_cache_template tcp_proxy_template server_ssl_template conn_reuse_template [] source_ip_persistence_template cookie_persistence_template	send client reset when server selection fails, either enabled(1) or disabled(0) syn cookie, either enabled(1) or disabled(0) source NAT pool name HTTP template name RAM caching template name TCP-Proxy template name Server-SSL template name Connection-Reuse template name one of the following properties: Source-IP-Persistent template name cookie persistence template name	String String Int	0 – 1 0 -`1	
https vport has the following additional properties: send_reset	send client reset when server selection fails,	String String Int		

sync_cookie source_nat http_template ram_cache_template tcp_proxy_template client_ssl_template server_ssl_template conn_reuse_template [] source_ip_persistence_template cookie_persistence_template	either enabled(1) or disabled(0) syn cookie, either enabled(1) or disabled(0) source NAT pool name HTTP template name RAM caching template name TCP-Proxy template name client ssl template Server-SSL template name Connection-Reuse template name one of the following properties: Source-IP-Persistent template name cookie persistence template name Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03	Int String String String String String String String String String String		
Fast http vport has the following additional properties: send_reset sync_cookie source_nat http_template tcp_template conn_reuse_template [] source_ip_persistence_template cookie_persistence_template	send client reset when server selection fails, either enabled(1) or disabled(0) syn cookie, either enabled(1) or disabled(0) source NAT pool name HTTP template name TCP template name Connection-Reuse template name one of the following properties: Source-IP-Persistent template name cookie persistence template name Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03	String String Int Int String String String String String String		
TCP vport has the following additional properties: send_reset ha_connection_mirror direct_server_return sync_cookie source_nat tcp_template conn_reuse_template [] source_ip_persistence_template cookie_persistence_template	send client reset when server selection fails, either enabled(1) or disabled(0) HA connection mirror enabled(1) or disabled (0) direct server return (no destination NAT) enabled (1) or disabled (0) syn cookie, either enabled(1) or disabled(0) source NAT pool name TCP template name Connection-Reuse template name one of the following properties: Source-IP-Persistent template name cookie persistence template name	String String Int Int Int Int String String String String String		

	<p>Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03</p>			
UDP vport has the following additional properties: ha_connection_mirror direct_server_return source_nat udp_template dns_template source_ip_persistence_template	HA connection mirror enabled(1) or disabled (0) direct server return (no destination NAT) enabled (1) or disabled (0) source NAT pool name UDP template name UDP template name one of the following properties: Source-IP-Persistent template name	Int Int String String String String		
RTSP vport has the following additional properties: send_reset ha_connection_mirror direct_server_return sync_cookie tcp_template source_ip_persistence_template	send client reset when server selection fails, either enabled(1) or disabled(0) HA connection mirror enabled(1) or disabled (0) direct server return (no destination NAT) enabled (1) or disabled (0) syn cookie, either enabled(1) or disabled(0) TCP template name Source-IP-Persistent template name	Int Int Int String String		
FTP vport has the following additional properties: send_reset ha_connection_mirror direct_server_return sync_cookie source_nat tcp_template source_ip_persistence_template	send client reset when server selection fails, either enabled(1) or disabled(0) HA connection mirror enabled(1) or disabled (0) direct server return (no destination NAT) enabled (1) or disabled (0) syn cookie, either enabled(1) or disabled(0) sourc nat pool name TCP template name Source-IP-Persistent template name	Int Int Int String String String		

	In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03			
mms vport has the following additional properties: send_reset ha_connection_mirror sync_cookie tcp_temp source_ip_p_temp	send client reset when server selection fails, either enabled(1) or disabled(0) HA connection mirror enabled(1) or disabled (0) Syn Cookie flag, either enabled(1) or disabled(0) TCP template name Source IP persistent template name Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03	Int Int String String	1 or 0	0
SSL-Proxy vport has the following additional properties: send_reset sync_cookie source_nat tcp_proxy_template client_ssl_template server_ssl_template conn_reuse_template source_ip_persistence_template	send client reset when server selection fails, either enabled(1) or disabled(0) syn cookie, either enabled(1) or disabled(0) source NAT pool name TCP-Proxy template name client ssl template Server-SSL template name Connection-Reuse template name Source-IP-Persistent template name Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03	Int Int String String String String String String		
SMPT vport has the following additional properties: send_reset sync_cookie source_nat tcp_proxy_template client_ssl_template smtp_template source_ip_persistence_template	send client reset when server selection fails, either enabled(1) or disabled(0) syn cookie, either enabled(1) or disabled(0) source NAT pool name TCP-Proxy template name client ssl template smtp template Source-IP-Persistent template name Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03	Int Int String String String String String		

SIP vport has the following additional properties: send_reset ha_connection_mirror udp_template source_ip_persistence_template sip_template dns_template	send client reset when server selection fails, either enabled(1) or disabled(0) HA connection mirror enabled(1) or disabled (0) Udp template name Source-IP-Persistent template name SIP template name Dns template name <p>Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03</p>	Int Int String String String String		
SIP-TCP vport has the following additional properties: send_reset sync_cookie source_nat server_ssl_template conn_reuse_template tcp_proxy_template source_ip_persistence_template sip_template	send client reset when server selection fails, either enabled(1) or disabled(0)\syn cookie, either enabled(1) or disabled(0) source NAT pool name Server-SSL template name Connection-Reuse template name TCP-Proxy template name Source-IP-Persistent template name SIP template name <p>Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03</p>	Int Int String String String String String String		
sip-tls vport has the following additional properties: send_reset sync_cookie source_nat client_ssl_template server_ssl_template conn_reuse_template tcp_proxy_template source_ip_persistence_template sip_template	send client reset when server selection fails, either enabled(1) or disabled(0)syn cookie, either enabled(1) or disabled(0) source NAT pool name client ssl template Server-SSL template name Connection-Reuse template name TCP-Proxy template name Source-IP-Persistent template name Sip template name	Int Int String String String String String String String		

	<p>Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03</p>		
tcp-proxy vport has the following additional properties: send_reset sync_cookie source_nat tcp_proxy_template source_ip_persistence_template	<p>send client reset when server selection fails, either enabled(1) or disabled(0) syn cookie, either enabled(1) or disabled(0) source NAT pool name TCP-Proxy template name</p> <p>Source-IP-Persistent template name</p> <p>Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03</p>		
dns-udp vport has the following additional properties: send_reset source_nat udp_template source_ip_persistence_template dns_template	<p>send client reset when server selection fails, either enabled(1) or disabled(0) source NAT pool name</p> <p>UDP template name Source-IP-Persistent template name</p> <p>Dns template name</p> <p>Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03</p>		
diameter-udp vport has the following additional properties: send_reset source_nat udp_template source_ip_persistence_template dns_template	<p>send client reset when server selection fails, either enabled(1) or disabled(0) source NAT pool name</p> <p>UDP template name Source-IP-Persistent template name</p>		

	<p>UDP template name</p> <p>Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03</p>			
tftp vport has the following additional properties:				
ha_connection_mirror direct_server_return source_nat udp_template	<p>HA connection mirror enabled(1) or disabled (0)</p> <p>direct server return (no destination NAT) enabled (1)</p> <p>source NAT pool name</p> <p>UDP template name</p> <p>Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03</p>			
'other' vport has the following additional properties:				
direct_server_return source_nat udp_template l4_template_type [] udp_template tcp_template	<p>direct server return (no destination NAT) enabled (1)</p> <p>source NAT pool name</p> <p>UDP template name</p> <p>TCP (2) or UDP(3)</p> <p>one of the following properties: UDP template name TCP template name</p> <p>Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03</p>			

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Example 1: Update a virtual server with subnet (however, you still must specify the unic index – virtual server name).

Request as:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
- **HTTP Action:**
POST

CLI configuration before aXAPI call:

```
!
slb virtual-server pltest7 15.22.55.5 /24
  disable
  redistribution-flagged
  arp-disable
  template policy 111
  stats-data-disable
  port 80  tcp
    name _15.22.55.5(24)_TCP_80
!
```

1. HTTP Request in URL Format:

```
method=slb.virtual_server.update&virtual_server=name%03pltest7%02subnet%02vport_li
st%02status%03%02arp_status%030%02stats_data%030%02extended_stat%031%02disable_vserver_o
n_condition%030%02redistribution_flagged%031%02ha_group%031%02pbslb_template%03111&subnet
=address%0315.22.55.5%02mask_len%0324&vport_list=11%0212&11=protocol%032%02port%0380&12=p
rotocol%033%02port%0320%02port_acl_id%03100%02aflex_list%02acl_natpool_binding_list&aflex
_list=f1%02f2&f1=DownloadFile1&f2=aflex2&acl_natpool_binding_list=h1%02h2&h1=acl_id%031%0
2nat_pool%03test&h2=acl_id%03100%02nat_pool%03123&format=url
```

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in JSON Format:

```
method=slb.virtual_server.update&format=json
```

Request POST data (JSON Format):

```
{
  "virtual_server": {
    "name": "pltest7",
    "subnet": {
      "address": "15.22.55.5",
      "mask_len": 24
    },
    "status": 0,
    "arp_status": 0,
    "stats_data": 0,
    "extended_stats": 0,
    "disable_vserver_on_condition": 0,
    "redistribution_flagged": 1,
    "ha_group": {
      "status": 0
    },
    "vip_template": "default",
    "pbslb_template": "111",
```

```

"vport_list": [
    {
        "protocol": 3,
        "port": 20,
        "service_group": "",
        "connection_limit": {
            "status": 0,
            "connection_limit": 8000000,
            "connection_limit_action": 0,
            "connection_limit_log": 0
        },
        "default_selection": 1,
        "received_hop": 0,
        "status": 1,
        "stats_data": 1,
        "extended_stats": 0,
        "snat_against_vip": 0,
        "vport_template": "default",
        "port_acl_id": 3,
        "aflex_list": ["DownloadFile1"],
        "ha_connection_mirror": 0,
        "direct_server_return": 0,
        "source_nat": "",
        "udp_template": "",
        "source_ip_persistence_template": "",
        "dns_template": "",
        "pbslb_template": "",
        "acl_natpool_binding_list": [
            {
                "acl_id": 1,
                "nat_pool": "test"
            }
        ]
    },
    {
        "protocol": 2,
        "port": 80,
        "service_group": "",
        "connection_limit": {
            "status": 0,
            "connection_limit": 8000000,
            "connection_limit_action": 0,
            "connection_limit_log": 0
        },
        "default_selection": 1,
        "received_hop": 0,
        "status": 1,
        "stats_data": 1,
        "extended_stats": 0,
        "snat_against_vip": 0,
        "vport_template": "default",
        "aflex_list": [],
        "send_reset": 0,
        "ha_connection_mirror": 0,
        "direct_server_return": 0,
        "sync_cookie": {
            "sync_cookie": 0,
            "sack": 0
        },
        "source_nat": "",
        "tcp_template": "",
        "source_ip_persistence_template": "",
        "pbslb_template": ""
    }
]

```

```

        "acl_natpool_binding_list": []
    }
]
}
}
```

Response as the HTTP body (JSON Format):

```
{
  "response": [
    {
      "status": "OK"
    }
  ]
}
```

CLI configuration after aXAPI call:

```
!
slb virtual-server pltest7 15.22.55.5 /24
  disable
  redistribution-flagged
  arp-disable
  template policy 111
  stats-data-disable
  port 80  tcp
    name _15.22.55.5(24)_TCP_80
  port 20  udp
    name _15.22.55.5(24)_UDP_20
    access-list 1 source-nat-pool test
    access-list 100
    aflex DownloadFile1
!
```

Example 2: Update a virtual server with IPv6 address. (however, you still must specify the unique index – virtual server name).

method=slb.virtual_server.create&format=json

Request POST data (JSON Format):

```
{
  "virtual_server": [
    {
      "name": "pltest7",
      "subnet": [
        {
          "address": "3ff5::1234"
        },
        "status": 0,
        "arp_status": 0,
        "stats_data": 0,
        "extended_stats": 0,
        "disable_vserver_on_condition": 0,
        "redistribution_flagged": 1,
        "ha_group": [
          {
            "status": 0
          }
        ],
        "vip_template": "default",
        "pbslb_template": "111",
        "vport_list": [
          {
            "protocol": 3,
            "port": 20,
            "status": 0
          }
        ]
      ]
    }
  ]
}
```

```

"service_group":"",
"connection_limit":
{
    "status":0,
    "connection_limit":8000000,
    "connection_limit_action":0,
    "connection_limit_log":0
},
"default_selection":1,
"received_hop":0,
"status":1,
"stats_data":1,
"extended_stats":0,
"snat_against_vip":0,
"vport_template":"default",
"port_acl_id":3,
"aflex_list":["tt"],
"ha_connection_mirror":0,
"direct_server_return":0,
"source_nat":"",
"udp_template":"",
"source_ip_persistence_template":"",
"dns_template":"",
"pbslb_template":"",
"acl_natpool_binding_list":[
{
    "acl_id":1,
    "nat_pool":"test"
}
]
},
{
"protocol":2,
"port":80,
"service_group":"",
"connection_limit":
{
    "status":0,
    "connection_limit":8000000,
    "connection_limit_action":0,
    "connection_limit_log":0
},
"default_selection":1,
"received_hop":0,
"status":1,
"stats_data":1,
"extended_stats":0,
"snat_against_vip":0,
"vport_template":"default",
"aflex_list":[],
"send_reset":0,
"ha_connection_mirror":0,
"direct_server_return":0,
"sync_cookie":
{
    "sync_cookie":0,
    "sack":0
},
"source_nat":"",
"tcp_template":"",
"source_ip_persistence_template":"",
"pbslb_template":"",
"acl_natpool_binding_list":[]
}
]
}

```

```
}
```

Menus Privilege

Config Mode >> Service >> SLB >> Virtual Server

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.3.7 “slb.virtual_server.delete” Method

This method is used to delete a virtual server by specifying the virtual server name, or server IP address (IPv4 or IPv6). **This method is required to use the HTTP POST method.**

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.virtual_server.delete	String		
name (**)	Virtual server name	String		
One of the following parameters (address, subnet, acl_id, acl_name) (**)				
address	The address of this virtual server (fetch the virtual server by address, either IPv4 or IPv6)	Ip		
subnet address mask_len	The subnet of this virtual server (fetch the virtual server by subnet address and mask length, only for IPv4)	IPv4 Int	1 – 32	
acl_id	The acl id of this virtual server (fetch the virtual server by acl id, for IPv4 wildcard virtual server only)	Int	1 - 199	
acl_name	The acl name of this virtual server (fetch the virtual server by acl name, for IPv6 wildcard virtual server only)	String		

Note: (*) parameter is required in the API.

(**) Either one of parameters is required.

Response Fields

General XML status response

Example Response

Example1: Delete “http_vip1” by the virtual server name.

Request as:

- **URL:**
[https://\[AX_IP_ADDRESS\]/services/rest/V2/](https://[AX_IP_ADDRESS]/services/rest/V2/)
- **HTTP Action:**
POST

CLI configuration before aXAPI call:

```
! slb virtual-server http_vip1 15.22.55.5 /24
```

```

    disable
    redistribution-flagged
    arp-disable
    template policy 111
    stats-data-disable
    port 80  tcp
        name _15.22.55.5(24)_TCP_80
    port 20  udp
        name _15.22.55.5(24)_UDP_20
        access-list 1 source-nat-pool test
        access-list 100
        aflex DownloadFile1
!

```

1. HTTP Request in URL Format:

```
session_id=123456789&method=slb.virtual_server.delete&name=http_vip1&format=url
```

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in JSON Format:

```
session_id=123456789&method=slb.virtual_server.delete&format=json
```

Request POST data (JSON Format):

```
{
    "name": "http_vip1"
}
```

Response as the HTTP body (JSON Format):

```
{
    "response": {
        "status": "OK"
    }
}
```

CLI configuration after aXAPI call:

```
!
!
!
```

Example 2: Delete “http_vip1” by the virtual server IPv6 address, a:a::b:c.

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST
- HTTP Body
session_id=123456789&method=slb.virtual_server.delete&format=json

Request POST data (JSON Format):

```
{
```

```
    "address": "a:a::b:c"  
}
```

Response as the HTTP body (JSON Format):

```
  "response":  
  {  
    "status" : "OK"  
  }  
}
```

Menus Privilege

[Config Mode](#) >> [Service](#) >> [SLB](#) >> [Virtual Server](#)

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.3.8 “slb.virtual_server.vport.create” Method

This method is used to add a virtual port for a given virtual server. Note that you can also use the `slb.virtual_server.update` method to perform the same operation.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.virtual_server.vport.create	String		
name (*)	virtual server name	String		
[] one of following token (address, subnet, acl_id, acl_name)				
name(*)	the name of the virtual server	String		
address(*)	virtual server IP address, either IPv4 or IPv6	IPv4/v6		
subnet(*)	IPv4 subnet	IPv4/v6		
address	address of subnet	IPv4/v6		
mask_len	the length of this subnet	Int		
acl_id(*)	IPv4 ACL id	Int		
acl_name(*)	IPv6 ACL name	String	1 – 32	1 – 199
vport protocol: (*)	Virtual service object Virtual service type: TCP 2 UDP 3 Other 4 RTSP 5 FTP 6 MMS 7 SIP 8 FAST-HTTP 9 TCP-PROXY 10 HTTP 11 HTTPS 12 SSL-PROXY 13 SMTP 14 SIP-TCP 15 SIP-TLS 16 DIAMETER 17 DNS-UDP 18 TFTP 19	Object Int		
port: (*)	Virtual service port number	Int	0 to 65534	
service_group: (*)	Service group assigned to the virtual service	String		
status:	Virtual service status, either enabled(1) or disabled(0)	Int	1 or 0	
connection_limit:	Connection limit of the virtual service	Int		
status	Connection status	Int	0 or 1	

connection_limit connection_limit_action connection_limit_log default_selection received_hop stats_data extended_stats snat_against_vip vport_template port_acl_id: aFlex_list: aFlex pbslb_template: acl_natpool_binding_list acl_natpool_binding [] acl_id acl_name nat_pool	connection limit value connection limit action, either reset(1) or drop(0) connection limit logging, either enabled(1) or disabled(0) use default server selection when preferred method fails, either enabled(1) or disabled(0) use received hop for response either enabled(1) or disabled(0) stats data, either enabled(1) or disabled(0) extended stats, either enabled(1) or disabled(0) source nat traffic against vip either enabled(1) or disabled(0) virtual server port template name Assigned access-list number aFlex list aFlex name Assigned PBSLB template name acl – natpool binding list acl – natpool binding one of the following properties: IPv4 acl id IPv6 acl name source nat pool name	Int Int Int Int Int Int Int String Int String String	1 - 8000000 0 or 1 0 or 1 0 or 1 0 or 1 0 or 1 0 or 1 1 – 199 1 – 199	
http vport has the following additional properties: send_reset sync_cookie source_nat http_template ram_cache_template tcp_proxy_template server_ssl_template conn_reuse_template [] source_ip_persistence_template cookie_persistence_template	send client reset when server selection fails, either enabled(1) or disabled(0) syn cookie, either enabled(1) or disabled(0) source NAT pool name HTTP template name RAM caching template name TCP-Proxy template name Server-SSL template name Connection-Reuse template name one of the following properties: Source-IP-Persistent template name cookie persistence template name	String String Int Int String String String String String String String String	0 – 1 0 - `1	
https vport has the following additional properties: send_reset sync_cookie source_nat http_template ram_cache_template tcp_proxy_template client_ssl_template server_ssl_template conn_reuse_template [] source_ip_persistence_template cookie_persistence_template	send client reset when server selection fails, either enabled(1) or disabled(0) syn cookie, either enabled(1) or disabled(0) source NAT pool name HTTP template name RAM caching template name TCP-Proxy template name client SSL template Server-SSL template name Connection-Reuse template name one of the following properties: Source-IP-Persistent template name cookie persistence template name	String String Int Int String String String String String String String		

	<p>Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03</p>			
Fast http vport has the following additional properties:	<p>send_reset sync_cookie source_nat http_template tcp_template conn_reuse_template [] source_ip_persistence_template cookie_persistence_template</p> <p>send client reset when server selection fails, either enabled(1) or disabled(0) syn cookie, either enabled(1) or disabled(0) source NAT pool name HTTP template name TCP template name Connection-Reuse template name one of the following properties: Source-IP-Persistent template name Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03</p>	String String Int Int String String String String String String		
TCP vport has the following additional properties:	<p>send_reset ha_connection_mirror direct_server_return sync_cookie source_nat tcp_template conn_reuse_template [] source_ip_persistence_template cookie_persistence_template</p> <p>send client reset when server selection fails, either enabled(1) or disabled(0) HA connection mirror enabled(1) or disabled (0) direct server return (no destination NAT) enabled (1) or disabled (0) syn cookie, either enabled(1) or disabled(0) source NAT pool name TCP template name Connection-Reuse template name one of the following properties: Source-IP-Persistent template name Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03</p>	String String Int Int Int Int String String String String String		
UDP vport has the following additional properties:	<p>ha_connection_mirror direct_server_return source_nat udp_template dns_template source_ip_persistence_template</p> <p>HA connection mirror enabled(1) or disabled (0) direct server return (no destination NAT) enabled (1) or disabled (0) source NAT pool name UDP template name UDP template name one of the following properties: Source-IP-Persistent template name Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03</p>	Int Int String String String String		
RTSP vport has the following additional properties:				

send_reset	send client reset when server selection fails, either enabled(1) or disabled(0)	Int		
ha_connection_mirror	HA connection mirror enabled(1) or disabled (0)	Int		
direct_server_return	direct server return (no destination NAT) enabled (1) or disabled (0)	Int		
sync_cookie	syn cookie, either enabled(1) or disabled(0)	Int		
tcp_template	TCP template name	String		
source_ip_persistence_template	Source-IP-Persistent template name	String		
Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03				
FTP vport has the following additional properties:				
send_reset	send client reset when server selection fails, either enabled(1) or disabled(0)	Int		
ha_connection_mirror	HA connection mirror enabled(1) or disabled (0)	Int		
direct_server_return	direct server return (no destination NAT) enabled (1) or disabled (0)	Int		
sync_cookie	syn cookie, either enabled(1) or disabled(0)	Int		
source_nat	source nat pool name	String		
tcp_template	TCP template name	String		
source_ip_persistence_template	Source-IP-Persistent template name	String		
Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03				
mms vport has the following additional properties:				
send_reset	send client reset when server selection fails, either enabled(1) or disabled(0)	Int		
ha_connection_mirror	HA connection mirror enabled(1) or disabled (0)	Int		
sync_cookie	Syn Cookie flag, either enabled(1) or disabled(0)	Int		
tcp_template	TCP template name	String		
source_ip_p_template	Source IP persistent template name	String		
Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03				
SSL-Proxy vport has the following additional properties:				
send_reset	send client reset when server selection fails, either enabled(1) or disabled(0)	Int		
sync_cookie	syn cookie, either enabled(1) or disabled(0)	Int		
source_nat	source NAT pool name	String		
tcp_proxy_template	TCP-Proxy template name	String		
client_ssl_template	client ssl template	String		
server_ssl_template	Server-SSL template name	String		
conn_reuse_template	Connection-Reuse template name	String		
source_ip_persistence_template	Source-IP-Persistent template name	String		
Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03				
SMTP vport has the following additional properties:				
send_reset	send client reset when server selection fails,	Int		

sync_cookie source_nat tcp_proxy_template client_ssl_template smtp_template source_ip_persistence_template	<p>either enabled(1) or disabled(0) syn cookie, either enabled(1) or disabled(0) source NAT pool name TCP-Proxy template name client ssl template smtp template Source-IP-Persistent template name</p> <p>Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03</p>	Int String String String String String		
SIP vport has the following additional properties: send_reset ha_connection_mirror udp_template source_ip_persistence_template sip_template dns_template	<p>send client reset when server selection fails, either enabled(1) or disabled(0) HA connection mirror enabled(1) or disabled (0) Udp template name Source-IP-Persistent template name</p> <p>SIP template name Dns template name</p> <p>Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03</p>	Int Int String String String String		
SIP-TCP vport has the following additional properties: send_reset sync_cookie source_nat server_ssl_template conn_reuse_template tcp_proxy_template source_ip_persistence_template sip_template	<p>send client reset when server selection fails, either enabled(1) or disabled(0) syn cookie, either enabled(1) or disabled(0) source NAT pool name Server-SSL template name Connection-Reuse template name TCP-Proxy template name Source-IP-Persistent template name</p> <p>SIP template name</p> <p>Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03</p>	Int Int String String String String String		
sip-tls vport has the following additional properties: send_reset sync_cookie source_nat client_ssl_template server_ssl_template conn_reuse_template tcp_proxy_template source_ip_persistence_template sip_template	<p>send client reset when server selection fails, either enabled(1) or disabled(0) syn cookie, either enabled(1) or disabled(0)</p> <p>source NAT pool name</p> <p>client ssl template</p> <p>Server-SSL template name</p> <p>Connection-Reuse template name</p> <p>TCP-Proxy template name</p> <p>Source-IP-Persistent template name</p> <p>Sip template name</p>	Int Int String String String String String String		

	<p>Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03</p>			
tcp-proxy vport has the following additional properties:	<p>send_reset sync_cookie source_nat tcp_proxy_template source_ip_persistence_template</p> <p>send client reset when server selection fails, either enabled(1) or disabled(0) syn cookie, either enabled(1) or disabled(0) source NAT pool name TCP-Proxy template name</p> <p>Source-IP-Persistent template name</p> <p>Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03</p>			
dns-udp vport has the following additional properties:	<p>send_reset source_nat udp_template source_ip_persistence_template dns_template</p> <p>send client reset when server selection fails, either enabled(1) or disabled(0) source NAT pool name</p> <p>UDP template name Source-IP-Persistent template name</p> <p>Dns template name</p> <p>Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03</p>			
diameter-udp vport has the following additional properties:	<p>send_reset source_nat tcp_template source_ip_persistence_template diameter template</p> <p>send client reset when server selection fails, either enabled(1) or disabled(0)</p> <p>source NAT pool name</p> <p>TCP template name</p> <p>Source-IP-Persistent template name</p> <p>Diameter template name</p> <p>Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03</p>			
tftp vport has the following additional properties:				

ha_connection_mirror	HA connection mirror enabled(1) or disabled (0)			
direct_server_return	direct server return (no destination NAT) enabled (1)			
source_nat	source NAT pool name			
udp_template	UDP template name			
	<p>Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03 </p>			
'other' vport has the following additional properties:				
direct_server_return	direct server return (no destination NAT) enabled (1)			
source_nat	source NAT pool name			
udp_template	UDP template name			
l4_template_type	TCP (2) or UDP(3)			
[] udp_template tcp_template	one of the following properties: UDP template name TCP template name			
	<p>Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03 </p>			

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
slb virtual-server AAA 25.25.25.65
  port 80  tcp
    name _25.25.25.65_TCP_80
    service-group http
!
```

1. HTTP Request in URL Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.virtual_server.vport.create&name=AAA&vp
ort=protocol%0317%02port%033000&format=url
```

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
```

```
</response>
```

2. HTTP Request in JSON Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.virtual_server.vport.create&format=json
```

Request POST data (JSON Format):

```
{  
    "name": "AAA",  
    "vport":  
    {  
        "protocol": 17,  
        "port": 3000  
    }  
}
```

Response as the HTTP body (JSON Format):

```
{  
    "response":  
    {  
        "status": "OK"  
    }  
}
```

CLI configuration after aXAPI call:

```
!  
slb virtual-server AAA 25.25.25.65  
    port 80  tcp  
        name _25.25.25.65_TCP_80  
        service-group http  
    port 3000  diameter  
        name _25.25.25.65_Diameter_3000  
!
```

Menus Privilege

Config Mode >> Service >> SLB >> Virtual Server

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.3.9 “slb.virtual_server.vport.update” Method

This method is used to update a virtual port for a given virtual server. You can use slb.virtual_server.update method to perform the same operation.

This method is required to use the HTTP POST method.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.virtual_server.vport.update	String		

name (*)	virtual server name	String		
[] one of following token (address, subnet, acl_id, acl_name)				
name(*)	the name of the virtual server	String		
address(*)	virtual server IP address, either IPv4 or IPv6	IPv4/v6		
subnet(*)	IPv4 subnet	IPv4/v6		
address	address of subnet	Int		
mask_len	the length of this subnet	Int		
acl_id(*)	IPv4 acl id	Int	1 – 32	
acl_name(*)	IPv6 acl name	String	1 – 199	
vport	Virtual service object	Object		
protocol: (*)	Virtual service type:	Int		
	TCP	2		
	UDP	3		
	Other	4		
	RTSP	5		
	FTP	6		
	MMS	7		
	SIP	8		
	FAST-HTTP	9		
	TCP-PROXY	10		
	HTTP	11		
	HTTPS	12		
	SSL-PROXY	13		
	SMTP	14		
	SIP-TCP	15		
	SIP-TLS	16		
	DIAMETER	17		
	DNS-UDP	18		
	TFTP	19		
port: (*)	Virtual service port number	Int	0 to 65534	
service_group: (*)	Service group assigned to the virtual service	String		
status:	Virtual service status, either enabled(1) or disabled(0)	Int	0 or 1	
connection_limit:	Connection limit of the virtual service			
status	Status of connection limit	Int	0 or 1	
connection_limit	connection limit value	Int	1 - 8000000	
connection_limit_action	connection limit action, either reset(1) or drop(0)	Int	0 or 1	
connection_limit_log	connection limit logging, either enabled(1) or disabled(0)	Int	0 or 1	
default_selection				
received_hop	use default server selection when preferred method fails, either enabled(1) or disabled(0)	Int	0 or 1	
stats_data	use received hop for response either enabled(1) or disabled(0)	Int	0 or 1	
extended_stats	stats data, either enabled(1) or disabled(0)	Int	0 or 1	
snat_against_vip	extended stats, either enabled(1) or disabled(0)	Int	0 or 1	
vport_template	source nat traffic against vip either enabled(1) or disabled(0)	Int	0 or 1	
port_acl_id:				
aflex_list:	virtual server port template name	String		
aflex	Assigned access-list number	Int	1 – 199	
pbslb_template:	AFlex list			
acl_natpool_binding_list	Aflex name	String		
 acl_natpool_binding	Assigned PBSLB template name	String		
[]	acl – natpool binding list			
acl_id	acl – natpool binding			
acl_name	one of the following properties:			
nat_pool	IPv4 acl id	Int	1 – 199	
	IPv6 acl name	String		
	source nat pool name	String		
	Note:			
	In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option.			
	^B: ASCII Code 0x02, URL-encode %02			
	^C: ASCII Code 0x03, URL-encode %03			
http vport has the following additional properties:				
send_reset	send client reset when server selection fails, either enabled(1) or disabled(0)	String String Int	0 – 1	

sync_cookie source_nat http_template ram_cache_template tcp_proxy_template server_ssl_template conn_reuse_template [] source_ip_persistence_template cookie_persistence_template	syn cookie, either enabled(1) or disabled(0) source NAT pool name HTTP template name RAM caching template name TCP-Proxy template name Server-SSL template name Connection-Reuse template name one of the following properties: Source-IP-Persistent template name cookie persistence template name	Int String String String String String String String String	0 -`1	
https vport has the following additional properties: send_reset sync_cookie source_nat http_template ram_cache_template tcp_proxy_template client_ssl_template server_ssl_template conn_reuse_template [] source_ip_persistence_template cookie_persistence_template	send client reset when server selection fails, either enabled(1) or disabled(0) syn cookie, either enabled(1) or disabled(0) source NAT pool name HTTP template name RAM caching template name TCP-Proxy template name client ssl template Server-SSL template name Connection-Reuse template name one of the following properties: Source-IP-Persistent template name cookie persistence template name	String String Int Int String String String String String String String String String String		
Fast http vport has the following additional properties: send_reset sync_cookie source_nat http_template tcp_template conn_reuse_template [] source_ip_persistence_template cookie_persistence_template	send client reset when server selection fails, either enabled(1) or disabled(0) syn cookie, either enabled(1) or disabled(0) source NAT pool name HTTP template name TCP template name Connection-Reuse template name one of the following properties: Source-IP-Persistent template name cookie persistence template name	String String Int Int String String String String String String String String		
TCP vport has the following additional properties: send_reset ha_connection_mirror direct_server_return	send client reset when server selection fails, either enabled(1) or disabled(0) HA connection mirror enabled(1) or disabled (0) direct server return (no destination NAT) enabled (1) or	String String Int Int Int		

<pre> sync_cookie source_nat tcp_template conn_reuse_template [] source_ip_persistence_template cookie_persistence_template </pre>	<p>disabled (0) syn cookie, either enabled(1) or disabled(0) source NAT pool name TCP template name Connection-Reuse template name one of the following properties: Source-IP-Persistent template name cookie persistence template name</p> <p>Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03</p>	Int String String String String String		
UDP vport has the following additional properties: <pre> ha_connection_mirror direct_server_return source_nat udp_template dns_template source_ip_persistence_template </pre>	<p>HA connection mirror enabled(1) or disabled (0) direct server return (no destination NAT) enabled (1) or disabled (0) source NAT pool name UDP template name UDP template name one of the following properties: Source-IP-Persistent template name</p> <p>Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03</p>	Int Int String String String String		
RTSP vport has the following additional properties: <pre> send_reset ha_connection_mirror direct_server_return sync_cookie tcp_template source_ip_persistence_template </pre>	<p>send client reset when server selection fails, either enabled(1) or disabled(0) HA connection mirror enabled(1) or disabled (0) direct server return (no destination NAT) enabled (1) or disabled (0) syn cookie, either enabled(1) or disabled(0) TCP template name Source-IP-Persistent template name</p> <p>Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03</p>	Int Int Int Int String String		
FTP vport has the following additional properties: <pre> send_reset ha_connection_mirror direct_server_return sync_cookie source_nat tcp_template source_ip_persistence_template </pre>	<p>send client reset when server selection fails, either enabled(1) or disabled(0) HA connection mirror enabled(1) or disabled (0) direct server return (no destination NAT) enabled (1) or disabled (0) syn cookie, either enabled(1) or disabled(0) source nat pool name TCP template name Source-IP-Persistent template name</p> <p>Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03</p>	Int Int Int Int String String String		

mms vport has the following additional properties: <code>send_reset</code> <code>ha_connection_mirror</code> <code>sync_cookie</code> <code>tcp_temp</code> <code>source_ip_p_temp</code>	send client reset when server selection fails, either enabled(1) or disabled(0) HA connection mirror enabled(1) or disabled (0) Syn Cookie flag, either enabled(1) or disabled(0) TCP template name Source IP persistent template name Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03	Int Int Int String String	1 or 0	0
SSL-Proxy vport has the following additional properties: <code>send_reset</code> <code>sync_cookie</code> <code>source_nat</code> <code>tcp_proxy_template</code> <code>client_ssl_template</code> <code>server_ssl_template</code> <code>conn_reuse_template</code> <code>source_ip_persistence_template</code>	send client reset when server selection fails, either enabled(1) or disabled(0) syn cookie, either enabled(1) or disabled(0) source NAT pool name TCP-Proxy template name client ssl template Server-SSL template name Connection-Reuse template name Source-IP-Persistent template name Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03	Int Int String String String String String String		
SMPT vport has the following additional properties: <code>send_reset</code> <code>sync_cookie</code> <code>source_nat</code> <code>tcp_proxy_template</code> <code>client_ssl_template</code> <code>smtp_template</code> <code>source_ip_persistence_template</code>	send client reset when server selection fails, either enabled(1) or disabled(0) syn cookie, either enabled(1) or disabled(0) source NAT pool name TCP-Proxy template name client ssl template smtp template Source-IP-Persistent template name Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03	Int Int String String String String String		
SIP vport has the following additional properties: <code>send_reset</code> <code>ha_connection_mirror</code> <code>udp_temp</code> <code>source_ip_persistence_template</code> <code>sip_temp</code> <code>dns_temp</code>	send client reset when server selection fails, either enabled(1) or disabled(0) HA connection mirror enabled(1) or disabled (0) Udp template name Source-IP-Persistent template name SIP template name Dns template name Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03	Int Int String String		
SIP-TCP vport has the following additional properties: <code>send_reset</code>	send client reset when server selection fails, either enabled(1) or disabled(0)\	Int		

sync_cookie source_nat server_ssl_template conn_reuse_template tcp_proxy_template source_ip_persistence_template sip_template	syn cookie, either enabled(1) or disabled(0) source NAT pool name Server-SSL template name Connection-Reuse template name TCP-Proxy template name Source-IP-Persistent template name SIP template name	Int String String String String String String		
Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03				
sip-tls vport has the following additional properties: send_reset sync_cookie source_nat client_ssl_template server_ssl_template conn_reuse_template tcp_proxy_template source_ip_persistence_template sip_template	send client reset when server selection fails, either enabled(1) or disabled(0) syn cookie, either enabled(1) or disabled(0) source NAT pool name client ssl template Server-SSL template name Connection-Reuse template name TCP-Proxy template name Source-IP-Persistent template name Sip template name	Int Int String String String String String String String		
Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03				
tcp-proxy vport has the following additional properties: send_reset sync_cookie source_nat tcp_proxy_template source_ip_persistence_template	send client reset when server selection fails, either enabled(1) or disabled(0) syn cookie, either enabled(1) or disabled(0) source NAT pool name TCP-Proxy template name Source-IP-Persistent template name			
Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03				
dns-udp vport has the following additional properties: send_reset source_nat udp_template source_ip_persistence_template dns_template	send client reset when server selection fails, either enabled(1) or disabled(0) source NAT pool name UDP template name Source-IP-Persistent template name			

	Dns template name Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03			
diameter-udp vport has the following additional properties: send_reset source_nat tcp_template source_ip_persistence_template diameter template	 send client reset when server selection fails, either enabled(1) or disabled(0) source NAT pool name TCP template name Source-IP-Persistent template name Diameter template name Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03			
ftp vport has the following additional properties: ha_connection_mirror direct_server_return source_nat udp_template	 HA connection mirror enabled(1) or disabled (0) direct server return (no destination NAT) enabled (1) source NAT pool name UDP template name Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03			
'other' vport has the following additional properties: direct_server_return source_nat udp_template l4_template_type [] udp_template tcp_template	 direct server return (no destination NAT) enabled (1) source NAT pool name UDP template name TCP (2) or UDP(3) one of the following properties: UDP template name TCP template name Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option.			

	^B: ASCII Code 0x02, URL-encode %02			
	^C: ASCII Code 0x03, URL-encode %03			

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
slb virtual-server AAA 25.25.25.65
    port 80  tcp
        name _25.25.25.65_TCP_80
        service-group http
    port 3000  diameter
        name _25.25.25.65_Diameter_3000
!
```

1. HTTP Request in URL Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.virtual_server.vport.update&name=AAA&vp
ort=protocol%0317%02port%033000%02status%030&format=url
```

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in JSON Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.virtual_server.vport.update&format=json
```

Request POST data (JSON Format):

```
{
  "name": "AAA",
  "vport": {
    "protocol": 17,
    "port": 3000,
    "status": 0
  }
}
```

Response as the HTTP body (JSON Format):

```
{
  "response": [
    {
      "status": "OK"
    }
  ]
}
```

CLI configuration after aXAPI call:

```
!  
slb virtual-server AAA 25.25.25.65  
  port 80  tcp  
    name _25.25.25.65_TCP_80  
    service-group http  
  port 3000  diameter  
    name _25.25.25.65_Diameter_3000  
    disable  
!
```

Menus Privilege

Config Mode >> Service >> SLB >> Virtual Server

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.3.10 “slb.virtual_server.vport.delete” Method

This method is used to delete a virtual port for a specific virtual server. You can use `slb.virtual_server.update` method to perform the same operation.

This method is required to use the HTTP POST method.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.virtual_server.vport.delete	String		
[] One of the following token (address, subnet, acl_id, acl_name): name(*) address(*) subnet(*) address mask_len acl_id(*) acl_name(*)	The name of the virtual server virtual server IP address, either IPv4 or IPv6 IPv4 subnet address of subnet the length of this subnet IPv4 acl id IPv6 acl name	String IPv4/v6 IPv4/v6 Int Int String	1 – 32 1 – 199	
vport(*) protocol: (*)	virtual service object virtual service type: TCP UDP Other RTSP FTP MMS SIP FAST-HTTP TCP-PROXY HTTP HTTPS SSL-PROSY SMTP SIP-TCP SIP-TLS DIAMETER DNS-UDP TFTP	Object Int		
port: (*)	virtual service port number	Int	0 to 65534	

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
slb virtual-server AAA 25.25.25.65
  port 80  tcp
    name _25.25.25.65_TCP_80
    service-group http
  port 3000  diameter
    name _25.25.25.65_Diameter_3000
!
```

1. HTTP Request in URL Format:

```
session_id=123456789method=slb.virtual_server.vport.delete&name=AAA&vport=protocol
%0317%02port%033000&format=url
```

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in JSON Format:

```
session_id=123456789method=slb.virtual_server.vport.delete&format=json
```

Request POST data (JSON Format):

```
{
  "name": "AAA",
  "vport": {
    "protocol": 17,
    "port": 3000
  }
}
```

Response as the HTTP body (JSON Format):

```
{
  "response": {
    "status": "OK"
  }
}
```

CLI configuration after aXAPI call:

```
!
slb virtual-server AAA 25.25.25.65
  port 80  tcp
    name _25.25.25.65_TCP_80
    service-group http
!
```

Menus Privilege

Config Mode >> Service >> SLB >> Virtual Server

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.4 Virtual Server Template

7.4.1 “slb.template.vip.getAll” Method

This method is used to get configuration information for virtual IP (VIP) templates configured on the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.template.vip.getAll	String		

Note: (*) parameter is required in the API.

Response Fields

vip_template_list	XML tag for the collection of virtual server templates
vip_template	XML tag for virtual server template
name	virtual server template name
subnet_gratuitous_arp	subnet gratuitous ARP
conn_limit	XML tag for virtual server template connection limit
status	connection limit status, disabled(0) or enabled(1)
num	only when conn_limit is enabled(1)
action	drop(0) or reset(1), only when conn_limit is enabled(1)
logging	disabled(0) or enabled(1), only when conn_limit is 1
conn_rate_limit	XML tag for virtual server template connection rate limit
status	connection rate limit status, disabled(0) or enabled(1)
num	only when conn_rate_limit is enabled(1)
sample_per	100ms(0) or 1 second(1), only when conn_rate_limit is 1
action	drop(0) or reset(1), only when conn_rate_limit is enabled(1)
logging	disabled(0) or enabled(1), only when conn_rate_limit is 1
icmp_rate_limit	XML tag for virtual server template ICMP rate limit
status	ICMP rate limit status, disabled(0) or enabled(1)
normal_rate	only when icmp_rate_limit is enabled(1)
lockup_status	disabled(0) or enabled(1), only when icmp_rate_limit is 1
lockup_rate	only when icmp_rate_limit is enabled(1)
lockup_period	(second) only when icmp_rate_limit is enabled(1)

Example Response

Request as:

- URL:

- [https://\[AX_IP_ADDRESS\]/services/rest/V2/](https://[AX_IP_ADDRESS]/services/rest/V2/)
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
slb template virtual-server vip_template_name1
  conn-limit 800
  conn-rate-limit 100 reset
  icmp-rate-limit 1000 lockup 10000 1000
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxxxx&method=slb.template.vip.getAll

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
  <vip_template_list>
    <vip_template>
      <name>default</name>
      <subnet_gratuitous_arp>0</subnet_gratuitous_arp>
      <conn_limit>
        <status>1</status>
        <num>8000000</num>
        <action>0</action>
        <logging>0</logging>
      </conn_limit>
      <conn_rate_limit>
        <status>0</status>
      </conn_rate_limit>
      <icmp_rate_limit>
        <status>1</status>
        <normal_rate>1000</normal_rate>
        <lockup_status>0</lockup_status>
      </icmp_rate_limit>
    </vip_template>
    <vip_template>
      <name>vip_template_name1</name>
      <subnet_gratuitous_arp>1</subnet_gratuitous_arp>
      <conn_limit>
        <status>1</status>
        <num>800</num>
        <action>0</action>
        <logging>1</logging>
      </conn_limit>
      <conn_rate_limit>
        <status>1</status>
        <num>100</num>
        <sample_per>1</sample_per>
        <action>0</action>
        <logging>1</logging>
      </conn_rate_limit>
      <icmp_rate_limit>
        <status>1</status>
        <normal_rate>1000</normal_rate>
        <lockup_status>1</lockup_status>
        <lockup_rate>10000</lockup_rate>
        <lockup_period>1000</lockup_period>
      </icmp_rate_limit>
    </vip_template>
  </vip_template_list>
</response>
```

Menus Privilege

Config Mode >> Service >> SLB >> Template >> Virtual Server

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.4.2 “slb.template.vip.search” Method

This method is used to get virtual IP (VIP) template information for a given VIP template name.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.template.vip.search	String		
name	virtual server template name	String		

Note: (*) parameter is required in the API.

Response Fields

vip_template	XML tag for virtual server template
name	virtual server template name
subnet_gratuitous_arp	subnet gratuitous ARP
conn_limit	XML tag for virtual server template connection limit
status	connection limit status, disabled(0) or enabled(1)
num	only when conn_limit is enabled(1)
action	drop(0) or reset(1), only when conn_limit is enabled(1)
logging	disabled(0) or enabled(1), only when conn_limit is 1
conn_rate_limit	XML tag for virtual server template connection rate limit
status	connection rate limit status, disabled(0) or enabled(1)
num	only when conn_rate_limit is enabled(1)
sample_per	100ms(0) or 1 second(1), only when conn_rate_limit is 1
action	drop(0) or reset(1), only when conn_rate_limit is enabled(1)
logging	disabled(0) or enabled(1), only when conn_rate_limit is 1
icmp_rate_limit	XML tag for virtual server template ICMP rate limit
status	ICMP rate limit status, disabled(0) or enabled(1)
normal_rate	only when icmp_rate_limit is enabled(1)
lockup_status	disabled(0) or enabled(1), only when icmp_rate_limit is 1
lockup_rate	only when icmp_rate_limit is enabled(1)
lockup_period	(second) only when icmp_rate_limit is enabled(1)

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!  
slb template virtual-server vip template name1
```

```

conn-limit 800
conn-rate-limit 100 reset
icmp-rate-limit 1000 lockup 10000 1000
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxxxx&method=slb.template.vip.search&name=vip_template_name1

Response as the HTTP body:

```

<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
    <vip_template>
        <name>vip_template_name1</name>
        <subnet_gratuitous_arp>1</subnet_gratuitous_arp>
        <conn_limit>
            <status>1</status>
            <num>800</num>
            <action>0</action>
            <logging>1</logging>
        </conn_limit>
        <conn_rate_limit>
            <status>1</status>
            <num>100</num>
            <sample_per>1</sample_per>
            <action>0</action>
            <logging>1</logging>
        </conn_rate_limit>
        <icmp_rate_limit>
            <status>1</status>
            <normal_rate>1000</normal_rate>
            <lockup_status>1</lockup_status>
            <lockup_rate>10000</lockup_rate>
            <lockup_period>1000</lockup_period>
        </icmp_rate_limit>
    </vip_template>
</response>
```

Menus Privilege

Config Mode >> Service >> SLB >> Template >> Virtual Server

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.4.3 “slb.template.vip.create” Method

This method is used to create a Virtual IP (VIP) template.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.template.vip.create	String		
name (*)	virtual server template name	String		
conn_limit_status	connection limit status, disabled(0) or enabled(1)	Int	0 or 1	0
conn_limit	connection limit, only when conn_limit_status is enabled(1)	Int	1 - 8000000	0
conn_limit_act	drop(0) or reset(1), only when conn_limit_status is enabled(1)	Int	0 or 1	0
conn_limit_log	disabled(0) or enabled(1), only when conn_limit_status is enabled(1)	Int	0 or 1	0
conn_rate_limit_status	connection rate limit status, disabled(0) or	Int	0 or 1	0

	enabled(1)			
conn_rate_limit	connection rate limit, only when conn_rate_limit_status is enabled(1)	Int	1 - 1048575	1000
conn_rate_limit_sample_per	100ms(0) or 1 second(1), only when conn_limit_rate_status is enabled(1)	Int	0 or 1	1
conn_rate_limit_act	drop(0) or reset(1), only when conn_rate_limit_status is enable(1)	Int	0 or 1	0
conn_rate_limit_log	disabled(0) or enabled(1) only when conn_rate_limit_status is enable(1)	Int	0 or 1	0
icmp_rate_limit_status	icmp rate limit status, disabled(0) or enabled(1)	Int	0 or 1	0
icmp_rate_limit_nor	ICMP rate limit normal rate, only when icmp_rate_limit_status is enabled(1)	Int	1 - 65535	1000
icmp_rate_limit_lockup_status	icmp rate limit lockup status, disabled(0) or enabled(1) only when icmp_rate_limit_status is enabled(1)	Int	0 or 1	0
icmp_rate_limit_lockup	ICMP rate limit lockup rate only when icmp_rate_limit_status is enabled(1)	Int	1 - 65535	10000
icmp_rate_limit_period	ICMP rate limit lockup period only when icmp_rate_limit_status is enabled(1)	Int	1 - 16383	1000
subnet_gratuitous_arp	subnet gratuitous ARP status, disabled(0) or enabled(1)	Int	0 or 1	0

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration after aXAPI call:

```
!
slb template virtual-server vip_template_name1
  conn-limit 1000000
  conn-rate-limit 1000
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxxx&method=slb.template.vip.create&name=vip_template_name1
&conn_limit_status=1&conn_limit=1000000&conn_limit_act=0&conn_limit_log=1&conn_rate_limit_status=1&conn_rate_limit=1000&conn_rate_limit_act=0&icmp_rate_limit_status=0
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> SLB >> Template >> Virtual Server

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.4.4 “slb.template.vip.update” Method

This method is used to update a virtual IP (VIP) template information for a given VIP template name.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.template.vip.update	String		
name (*)	virtual server template name	String		
conn_limit_status	connection limit status, disabled(0) or enabled(1)	Int	0 or 1	
conn_limit	connection limit, only when conn_limit_status is enabled(1)	Int	1 - 8000000	
conn_limit_act	drop(0) or reset(1), only when conn_limit_status is enabled(1)	Int	0 or 1	
conn_limit_log	disabled(0) or enabled(1), only conn_limit_status is enabled(1)	Int	0 or 1	
conn_rate_limit_status	connection rate limit status, disabled(0) or enabled(1)	Int	0 or 1	
conn_rate_limit	connection rate limit, only conn_rate_limit_status is enabled(1)	Int	1 - 1048575	
conn_rate_limit_sample_per	100ms(0) or 1 second(1), only when conn_rate_limit_status is enabled(1)	Int	0 or 1	
conn_rate_limit_act	drop(0) or reset(1), only when conn_rate_limit_status is enabled(1)	Int	0 or 1	
conn_rate_limit_log	disabled(0) or enabled(1) only conn_rate_limit_status is enabled(1)	Int	0 or 1	
icmp_rate_limit_status	icmp rate limit status, disabled(0) or enabled(1)	Int	0 or 1	
icmp_rate_limit_nor	icmp rate limit normal rate, only when icmp_rate_limit_status is enabled(1)	Int	1 - 65535	
icmp_rate_limit_lockup_status	icmp rate limit lockup status, disabled(0) or enabled(1) only when icmp_rate_limit_status is enabled(1)	Int	0 or 1	
icmp_rate_limit_lockup	icmp rate limit lockup rate only when icmp_rate_limit_status is enabled(1)	Int	1 - 65535	
icmp_rate_limit_period	icmp rate limit lockup period only when icmp_rate_limit_status is enabled(1)	Int	1 - 16383	
subnet_gratuitous_arp	subnet gratuitous ARP status, disabled(0) or enabled(1)	Int	0 or 1	

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
slb template virtual-server vip_template_name1
  conn-limit 800
  conn-rate-limit 100 reset
  icmp-rate-limit 1000 lockup 10000 1000
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.template.vip.update&name=vip_template_name1&conn_limit_status=1&conn_limit=1000000&conn_limit_act=0&conn_limit_log=1&conn_rate_limit_status=1&conn_rate_limit=1000&conn_rate_limit_act=0&icmp_rate_limit_status=0
```

CLI configuration after aXAPI call:

```
!
slb template virtual-server vip_template_name1
  conn-limit 1000000
  conn-rate-limit 1000
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> SLB >> Template >> Virtual Server

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.4.5 “slb.template.vip.delete” Method

This method is used to delete a virtual server template by specifying the template name.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.template.vip.delete	String		
name (*)	virtual server template name	String		

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
- **HTTP Action:**
POST

CLI configuration before aXAPI call:

```
!
slb template virtual-server vip_template_name1
  conn-limit 1000000
  conn-rate-limit 1000
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.template.vip.delete&name=vip_template_name1
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> SLB >> Template >> Virtual Server

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.5 Virtual Server Port Template

7.5.1 “slb.template.vport.getAll” Method

This method is used to get configuration information for all virtual port templates configured on the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.template.vport.getAll	String		

Note: (*) parameter is required in the API.

Response Fields

vport_template_list	XML tag for the collection of virtual server port templates
vport_template	XML tag for virtual server port template
name	virtual server port template name
aflow	disabled(0) or enabled(1)
reset_unk_conn	reset unknown connection status, disabled(0) or enabled(1)
ignore_tcp_msl	ignore TCP MSL, disabled(0) or enabled(1)
conn_limit	XML tag for virtual server port template connection limit
status	connection limit status, disabled(0) or enabled(1)
num	only when conn_limit is enabled(1)
action	drop(0) or reset(1), only when conn_limit is enabled(1)
logging	disabled(0) or enabled(1), only conn_limit is enabled(1)
conn_rate_limit	XML tag for vport template connection rate limit
status	connection rate limit status, disabled(0) or enabled(1)
num	only when conn_rate_limit is enabled(1)
sample_per	100ms(0) or 1 second(1), only conn_rate_limit is enabled(1)
action	drop(0) or reset(1), only when conn_rate_limit is enabled(1)
logging	disabled(0) or enabled(1), only conn_rate_limit is enable(1)

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
slb template virtual-port vport_template_name1
  conn-limit 8000000
  conn-rate-limit 1000
  reset-unknown-conn
  ignore-tcp-msl
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxxxx&method=slb.template.vport.getAll

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
  <vport_template_list>
    <vport_template>
      <name>default</name>
      <aflow>0</aflow>
      <reset_unk_conn>0</reset_unk_conn>
      <ignore_tcp_msl>0</ignore_tcp_msl>
      <conn_limit>
        <status>0</status>
      </conn_limit>
      <conn_rate_limit>
        <status>0</status>
      </conn_rate_limit>
    </vport_template>
    <vport_template>
      <name>vport_template_name1</name>
      <aflow>0</aflow>
      <reset_unk_conn>1</reset_unk_conn>
      <ignore_tcp_msl>1</ignore_tcp_msl>
      <conn_limit>
        <status>1</status>
        <num>8000000</num>
        <action>0</action>
        <logging>1</logging>
      </conn_limit>
      <conn_rate_limit>
        <status>1</status>
        <num>1000</num>
        <sample_per>1</sample_per>
        <action>0</action>
        <logging>1</logging>
      </conn_rate_limit>
    </vport_template>
  </vport_template_list>
</response>
```

Menus Privilege

Config Mode >> Service >> SLB >> Template >> Virtual Server Port
 Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.5.2 “slb.template.vport.search” Method

This method is used to get virtual port (vport) template information for a given vport template name.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.template.vport.search	String		
name (*)	virtual server port template name	String		

Note: (*) parameter is required in the API.

Response Fields

vport_template	XML tag for virtual server port template
name	virtual server port template name
aflow	disabled(0) or enabled(1)
reset_unk_conn	reset unknown connection status, disabled(0) or enabled(1)
ignore_tcp_msl	ignore TCP MSL, disabled(0) or enabled(1)
conn_limit	XML tag for virtual server port template connection limit
status	connection limit status, disabled(0) or enabled(1)
num	only when conn_limit is enabled(1)
action	drop(0) or reset(1), only when conn_limit is enabled(1)
logging	disabled(0) or enabled(1), only conn_limit is enabled(1)
conn_rate_limit	XML tag for vport template connection rate limit
status	connection rate limit status, disabled(0) or enabled(1)
num	only when conn_rate_limit is enabled(1)
sample_per	100ms(0) or 1 second(1), only conn_rate_limit is enabled(1)
action	drop(0) or reset(1), only when conn_rate_limit is enabled(1)
logging	disabled(0) or enabled(1), only conn_rate_limit is enable(1)

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
slb template virtual-port vport_template_name1
    conn-limit 8000000
    conn-rate-limit 1000
    reset-unknown-conn
    ignore-tcp-msl
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.template.vport.search&name=vport_template_name1

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
    <vport_template>
        <name>vport_template_name1</name>
        <aflow>0</aflow>
```

```

<reset_unk_conn>1</reset_unk_conn>
<ignore_tcp_msl>1</ignore_tcp_msl>
<conn_limit>
    <status>1</status>
    <num>8000000</num>
    <action>0</action>
    <logging>1</logging>
</conn_limit>
<conn_rate_limit>
    <status>1</status>
    <num>1000</num>
    <sample_per>1</sample_per>
    <action>0</action>
    <logging>1</logging>
</conn_rate_limit>
</vport_template>
</response>

```

Menus Privilege

Config Mode >> Service >> SLB >> Template >> Virtual Server Port

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.5.3 “slb.template.vport.create” Method

This method is used to create a virtual port (vport) template.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.template.vport.create	String		
name (*)	virtual server port template name	String		
conn_limit_status	connection limit status, disabled(0) or enabled(1)	Int	0 or 1	0
conn_limit	connection limit, only when conn_limit_status is enabled(1)	Int	1 – 8000000 0	800000
conn_limit_act	drop(0) or reset(1), only when conn_limit_status is enabled(1)	Int	0 or 1	0
conn_limit_log	disabled(0) or enabled(1), only conn_limit_status is enabled(1)	Int	0 or 1	0
conn_rate_limit_status	connection rate limit status, disabled(0) or enabled(1)	Int	0 or 1	0
conn_rate_limit	connection rate limit, only conn_rate_limit_status is enabled(1)	Int	1 – 1048575	1000
conn_rate_limit_sample_per	100ms(0) or 1 second(1), only when conn_rate_limit_status is enabled(1)	Int	0 or 1	0
conn_rate_limit_act	drop(0) or reset(1), only when conn_rate_limit_status is enable(1)	Int	0 or 1	0
conn_rate_limit_log	disabled(0) or enabled(1) only conn_rate_limit_status is enable(1)	Int	0 or 1	0
aflow	aflow status, disabled(0) or enabled(1)	Int	0 or 1	0
reset_unk_conn	Reset unknown connection status, disabled(0) or enabled(1)	Int	0 or 1	0
ignore_tcp_msl	Ignore TCP MSL, disabled(0) or enabled(1)	Int	0 or 1	0

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
- **HTTP Action:**
POST

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxxxx&method=slb.template.vport.create&name=vport_template_n
ame1&conn_limit_status=1&conn_limit=1000000&conn_limit_act=0&conn_limit_log=1&conn_rat
e_limit_status=1&conn_rate_limit=1000&conn_rate_limit_act=0
```

CLI configuration after aXAPI call:

```
!
slb template virtual-port vport_template_name1
  conn-limit 1000000
  conn-rate-limit 1000
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> SLB >> Template >> Virtual Server Port
 Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.5.4 “slb.template.vport.update” Method

This method is used to update one or more parameters in a virtual port template.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.template.vport.update	String		
name (*)	virtual server port template name	String		
conn_limit_status	connection limit status, disabled(0) or enabled(1)	Int	0 or 1	
conn_limit	connection limit, only when conn_limit_status is enabled(1)	Int	1 – 8000000	
conn_limit_act	drop(0) or reset(1), only when conn_limit_status is enabled(1)	Int	0 or 1	
conn_limit_log	disabled(0) or enabled(1), only conn_limit_status is enabled(1)	Int	0 or 1	
conn_rate_limit_status	connection rate limit status, disabled(0) or enabled(1)	Int	0 or 1	
conn_rate_limit	connection rate limit, only conn_rate_limit_status is enabled(1)	Int	1 – 1048575	
conn_rate_limit_sample_per	100ms(0) or 1 second(1), only when conn_limit_rate_status is enabled(1)	Int	0 or 1	
conn_rate_limit_act	drop(0) or reset(1), only when conn_rate_limit_status is enabled(1)	Int	0 or 1	

conn_rate_limit_log	disabled(0) or enabled(1) only conn_rate_limit_status is enable(1)	Int	0 or 1	
aflow	aflow status, disabled(0) or enabled(1)	Int	0 or 1	0
reset_unk_conn	Reset unknown connection status, disabled(0) or enabled(1)	Int	0 or 1	
ignore_tcp_msl	Ignore TCP MSL, disabled(0) or enabled(1)	Int	0 or 1	0

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
- **HTTP Action:**
POST

CLI configuration before aXAPI call:

```
!
slb template virtual-port vport_template_name1
  conn-limit 8000000
  conn-rate-limit 1000
  reset-known-conn
  ignore-tcp-msl
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.template.vport.update&name=vport_template_name1&
conn_limit_status=1&conn_limit=1000000&conn_limit_act=0&conn_limit_log=1&conn_rate_limit_
status=1&conn_rate_limit=1000&conn_rate_limit_act=0
```

CLI configuration after aXAPI call:

```
!
slb template virtual-port vport_template_name1
  conn-limit 1000000
  conn-rate-limit 1000
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> SLB >> Template >> Virtual Server Port

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.5.5 “slb.template.vport.delete” Method

This method is used to delete a virtual server port template.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.template.vport.delete	String		
name (*)	virtual server port template name	String		

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.template.vport.delete&name=vport_template_name1

CLI configuration before aXAPI call:

```
!
slb template virtual-port vport_template_name1
  conn-limit 8000000
  conn-rate-limit 1000
  reset-unknown-conn
  ignore-tcp-msl
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> SLB >> Template >> Virtual Server Port

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.6 Server Template

7.6.1 “slb.template.server.getAll” Method

This method is used to get configuration information for all server templates configured on the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.template.server.getAll	String		

Note: (*) parameter is required in the API.

Response Fields

server_template_list	XML tag for the collection of server templates
server_template	XML tag for server template
name	server template name
health_monitor	health monitor name
dns_que_int	dns querry interval (minute)
min_ttl_radio	minimum TTL radio
max_dyn_srv	maximum dynamic server number
pre_dyn_srv	prefix of dynamic server
 conn_limit	
status	XML tag for server template connection limit
num	connection limit status, disabled(0) or enabled(1)
conn_resume	connection limit, only when conn_limit is enabled(1)
logging	only when conn_limit is enabled(1)
	disabled(0) or enabled(1), only conn_limit is enabled(1)
 conn_rate_limit	
status	XML tag for server template connection rate limit
num	connection rate limit status, disabled(0) or enabled(1)
sample_per	only when conn_rate_limit is enabled(1)
logging	100ms(0) or 1 second(1), only conn_rate_limit is enabled(1)
	disabled(0) or enabled(1), only conn_rate_limit is enable(1)
 slow_start	
status	XML tag for server template slow start
from	disabled(0) or enabled(1)
by_mul	slow start from
by_add	slow start by multiplying
every	slow start by adding
till	slow start every number
	slow start till number

Example Response

Request as:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
- **HTTP Action:**
GET

CLI configuration before aXAPI call:

```
!
slb template server server_template_name1
    health-check ping
    slow-start
    conn-limit 8000000 resume 222
    conn-rate-limit 1000
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.template.server.getAll
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
    <server_template_list>
        <server_template>
            <name>default</name>
```

```

<health_monitor>(default)</health_monitor>
<dns_que_int>10</dns_que_int>
<min_ttl_radio>2</min_ttl_radio>
<max_dyn_srv>255</max_dyn_srv>
<pre_dyn_srv>DRS</pre_dyn_srv>
<conn_limit>
    <status>0</status>
</conn_limit>
<conn_rate_limit>
    <status>0</status>
</conn_rate_limit>
<slow_start>
    <status>0</status>
</slow_start>
</server_template>
<server_template>
    <name>server_template_name1</name>
    <health_monitor>ping</health_monitor>
    <dns_que_int>10</dns_que_int>
    <min_ttl_radio>2</min_ttl_radio>
    <max_dyn_srv>255</max_dyn_srv>
    <pre_dyn_srv>DRS</pre_dyn_srv>
    <conn_limit>
        <status>1</status>
        <num>8000000</num>
        <conn_resume>222</conn_resume>
        <logging>1</logging>
    </conn_limit>
    <conn_rate_limit>
        <status>1</status>
        <num>1000</num>
        <sample_per>1</sample_per>
        <action>0</action>
        <logging>1</logging>
    </conn_rate_limit>
    <slow_start>
        <status>1</status>
        <from>128</from>
        <by_mul>2</by_mul>
        <every>10</every>
        <till>4096</till>
    </slow_start>
</server_template>
</server_template_list>
</response>

```

Menus Privilege

Config Mode >> Service >> SLB >> Template >> Server

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.6.2 “slb.template.server.search” Method

This method is used to get server template information for a given server template name.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.template.server.search	String		
name (*)	server template name	String		

Note: (*) parameter is required in the API.

Response Fields

server_template	XML tag for server template
name	server template name
health_monitor	health monitor name
dns_que_int	dns querry interval (minute)
min_ttl_radio	minimum TTL radio
max_dyn_srv	maximum dynamic server number
pre_dyn_srv	prefix of dynamic server
conn_limit	XML tag for server template connection limit
status	connection limit status, disabled(0) or enabled(1)
num	connection limit, only when conn_limit is enabled(1)
conn_resume	only when conn_limit is enabled(1)
logging	disabled(0) or enabled(1), only conn_limit is enabled(1)
conn_rate_limit	XML tag for server template connection rate limit
status	connection rate limit status, disabled(0) or enabled(1)
num	only when conn_rate_limit is enabled(1)
sample_per	100ms(0) or 1 second(1), only conn_rate_limit is enabled(1)
logging	disabled(0) or enabled(1), only conn_rate_limit is enable(1)
slow_start	XML tag for server template slow start
status	disabled(0) or enabled(1)
from	slow start from
by_mul	slow start by multiplying
by_add	slow start by adding
every	slow start every number
till	slow start till number

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
slb template server server_template_name1
    health-check ping
    slow-start
    conn-limit 8000000 resume 222
    conn-rate-limit 1000
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxxxx&method=slb.template.server.search&name=server_template_na
me1

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
    <server_template>
        <name>server_template_name1</name>
        <health_monitor>ping</health_monitor>
        <dns_que_int>10</dns_que_int>
        <min_ttl_radio>2</min_ttl_radio>
```

```

<max_dyn_srv>255</max_dyn_srv>
<pre_dyn_srv>DRS</pre_dyn_srv>
<conn_limit>
    <status>1</status>
    <num>8000000</num>
    <conn_resume>222</conn_resume>
    <logging>1</logging>
</conn_limit>
<conn_rate_limit>
    <status>1</status>
    <num>1000</num>
    <sample_per>1</sample_per>
    <action>0</action>
    <logging>1</logging>
</conn_rate_limit>
<slow_start>
    <status>1</status>
    <from>128</from>
    <by_mul>2</by_mul>
    <every>10</every>
    <till>4096</till>
</slow_start>
</server_template>
</response>

```

Menus Privilege

Config Mode >> Service >> SLB >> Template >> Server

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.6.3 “slb.template.server.create” Method

This method is used to create a Virtual IP (VIP) template.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.template.server.create	String		
name (*)	server template name	String		
health_monitor	health monitor	String		Default
conn_limit_status	connection limit status, disabled(0) or enabled(1)	Int	0 or 1	0
conn_limit	connection limit, only when conn_limit_status is enabled(1)	Int	1 – 8000000	8000000
conn_resume	only when conn_limit_status is enabled(1)	Int	1 - 1048575	0
conn_limit_log	disabled(0) or enabled(1), only conn_limit_status is enabled(1)	Int	0 or 1	0
conn_rate_limit_status	connection rate limit status, disabled(0) or enabled(1)	Int	0 or 1	0
conn_rate_limit	connection rate limit, only conn_rate_limit_status is enabled(1)	Int	1 – 1048575	1000
conn_rate_limit_sample_per	100ms(0) or 1 second(1), only when conn_limit_rate_status is enabled(1)	Int	0 or 1	0
conn_rate_limit_log	disabled(0) or enabled(1) only conn_rate_limit_status is enabled(1)	Int	0 or 1	0
slow_start_status	slow start status, disabled(0) or enabled(1)	Int	0 or 1	0
slow_start_from	only when slow_start_status is enabled(1)	Int	1 - 4095	128
slow_start_by_mul	only when slow_start_status is enabled(1)	Int	1 - 10	2
slow_start_by_add	only when slow_start_status is enabled(1)	Int	1 - 4095	128
slow_start_every	only when slow_start_status is enabled(1)	Int	1 - 60	10
slow_start_till	only when slow_start_status is enabled(1)	Int	1 - 65535	4096
dns_que_int	DNS query interval	Int	1 - 1440	10

min_tll_radio	minimum TLL radio	Int	1 – 15	2
max_dyn_srv	maximum dynamic server	Int	1 – 1023	255
pre_dyn_srv	prefix of dynamic server	String		DRS

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
health monitor default
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxxxx&method=slb.template.server.create&name=server_template
_name1&conn_limit_status=1&conn_limit=1000000&conn_resume=1000&conn_limit_log=1&conn_rate_limit_status=1&conn_rate_limit=1000&conn_rate_limit_log=0&health_monitor=default&dns_que_int=1
```

CLI configuration after aXAPI call:

```
!
health monitor default
!
!
slb template server server_template_name1
  dns-query-interval 1
  health-check default
  conn-limit 1000000 resume 1000
  conn-rate-limit 1000 no-logging
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> SLB >> Template >> Server

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.6.4 “slb.template.server.update” Method

This method is used to update one or more parameters in a server template.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.template.server.update	String		
name (*)	server template name	String		
health_monitor	health monitor	String		
conn_limit_status	connection limit status, disabled(0) or enabled(1)	Int	0 or 1	
conn_limit	connection limit, only when conn_limit_status is enabled(1)	Int	1 - 8000000	
conn_resume	only when conn_limit_status is enabled(1)	Int	1 - 1048575	
conn_limit_log	disabled(0) or enabled(1), only conn_limit_status is enabled(1)	Int	0 or 1	
conn_rate_limit_status	connection rate limit status, disabled(0) or enabled(1)	Int	0 or 1	
conn_rate_limit	connection rate limit, only conn_rate_limit_status is enabled(1)	Int	1 - 1048575	
conn_rate_limit_sample_per	100ms(0) or 1 second(1), only when conn_limit_rate_status is enabled(1)	Int	0 or 1	
conn_rate_limit_log	disabled(0) or enabled(1) only conn_rate_limit_status is enabled(1)	Int	0 or 1	
slow_start_status	slow start status, disabled(0) or enabled(1)	Int	0 or 1	
slow_start_from	only when slow_start_status is enabled(1)	Int	1 - 4095	
slow_start_by_mul	only when slow_start_status is enabled(1)	Int	1 - 10	
slow_start_by_add	only when slow_start_status is enabled(1)	Int	1 - 4095	
slow_start_every	only when slow_start_status is enabled(1)	Int	1 - 60	
slow_start_till	only when slow_start_status is enabled(1)	Int	1 - 65535	
dns_que_int	DNS query interval	Int	1 - 1440	
min_tll_radio	minimal TLL radio	Int	1 - 15	
max_dyn_srv	maximum dynamic server	Int	1 - 1023	
pre_dyn_srv	prefix of dynamic server	String		

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
health monitor default
!
!
slb template server server_template_name1
  slow-start
  conn-limit 8000000 resume 222
  conn-rate-limit 1000
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.template.server.update&name=server_template_na
me1&conn_limit_status=1&conn_limit=1000000&conn_resume=1000&conn_limit_log=1&conn_rate_li
mit_status=1&conn_rate_limit=1000&conn_rate_limit_log=0&health_monitor=default&dns_que_in
t=1
```

CLI configuration after aXAPI call:

```
!
health monitor default
```

```
!
!
slb template server server_template_name1
  slow-start
  dns-query-interval 1
  health-check default
  conn-limit 1000000 resume 1000
  conn-rate-limit 1000 no-logging
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> SLB >> Template >> Server

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.6.5 “slb.template.server.delete” Method

This method is used to delete a server template.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.template.server.delete	String		
name (*)	server template name	String		

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
slb template server server_template_name1
  slow-start
  conn-limit 8000000 resume 222
  conn-rate-limit 1000
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.template.server.delete&name=server_template_na
me1
```

Response as the HTTP body:

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

```
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> SLB >> Template >> Server

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.7 Server Port Template

7.7.1 “slb.template.rport.getAll” Method

This method is used to get configuration information for all rport templates configured on the AX.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.template.rport.getAll	String		

Note: (*) parameter is required in the API.

Response Fields

rport_template_list	XML tag for the collection of rport templates
rport_template	XML tag for rport template
name	rport template name
health_monitor	health monitor name
weight	
nat_pool	source nat pool name
dsr	direct server return status, disabled(0) or enabled(1)
dscp_status	disabled(0) or enabled(1)
dscp	DSCP number, only when dscp_status is enabled(1)
inband_health_check	XML tag for rport template inband health check
status	disabled(0) or enabled(1)
inband_health_check_retry	only when status is enabled(1)
inband_health_check_reassign	only when status is enabled(1)
dyn_mem_pri	dynamic member priority
dyn_mem_pri_dec	dynamic member priority decrement
conn_limit	XML tag for rport template connection limit
status	connection limit status, disabled(0) or enabled(1)
num	connection limit, only when conn_limit is enabled(1)
conn_resume	only when conn_limit is enabled(1)
logging	disabled(0) or enabled(1), only conn_limit is enabled(1)
conn_rate_limit	XML tag for rport template connection rate limit
status	connection rate limit status, disabled(0) or enabled(1)
num	only when conn_rate_limit is enabled(1)

sample_per logging	100ms(0) or 1 second(1), only conn_rate_limit is enabled(1) disabled(0) or enabled(1), only conn_rate_limit is enable(1)
slow_start status from by_mul by_add every till	XML tag for rport template slow start disabled(0) or enabled(1) slow start from slow start by multiplying slow start by adding slow start every number slow start till number

Example Response

Request as:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
- **HTTP Action:**
GET

CLI configuration before aXAPI call:

```
!
ip nat pool nat_pool 3.3.3.3 3.3.3.3 netmask /24
!
slb template port rport_template_name1
  inband-health-check
  slow-start
  conn-limit 8000000 resume 1000
  conn-rate-limit 1000
  dscp 10
  source-nat nat_pool
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.template.rport.getAll

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
  <rport_template_list>
    <rport_template>
      <name>default</name>
      <health_monitor>(default)</health_monitor>
      <weight>0</weight>
      <nat_pool/>
      <dsr>0</dsr>
      <dscp_status>0</dscp_status>
      <inband_health_check>
        <status>0</status>
      </inband_health_check>
      <dyn_mem_pri>0</dyn_mem_pri>
      <dyn_mem_pri_dec>0</dyn_mem_pri_dec>
      <conn_limit>
        <status>0</status>
      </conn_limit>
      <conn_rate_limit>
        <status>0</status>
      </conn_rate_limit>
      <slow_start>
        <status>0</status>
      </slow_start>
    </rport_template>
    <rport_template>
      <name>rport_template_name1</name>
      <health_monitor>(default)</health_monitor>
      <weight>1</weight>
      <nat_pool>nat_pool</nat_pool>
    </rport_template>
  </rport_template_list>
</response>
```

```

<dsr>0</dsr>
<dscp_status>1</dscp_status>
<dscp>10</dscp>
<inband_health_check>
  <status>1</status>
  <inband_health_check_retry>2</inband_health_check_retry>
  <inband_health_check_reassign>25</inband_health_check_reassign>
</inband_health_check>
<dyn_mem_pri>16</dyn_mem_pri>
<dyn_mem_pri_dec>5</dyn_mem_pri_dec>
<conn_limit>
  <status>1</status>
  <num>8000000</num>
  <conn_resume>1000</conn_resume>
  <logging>1</logging>
</conn_limit>
<conn_rate_limit>
  <status>1</status>
  <num>1000</num>
  <sample_per>1</sample_per>
  <action>0</action>
  <logging>1</logging>
</conn_rate_limit>
<slow_start>
  <status>1</status>
  <from>128</from>
  <by_mul>2</by_mul>
  <every>10</every>
  <till>4096</till>
</slow_start>
</rport_template>
</rport_template_list>
</response>

```

Menus Privilege

Config Mode >> Service >> SLB >> Template >> Server Port

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.7.2 “slb.template.rport.search” Method

This method is used to get rport template information for a given rport template name.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.template.rport.search	String		
name	rport template name	String		

Note: (*) parameter is required in the API.

Response Fields

rport_template	XML tag for rport template
name	rport template name
health_monitor	health monitor name
weight	
nat_pool	source nat pool name
dsr	direct server return status, disabled(0) or enabled(1)
dscp_status	disabled(0) or enabled(1)
dscp	DSCP number, only when dscp_status is enabled(1)

inband_health_check	XML tag for rport template inband health check
status	disabled(0) or enabled(1)
inband_health_check_retry	only when status is enabled(1)
inband_health_check_reassign	only when status is enabled(1)
dyn_mem_pri	dynamic member priority
dyn_mem_pri_dec	dynamic member priority decrement
conn_limit	XML tag for rport template connection limit
status	connection limit status, disabled(0) or enabled(1)
num	connection limit, only when conn_limit is enabled(1)
conn_resume	only when conn_limit is enabled(1)
logging	disabled(0) or enabled(1), only conn_limit is enabled(1)
conn_rate_limit	XML tag for rport template connection rate limit
status	connection rate limit status, disabled(0) or enabled(1)
num	only when conn_rate_limit is enabled(1)
sample_per	100ms(0) or 1 second(1), only conn_rate_limit is enabled(1)
logging	disabled(0) or enabled(1), only conn_rate_limit is enable(1)
slow_start	XML tag for rport template slow start
status	disabled(0) or enabled(1)
from	slow start from
by_mul	slow start by multiplying
by_add	slow start by adding
every	slow start every number
till	slow start till number

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
ip nat pool nat_pool 3.3.3.3 3.3.3.3 netmask /24
!
slb template port rport_template_name1
  inband-health-check
  slow-start
  conn-limit 8000000 resume 1000
  conn-rate-limit 1000
  dscp 10
  source-nat nat_pool
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.template.rport.search&name=rport_template_name1

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
  <rport_template>
```

```

<name>rport_template_name1</name>
<health_monitor>(default)</health_monitor>
<weight>1</weight>
<nat_pool>nat_pool</nat_pool>
<dsr>0</dsr>
<dscp_status>1</dscp_status>
<dscp>10</dscp>
<inband_health_check>
  <status>1</status>
  <inband_health_check_retry>2</inband_health_check_retry>
  <inband_health_check_reassign>25</inband_health_check_reassign>
</inband_health_check>
<dyn_mem_pri>16</dyn_mem_pri>
<dyn_mem_pri_dec>5</dyn_mem_pri_dec>
<conn_limit>
  <status>1</status>
  <num>8000000</num>
  <conn_resume>1000</conn_resume>
  <logging>1</logging>
</conn_limit>
<conn_rate_limit>
  <status>1</status>
  <num>1000</num>
  <sample_per>1</sample_per>
  <action>0</action>
  <logging>1</logging>
</conn_rate_limit>
<slow_start>
  <status>1</status>
  <from>128</from>
  <by_mul>2</by_mul>
  <every>10</every>
  <till>4096</till>
</slow_start>
</rport_template>
</response>

```

Menus Privilege

Config Mode >> Service >> SLB >> Template >> Server Port

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.7.3 “slb.template.rport.create” Method

This method is used to create an rport template.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.template.rport.create	String		
name (*)	rport template name	String		
health_monitor	health monitor	String		Default
weight	weight	Int	1 – 100	1
conn_limit_status	connection limit status, disabled(0) or enabled(1)	Int	0 or 1	0
conn_limit	connection limit, only when conn_limit_status is enabled(1)	Int	1 – 8000000	8000000
conn_resume	only when conn_limit_status is enabled(1)	Int	1 - 1048575	0
conn_limit_log	disabled(0) or enabled(1), only conn_limit_status is enabled(1)	Int	0 or 1	0
conn_rate_limit_status	connection rate limit status, disabled(0) or	Int	0 or 1	0

	enabled(1)			
conn_rate_limit	connection rate limit, only conn_rate_limit_status is enabled(1)	Int	1 – 1048575	1000
conn_rate_limit_sample_per	100ms(0) or 1 second(1), only when conn_limit_rate_status is enabled(1)	Int	0 or 1	0
conn_rate_limit_log	disabled(0) or enabled(1) only conn_rate_limit_status is enable(1)	Int	0 or 1	0
slow_start_status	slow start status, disabled(0) or enabled(1)	Int	0 or 1	0
slow_start_from	only when slow_start_status is enabled(1)	Int	1 - 4095	128
slow_start_by_mul	only when slow_start_status is enabled(1)	Int	1 – 10	2
slow_start_by_add	only when slow_start_status is enabled(1)	Int	1 - 4095	
slow_start_every	only when slow_start_status is enabled(1)	Int	1 – 60	10
slow_start_till	only when slow_start_status is enabled(1)	Int	1 – 65535	4096
nat_pool	source nat pool name	String		
dsr	direct server return status, disabled(0) or enabled(1)	Int	0 or 1	0
dscp_status	disabled(0) or enabled(1)	Int	0 or 1	0
dscp	only when dscp_status is enabled(1)	Int	1 – 63	
inband_health_check_status	disabled(0) or enabled(1)	Int	0 or 1	0
inband_health_check_retry	only when inband_health_check_status is enabled(1)	Int	0 – 7	2
inband_health_check_reassig	only when inband_health_check_status is enabled(1)	Int	0 – 255	25
dyn_mem_pri	dynamic member priority	Int	1 – 16	16
dyn_mem_pri_dec	dynamic member priority decrement	Int	0 – 7	0

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
- **HTTP Action:**
POST

CLI configuration before aXAPI call:

```
!
health monitor default
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxxx&method=slb.template.rport.create&name=rport_template_n
ame1&conn_limit_status=1&conn_limit=1000000&conn_resume=1000&conn_limit_log=1&conn_rat
e_limit_status=1&conn_rate_limit=1000&conn_rate_limit_log=0&health_monitor=default&dsc
p_status=0
```

CLI configuration after aXAPI call:

```
!
health monitor default
!
slb template port rport_template_name1
    conn-limit 1000000 resume 1000
    conn-rate-limit 1000 no-logging
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> SLB >> Template >> Server Port

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.7.4 “slb.template.rport.update” Method

This method is used to update one or more parameters in an rport template.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.template.rport.update	String		
name (*)	rport template name	String		
health_monitor	health monitor	String		
weight	weight	Int	1 – 100	
conn_limit_status	connection limit status, disabled(0) or enabled(1)	Int	0 or 1	
conn_limit	connection limit, only when conn_limit_status is enabled(1)	Int	1 – 8000000	
conn_resume	only when conn_limit_status is enabled(1)	Int	1 - 1048575	
conn_limit_log	disabled(0) or enabled(1), only conn_limit_status is enabled(1)	Int	0 or 1	
conn_rate_limit_status	connection rate limit status, disabled(0) or enabled(1)	Int	0 or 1	
conn_rate_limit	connection rate limit, only conn_rate_limit_status is enabled(1)	Int	1 – 1048575	
conn_rate_limit_sample_per	100ms(0) or 1 second(1), only when conn_rate_limit_status is enabled(1)	Int	0 or 1	
conn_rate_limit_log	disabled(0) or enabled(1) only conn_rate_limit_status is enabled(1)	Int	0 or 1	
slow_start_status	slow start status, disabled(0) or enabled(1)	Int	0 or 1	
slow_start_from	only when slow_start_status is enabled(1)	Int	1 - 4095	
slow_start_by_mul	only when slow_start_status is enabled(1)	Int	1 – 10	
slow_start_by_add	only when slow_start_status is enabled(1)	Int	1 - 4095	
slow_start_every	only when slow_start_status is enabled(1)	Int	1 – 60	
slow_start_till	only when slow_start_status is enabled(1)	Int	1 – 65535	
nat_pool	source nat pool name	String		
dsr	direct server return status, disabled(0) or enabled(1)	Int	0 or 1	
dscp_status	disabled(0) or enabled(1)	Int	0 or 1	
dscp	only when dscp_status is enabled(1)	Int	1 – 63	
inband_health_check_status	disabled(0) or enabled(1)	Int	0 or 1	
inband_health_check_retry	only when inband_health_check_status is enabled(1)	Int	0 – 7	
inband_health_check_reassig	only when inband_health_check_status is enabled(1)	Int	0 – 255	
dyn_mem_pri	dynamic member priority	Int	1 – 16	
Dyn_mem_pri_dec	dynamic member priority decrement	Int	0 – 7	

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:

https://[AX_IP_ADDRESS]/services/rest/V2/

- **HTTP Action:**
POST

CLI configuration before aXAPI call:

```
!
health monitor default
!
!
slb template port rport_template_name1
  inband-health-check
  slow-start
  conn-limit 8000000 resume 1000
  conn-rate-limit 1000
  dscp 10
  source-nat nat_pool
!!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.template.rport.update&name=rport_template_name1&
conn_limit_status=1&conn_limit=1000000&conn_resume=1000&conn_limit_log=1&conn_rate_limit_
status=1&conn_rate_limit=1000&conn_rate_limit_log=0&health_monitor=default&dscp_status=0
```

CLI configuration after aXAPI call:

```
!
health monitor default
!
slb template port rport_template_name1
  health-check default
  inband-health-check
  slow-start
  conn-limit 1000000 resume 1000
  conn-rate-limit 1000 per 100ms no-logging
  source-nat nat_pool
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> SLB >> Template >> Server Port

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.7.5 “slb.template.rport.delete” Method

This method is used to delete an rport template.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.template.rport.delete	String		
name (*)	rport template name	String		

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
health monitor default
!
!
slb template port rport_template_name1
  inband-health-check
  slow-start
  conn-limit 8000000 resume 1000
  conn-rate-limit 1000
  dscp 10
  source-nat nat_pool
!!
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.template.rport.delete&name=rport_template_name1

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> SLB >> Template >> Server Port

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.8 HTTP Template

7.8.1 “slb.template.http.getAll” Method

This method is used to get configuration information for all HTTP templates configured on the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.template.http.getAll	String		

Note: (*) parameter is required in the API.

Response Fields

http_template_list

XML tag for the collection of http templates

http_template	XML tag for rport template
name	http template name
fav_url	failover URL
str_trans_swi	strict transaction switching, disabled(0) or enabled(1)
cli_ip_head_insert_status	client IP header insert status, disable(0) enable(1)
cli_ip_head_insert	client IP header insert status, only status is enable(1)
cli_ip_head_insert_rep	client IP header insert replace, disabled(0) or enabled(1), only when cli_ip_head_insert status is enabled(1)
retry_http_req_status	retry http request status, disabled(0) or enabled(1)
retry_http_req_type	retry http request type, on HTTP 5xx code(0) or on HTTP code for each request(1), only when retry_http_req_status is enabled(1)
retry_http_req	retry http request, only when status is enabled(1)
ter_http_client	terminal HTTP 1.1 client when request has connection close, disabled(0) or enabled(1)
 header_erase	
requests	XML tag for http template header erase
request	XML tag for the collection of header erase requests
responses	XML tag for header erase request
response	XML tag for the collection of header erase responses
	XML tag for header erase response
 header_insert	
requests	XML tag for http template header insert
request	XML tag for the collection of header insert requests
name	XML tag for header insert request
type	header erase request name
responses	none(0), insert always(1) or insert if not already present(2)
response	XML tag for the collection of header insert responses
name	XML tag for header insert response
type	header insert response name
	none(0), insert always(1) or insert if not already present(2)
 app_switch	
url_switchs	XML tag for http template app switching
url_switch	XML tag for the collection of URL switchings
url	XML tag for URL switching
service_group	URL
match_type	service group name
host_switchs	contains(0), starts with(1), ends with(2) or equals(3)
host_switch	XML tag for the collection of host switchings
host	XML tag for host switching
service_group	hostname
match_type	service group name
url_hash	contains(0), starts with(1) or ends with(2)
status	XML tag for URL hash
type	disabled(0) or enabled(1)
num	first(0) or last(1), only when status is enabled(1)
use_server_status	url hash number, only when status is enabled(1)
	disabled(0) or enabled(1), only when status is enable(1)

offset	0 – 255, only when status is enable(1)
redirect_rewrite	XML tag for http template redirect rewrite
red_rews	XML tag for the collection of redirect rewrites
red_rewrite	XML tag for redirect rewrite
pattern	pattern
redirect	redirect to
https_rewrite	XML tag for https rewrite
status	disabled(0) or enabled(1)
num	https rewrite number, only when status is enabled(1)
compression	XML tag for http template compression
status	compression status, disabled(0) or enabled(1)
keep_acc_enc	keep accept encoding, disabled(0) or enabled(1)
level	from 1 to 9
min_con_len	min content length
con_types	XML tag for the collection of content type
con_type	XML tag for content type
ex_con_types	XML tag for the collection of exclude content type
ex_con_type	XML tag for exclude content type
ex_uris	XML tag for the collection of exclude URIs
ex_uri	XML tag for exclude URI

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
slb template http http_template_name1
!
slb template http http_template_name2
    failover-url abc
    request-header-insert 3::3
    response-header-insert 4::4 insert-always
    insert-client-ip replace
    strict-transaction-switch
    term-11client-hdr-conn-close
    retry-on-5xx
    compression enable
    compression level 5
    compression minimum-content-length 120
    compression keep-accept-encoding enable
    compression content-type 1
    compression content-type 2
    compression exclude-uri 3
    url-switching starts-with def service-group src
    url-switching contains abc service-group src
    redirect-rewrite match 1 rewrite-to 2
    redirect-rewrite match a rewrite-to b
    redirect-rewrite secure
    url-hash-persist last 4 use-server-status
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.template.http.getAll

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
  <http_template_list>
    <http_template>
      <name>http_template_name1</name>
      <fav_url/>
      <str_trans_swi>0</str_trans_swi>
      <cli_ip_head_insert_status>0</cli_ip_head_insert_status>
      <retry_http_req_status>0</retry_http_req_status>
      <ter_http_client>0</ter_http_client>

      <header_erase/>

      <header_insert/>

      <app_switch>
        <host_switchs/>
        <url_switchs/>
        <url_hash>
          <status>0</status>
        </url_hash>
      </app_switch>

      <redirect_rewrite>
        <red_rews/>
        <https_rew>
          <status>0</status>
        </https_rew>
      </redirect_rewrite>

      <compression>
        <status>0</status>
        <keep_acc_enc>0</keep_acc_enc>
        <level>1</level>
        <min_con_len>0</min_con_len>
        <con_types/>
        <ex_con_types/>
        <ex_uris/>
      </compression>
    </http_template>

    <http_template>
      <name>http_template_name2</name>
      <fav_url>abc</fav_url>
      <str_trans_swi>1</str_trans_swi>
      <cli_ip_head_insert_status>1</cli_ip_head_insert_status>
      <cli_ip_head_insert>X-ClientIP</cli_ip_head_insert>
      <cli_ip_head_insert_rep>1</cli_ip_head_insert_rep>
      <retry_http_req_status>1</retry_http_req_status>
      <retry_http_req_type>0</retry_http_req_type>
      <retry_http_req>3</retry_http_req>
      <ter_http_client>1</ter_http_client>

      <header_erase>
        <requests>
          <request>1</request>
        </requests>
        <responses>
          <response>2</response>
        </responses>
      </header_erase>

      <header_insert>
        <requests>
          <request>
            <name>3::3</name>
            <type>0</type>
          </request>
        </requests>
        <responses>
```

```

<response>
  <name>4::4</name>
  <type>1</type>
</response>
</responses>
</header_insert>

<app_switch>
  <host_switches/>
  <url_switches>
    <url_switch>
      <url>abc</url>
      <service_group>src</service_group>
      <match_type>0</match_type>
    </url_switch>
    <url_switch>
      <url>def</url>
      <service_group>src</service_group>
      <match_type>1</match_type>
    </url_switch>
  </url_switches>
  <url_hash>
    <status>1</status>
    <type>1</type>
    <num>4</num>
    <use_server_status>1</use_server_status>
    <offset>0</offset>
  </url_hash>
</app_switch>

<redirect_rewrite>
  <red_rews>
    <red_rew>
      <pattern>1</pattern>
      <redirect>2</redirect>
    </red_rew>
    <red_rew>
      <pattern>a</pattern>
      <redirect>b</redirect>
    </red_rew>
  </red_rews>
  <https_rew>
    <status>1</status>
    <num>443</num>
  </https_rew>
</redirect_rewrite>

<compression>
  <status>1</status>
  <keep_acc_enc>1</keep_acc_enc>
  <level>5</level>
  <min_con_len>120</min_con_len>
  <con_types>
    <con_type>1</con_type>
  </con_types>
  <ex_con_types>
    <ex_con_type>2</ex_con_type>
  </ex_con_types>
  <ex_uris>
    <ex_uri>3</ex_uri>
  </ex_uris>
</compression>
</http_template>
</http_template_list>
</response>

```

Menus Privilege

Config Mode >> Service >> Template >> Application >> HTTP

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.8.2 “slb.template.http.search” Method

This method is used to get http template information for a given http template name.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.template.http.search	String		
name	http template name	String		

Note: (*) parameter is required in the API.

Response Fields

http_template	XML tag for report template
name	http template name
fav_url	failover URL
str_trans_swi	strict transaction switching, disabled(0) or enabled(1)
cli_ip_head_insert_status	client IP header insert status, disable(0) enable(1)
cli_ip_head_insert	client IP header insert status, only status is enable(1)
cli_ip_head_insert_rep	client IP header insert replace, disabled(0) or enabled(1), only when cli_ip_head_insert status is enabled(1)
retry_http_req_status	retry http request status, disabled(0) or enabled(1)
retry_http_req_type	retry http request type, on HTTP 5xx code(0) or on HTTP code for each request(1), only when retry_http_req_status is enabled(1)
retry_http_req	retry http request, only when status is enabled(1)
ter_http_client	terminal HTTP 1.1 client when request has connection close, disabled(0) or enabled(1)
header_erase	
requests	XML tag for http template header erase
request	XML tag for the collection of header erase requests
responses	XML tag for header erase request
response	XML tag for the collection of header erase responses
	XML tag for header erase response
header_insert	
requests	XML tag for http template header insert
request	XML tag for the collection of header insert requests
name	XML tag for header insert request
type	header erase request name
responses	none(0), insert always(1) or insert if not already present(2)
response	XML tag for the collection of header insert responses
name	XML tag for header insert response
type	header insert response name
	none(0), insert always(1) or insert if not already present(2)
app_switch	XML tag for http template app switching

url_switchs	XML tag for the collection of URL switchings
url_switch	XML tag for URL switching
url	URL
service_group	service group name
match_type	contains(0), starts with(1), ends with(2) or equals(3)
host_switchs	XML tag for the collection of host switchings
host_switch	XML tag for host switching
host	hostname
service_group	service group name
match_type	contains(0), starts with(1) or ends with(2)
url_hash	XML tag for URL hash
status	disabled(0) or enabled(1)
type	first(0) or last(1), only when status is enabled(1)
num	url hash number, only when status is enabled(1)
use_server_status	disabled(0) or enabled(1), only when status is enable(1)
offset	0 – 255, only when status is enable(1)
 redirect_rewrite	
red_rews	XML tag for http template redirect rewrite
red_rew	XML tag for the collection of redirect rewrites
pattern	XML tag for redirect rewrite
redirect	pattern
https_rew	redirect to
status	XML tag for https rewrite
num	disabled(0) or enabled(1)
	https rewrite number, only when status is enabled(1)
 compression	
status	XML tag for http template compression
keep_acc_enc	compression status, disabled(0) or enabled(1)
level	keep accept encoding, disabled(0) or enabled(1)
min_con_len	from 1 to 9
con_types	min content length
con_type	XML tag for the collection of content type
ex_con_types	XML tag for content type
ex_con_type	XML tag for the collection of exclude content type
ex_uris	XML tag for exclude content type
ex_uri	XML tag for the collection of exclude URIs
	XML tag for exclude URI

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
slb template http http_template_name1
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.template.http.search&name=http_template_name1

Response as the HTTP body:

```

<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
    <http_template>
        <name>http_template_name1</name>
        <fav_url/>
        <str_trans_swi>0</str_trans_swi>
        <cli_ip_head_insert_status>0</cli_ip_head_insert_status>
        <retry_http_req_status>0</retry_http_req_status>
        <ter_http_client>0</ter_http_client>

        <header_erase/>

        <header_insert/>

        <app_switch>
            <host_switches/>
            <url_switches/>
            <url_hash>
                <status>0</status>
            </url_hash>
        </app_switch>

        <redirect_rewrite>
            <red_rews/>
            <https_rew>
                <status>0</status>
            </https_rew>
        </redirect_rewrite>

        <compression>
            <status>0</status>
            <keep_acc_enc>0</keep_acc_enc>
            <level>0</level>
            <min_con_len>0</min_con_len>
            <con_types/>
            <ex_con_types/>
            <ex_uris/>
        </compression>
    </http_template>
</response>

```

Menus Privilege

Config Mode >> Service >> Template >> Application >> HTTP

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.8.3 “slb.template.http.create” Method

This method is used to create an HTTP template.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.template.http.create	String		
name (*)	http template name	String		
fav_url	failover URL	String		
str_trans_swi	strict transaction switching, disabled(0) or enabled(1)	Int	0 or 1	0
cli_ip_head_insert_status	client IP header insert status, disabled(0) or enabled(1)	Int	0 or 1	0
cli_ip_head_inert	only when cli_ip_head_insert_status is enabled(1)	String		
cli_ip_head_insert_rep	client IP header insert replace, disabled(0) or enable(1)	Int	0 or 1	0

	only when cli_ip_head_insert_status is enabled(1)			
retry_http_req_status	retry http request status disabled(0) or enabled(1)	Int	0 or 1	0
retry_http_req_type	on http 5xx code(0) or on http 5xx code for each request(1), only when retry_http_req_status is enabled	Int	0 or 1	0
retry_http_req	only when retry_http_req_status is enabled (1)	Int	1 – 3	3
ter_http_client	terminal HTTP 1.1 client when request has connection:close, disabled(0) or enabled(1)	Int	0 or 1	0
header_erase_req_list	header erase request list to be added header_erase_req1^B ...header_erase_req N ^B: ASCII Code 0x02, URL-encode %02	String		
header_erase_req<n> name (*)	header erase request at element <n> request name	String		
header_erase_res_list	header erase response list to be added header_erase_res1^B ...header_erase_res N ^B: ASCII Code 0x02, URL-encode %02	String		
header_erase_res<n> name (*)	header erase response at element <n> response name	String		
header_insert_req_list	header insert request list to be added header_insert_req1^B ...header_insert_req N ^B: ASCII Code 0x02, URL-encode %02	String		
header_insert_req<n> name (*) type	header insert request at element <n> request name none(0), insert always(1) or insert if not present(2)	String Int	0, 1 or 2	0
header_insert_res_list	header insert response list to be added header_insert_res1^B ...header_insert_res N ^B: ASCII Code 0x02, URL-encode %02	String		
header_insert_res<n> name (*) type	header insert response at element <n> response name none(0), insert always(1) or insert if not present(2)	String Int	0 , 1 or 2	0
url_swi_list (*)	URL switching list to be added url_swi1^B ...url_swiN ^B: ASCII Code 0x02, URL-encode %02	String		
url_swi<n> URL (*) service_group match_type	URL switching at element <n> URL service group name contain(0), starts with(1), ends with(2) or equals(3)	String String Int	0, 1, 2 or 3	0
host_swi_list (*)	host switching list to be added host_swi1^B ...host_swiN ^B: ASCII Code 0x02, URL-encode %02	String		
host_swi<n> host (*) service_group match_type	host switching at element <n> hostname service group name contain(0), starts with(1) or ends with(2)	String String Int	0 , 1 or 2	0
url_hash_status	disabled(0) or enabled(1)	Int	0 or 1	0
url_hash_type	first(0) or last(1), only when url_hash_status is enabled	Int	0 or 1	0
url_hash	URL hash, only when url_hash_status is enabled(1)	Int	4 – 128	4
use_server_status	use server status, disabled(0) or enabled(1)	Int	0 or 1	0
offset	only when url_hash_status is enabled(1)	Int	0 – 255	0
red_rew_list (*)	redirect rewrite list to be added red_rew1^B ...red_rewN ^B: ASCII Code 0x02, URL-encode %02	String		
red_rew<n> pattern (*) red (*)	redirect rewrite at element <n> pattern redirect to	String String		
https_rew_status	HTTPS rewrite status, disabled(0) or enabled(1)	Int	0 or 1	0
https_rew	HTTPS rewrite	Int	1 – 65535	443
compression_status	disabled(0) or enabled(1)	Int	0 or 1	0
keep_acc_enc	keep accept encoding, disabled(0) or enabled(1)	Int	0 or 1	0
level	Level	Int	1 – 9	1
min_con_len	min content length	Int	0-2147483647	120
con_type_list (*)	content type list to be added con_type1^B ...con_typeN ^B: ASCII Code 0x02, URL-encode %02	String		
con_type<n> name (*)	content type at element <n> type name	String		
ex_con_type_list (*)	exclude content type list to be added ex_con_type1^B ...ex_con_typeN ^B: ASCII Code 0x02, URL-encode %02	String		

ex_con_type<n> name (*)	Exclude content type at element <n> type name	String		
ex_uri_list (*)	exclude URI list to be added ex_uri1^B ...ex_uriN ^B: ASCII Code 0x02, URL-encode %02	String		
ex_uri<n> name (*)	exclude URI at element <n> name	String		

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.template.http.create&name=http_template_name1&str_trans_sw1=1&cli_ip_head_insert_status=0&ter_http_cli=1&header_erase_req_list=header_erase_req1%02header_erase_req2&header_erase_req1=name%03request1&header_erase_req2=name%03request2&compression_status=1&keep_acc_enc=1&level=6&con_type_list=con_type1%02con_type2&con_type1=name%03type1&con_type2=name%03type2
```

CLI configuration after aXAPI call:

```
!
slb template http http_template_name1
  request-header-erase request1
  request-header-erase request2
  strict-transaction-switch
  compression enable
  compression level 6
  compression keep-accept-encoding enable
  compression content-type type1
  compression content-type type2
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> Template >> Application >> HTTP

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.8.4 “slb.template.http.update” Method

This method is used to update one or more parameters in an HTTP template.

Parameters

Parameter Name	Description	Data	Range	Default
----------------	-------------	------	-------	---------

		Type		
session_id (*)	user authenticated session id.	String		
method (*)	slb.template.http.update	String		
name (*)	http template name	String		
fav_url	failover URL	String		
str_trans_swi	strict transaction switching, disabled(0) or enabled(1)	Int	0 or 1	
cli_ip_head_insert_status	client IP header insert status, disabled(0) or enabled(1)	Int	0 or 1	
cli_ip_head_inert	only when cli_ip_head_insert_status is enabled(1)	String		
cli_ip_head_insert_rep	client IP header insert replace, disabled(0) or enable(1) only when cli_ip_head_insert_status is enabled(1)	Int	0 or 1	
retry_http_req_status	retry http request status disabled(0) or enabled(1)	Int	0 or 1	
retry_http_req_type	on http 5xx code(0) or on http 5xx code for each request(1), only when retry_http_req_status is enabled	Int	0 or 1	
retry_http_req	only when retry_http_req_status is enabled (1)	Int	1 – 3	
ter_http_client	terminal HTTP 1.1 client when request has connection:close, disabled(0) or enabled(1)	Int	0 or 1	
header_erase_req_list (*)	header erase request list to be added header_erase_req1^B ...header_erase_req N ^B: ASCII Code 0x02, URL-encode %02	String		
header_erase_req<n>_name (*)	header erase request at element <n> request name	String		
header_erase_res_list (*)	header erase response list to be added header_erase_res1^B ...header_erase_res N ^B: ASCII Code 0x02, URL-encode %02	String		
header_erase_res<n>_name (*)	header erase response at element <n> response name	String		
header_insert_req_list (*)	header insert request list to be added header_insert_req1^B ...header_insert_req N ^B: ASCII Code 0x02, URL-encode %02	String		
header_insert_req<n>_name (*)_type	header insert request at element <n> request name none(0), insert always(1) or insert if not present(2)	String Int	0, 1 or 2	
header_insert_res_list (*)	header insert response list to be added header_insert_res1^B ...header_insert_res N ^B: ASCII Code 0x02, URL-encode %02	String		
header_insert_res<n>_name (*)_type	header insert response at element <n> response name none(0), insert always(1) or insert if not present(2)	String Int	0 , 1 or 2	
url_swi_list (*)	URL switching list to be added url_swi1^B ...url_swiN ^B: ASCII Code 0x02, URL-encode %02	String		
url_swi<n>_URL (*)_service_group_match_type	URL switching at element <n> URL service group name contain(0), starts with(1), ends with(2) or equals(3)	String String Int	0, 1, 2 or 3	
host_swi_list (*)	host switching list to be added host_swi1^B ...host_swiN ^B: ASCII Code 0x02, URL-encode %02	String		
host_swi<n>_host (*)_service_group_match_type	host switching at element <n> hostname service group name contain(0), starts with(1) or ends with(2)	String String Int	0 , 1 or 2	
url_hash_status	disabled(0) or enabled(1)	Int	0 or 1	
url_hash_type	first(0) or last(1), only when url_hash_status is enabled	Int	0 or 1	
url_hash	URL hash, only when url_hash_status is enabled(1)	Int	4 – 128	
use_server_status	use server status, disabled(0) or enabled(1)	Int	0 or 1	
offset	only when url_hash_status is enabled(1)	Int	0 – 255	
red_rew_list (*)	redirect rewrite list to be added red_rew1^B ...red_rewN ^B: ASCII Code 0x02, URL-encode %02	String		
red_rew<n>_pattern (*)_red (*)	redirect rewrite at element <n> pattern redirect to	String String		
https_rew_status	HTTPS rewrite status, disabled(0) or enabled(1)	Int	0 or 1	
https_rew	HTTPS rewrite	Int	1 – 65535	
compression_status	disabled(0) or enabled(1)	Int	0 or 1	
keep_acc_enc	keep accept encoding, disabled(0) or enabled(1)	Int	0 or 1	
level	Level	Int	1 – 9	

min_con_len	min content length	Int	0-2147483647	
con_type_list (*)	content type list to be added con_type1^B ...con_typeN ^B: ASCII Code 0x02, URL-encode %02	String		
con_type<n> name (*)	content type at element <n> type name	String		
ex_con_type_list (*)	exclude content type list to be added ex_con_type1^B ...ex_con_typeN ^B: ASCII Code 0x02, URL-encode %02	String		
ex_con_type<n> name (*)	Exclude content type at element <n> type name	String		
ex_uri_list (*)	exclude URI list to be added ex_uri1^B ...ex_uriN ^B: ASCII Code 0x02, URL-encode %02	String		
ex_uri<n> name (*)	exclude URI at element <n> name	String		

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- **URL:**
[https://\[AX_IP_ADDRESS\]/services/rest/V2/](https://[AX_IP_ADDRESS]/services/rest/V2/)
- **HTTP Action:**
POST

CLI configuration before aXAPI call:

```
!
slb template http http_template_name1
  request-header-erase request1
  request-header-erase request2
  strict-transaction-switch
  compression enable
  compression level 5
  compression keep-accept-encoding enable
  compression content-type type1
  compression content-type type2
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.template.http.update&name=http_template_name1&
str_trans_swi=1&cli_ip_head_insert_status=0&ter_http_cli=1&header_erase_req_list=header_e
rase_req1%02header_erase_req2&header_erase_req1=name%03request1&header_erase_req2=name%0
3request2&compression_status=1&keep_acc_enc=1&level=6&con_type_list=con_type1%02con_type2
&con_type1=name%03type1&con_type2=name%03type2
```

CLI configuration after aXAPI call:

```
!
slb template http http_template_name1
  request-header-erase request1
  request-header-erase request2
  compression enable
  compression level 6
  compression keep-accept-encoding enable
  compression content-type type1
  compression content-type type2
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> Template >> Application >> HTTP

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.8.5 “slb.template.http.delete” Method

This method is used to delete an HTTP port template.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.template.http.delete	String		
name (*)	http template name	String		

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
slb template http http_template name1
  request-header-erase request1
  request-header-erase request2
  strict-transaction-switch
  compression enable
  compression level 5
  compression keep-accept-encoding enable
  compression content-type type1
  compression content-type type2
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.template.http.delete&name=http_template_name1

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> Template >> Application >> HTTP

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.9 Cache Template

7.9.1 “slb.template.cache.getAll” Method

This method is used to get configuration information for all cache templates configured on the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.template.cache.getAll	String		

Note: (*) parameter is required in the API.

Response Fields

cache_template_list	XML tag for the collection of cache templates
cache_template	XML tag for cache template
name	cache template name
age	(second)
max_cache	max cache size (MB)
min_content	min content size (Bytes)
max_content	max content size (Bytes)
rep_policy	least frequently used(0)
acc_rel_req	accept reload request, disabled(0) or enabled(1)
veri_host	verify host, disabled(0) or enabled(1)
def_pol_no_cache	default policy no cache, disabled(0) or enabled(1)
insert_age	insert age, disabled(0) or enabled(1)
insert_via	insert via, disabled(0) or enabled(1)
polcys	XML tag for the collection of polcys
policy	XML tag for policy
uri	URI
action	cache(0), no cache(1) or invalidate(2)
duration	duration (second), only when act is cache(0)
pattern	only when act is invalidate(2)

Example Response

Request as:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
- **HTTP Action:**
GET

CLI configuration before aXAPI call:

```
!
slb template cache cache_template_name1
!
slb template cache cache_template_name2
  policy uri abc invalidate dddddd
  policy uri bbb cache 345
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.template.cache.getAll
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
  <cache_template_list>
    <cache_template>
      <name>cache_template_name1</name>
      <age>3600</age>
      <max_cache>80</max_cache>
      <min_content>512</min_content>
      <max_content>81920</max_content>
      <rep_policy>0</rep_policy>
      <acc_rel_req>0</acc_rel_req>
      <veri_host>0</veri_host>
      <def_pol_no_cache>0</def_pol_no_cache>
      <insert_age>0</insert_age>
      <insert_via>0</insert_via>
      <policy>
        <uri>abc</uri>
        <action>2</action>
        <pattern>ddddd</pattern>
      </policy>
      <policy>
        <uri>bbb</uri>
        <action>0</action>
        <duration>345</duration>
      </policy>
    </cache_template>
  </cache_template_list>
</response>
```

Menus Privileq

Config Mode >> Service >> Template >> Application >> RAM Caching

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.9.2 “slb.template.cache.search” Method

This method is used to get cache template information for a given slb cache template name.

Parameters

Parameter Name	Description	Data Type	Range	Default
----------------	-------------	-----------	-------	---------

session_id (*)	user authenticated session id	String		
method (*)	slb.template.cache.search	String		
name	cache template name	String		

Note: (*) parameter is required in the API.

Response Fields

cache_template	XML tag for cache template
name	cache template name
age	(second)
max_cache	max cache size (MB)
min_content	min content size (Bytes)
max_content	max content size (Bytes)
rep_policy	least frequently used(0)
acc_rel_req	accept reload request, disabled(0) or enabled(1)
veri_host	verify host, disabled(0) or enabled(1)
def_pol_no_cache	default policy no cache, disabled(0) or enabled(1)
insert_age	insert age, disabled(0) or enabled(1)
insert_via	insert via, disabled(0) or enabled(1)
policy	XML tag for the collection of policys
uri	XML tag for policy
action	URI
duration	cache(0), no cache(1) or invalidate(2)
pattern	duration (second), only when act is cache(0) only when act is invalidate(2)

Example Response

Request as:

- URL:
[https://\[AX_IP_ADDRESS\]/services/rest/V2/](https://[AX_IP_ADDRESS]/services/rest/V2/)
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
slb template cache cache_template_name1
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.template.cache.search&name=cache_template_name1

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
    <cache_template>
        <name>cache_template_name1</name>
        <age>3600</age>
        <max_cache>80</max_cache>
        <min_content>512</min_content>
        <max_content>81920</max_content>
        <rep_policy>0</rep_policy>
        <acc_rel_req>0</acc_rel_req>
        <veri_host>0</veri_host>
        <def_pol_no_cache>0</def_pol_no_cache>
        <insert_age>0</insert_age>
        <insert_via>0</insert_via>
        <policy>
    </cache_template>
```

```
</response>
```

Menus Privilege

Config Mode >> Service >> Template >> Application >> RAM Caching

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.9.3 “slb.template.cache.create” Method

This method is used to create a cache template.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.template.cache.create	String		
name (*)	cache template name	String		
age	age	Int	1 – 999999	3600
max_cache	max cache size	Int	AX2500 1 – 1024 AX2600 1 – 2048 AX3000 1 – 2048 AX5100 1 – 4096 AX5200 1 – 4096 Others 1 - 512	80
min_content	min content size	Int	1 – 268435455	512
max_content	max content size	Int	1 – 268435455	81920
acc_rel_req	accept reload replacy, disabled(0) or enable(1)	Int	0 or 1	0
veri_host	verify host, disabled(0) or enabled(1)	Int	0 or 1	0
def_pol_no_cache	Default policy no-cache, disabled(0) or enabled(1)	Int	0 or 1	0
insert_age	insert age, disabled(0) or enabled(1)	Int	0 or 1	1
insert_via	Insert via, disabled(0) or enabled(1)	Int	0 or 1	1
policy_list (*)	policy list to be added policy1^B ...policyN ^B: ASCII Code 0x02, URL-encode %02	String		
policy<n> uri (*) action duration pattern	header erase request at element <n> URI action, cache(0), no cache(1) or invalidate(2) duration (second), only when action is 0 pattern, only when action is 2	String Int Int String	0, 1, or 2 1 – 999999	0 3600

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
[https://\[AX_IP_ADDRESS\]/services/rest/V2/](https://[AX_IP_ADDRESS]/services/rest/V2/)
- HTTP Action:
POST

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.template.cache.create&name=cache_template_name
1&age=3600&max_cache=80&veri_host=1&policy_list=policy1%02policy2&policy1=uri%03aaa%02ac
tion%031&policy2=uri%03bbb%02action%030%02duration%033600
```

CLI configuration after aXAPI call:

```
!
slb template cache cache_template_name1
    verify-host
    policy uri aaa nocache
    policy uri bbb cache
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> Template >> Application >> RAM Caching

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.9.4 “slb.template.cache.update” Method

This method is used to update one or more parameters in a cache template.

Note: aXAPI can not update a policy list if there is not policy_list parameter in the request. All policy parameters must be sent in the policy_list if you want to update the policy list. This is different when updating a server port before. For example, there are two policies before. If you want to change the duration in policy1, then you have to send both policy1 and policy2 in the request.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.template.cache.update	String		
name (*)	cache template name	String		
age	age	Int	1 – 999999	
mac_cache	max cache size	Int	AX2500 1 – 1024 AX2600 1 – 2048 AX3000 1 – 2048 AX5100 1 – 4096 AX5200 1 – 4096 Others 1 - 512	
min_content	min content size	Int	1 – 4194303	
max_content	max content size	Int	1 – 4194303	
acc_rel_req	accept reload replacy, disabled(0) or enable(1)	Int	0 or 1	
veri_host	verify host, disabled(0) or enabled(1)	Int	0 or 1	
def_pol_no_cache	Default policy no-cache, disabled(0) or enabled(1)	Int	0 or 1	
insert_age	insert age, disabled(0) or enabled(1)	Int	0 or 1	
insert_via	Insert via, disabled(0) or enabled(1)	Int	0 or 1	
policy_list (*)	policy list to be added policy1^B ...policyN ^B: ASCII Code 0x02, URL-encode %02	String		
policy<n> uri (*) action duration pattern	header erase request at element <n> URI action, cache(0), no cache(1) or invalidate(2) duration (second), only when action is 0 pattern, only when action is 2	String Int Int String	0, 1, or 2 1 – 999999	0 3600

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
slb template cache cache_template_name1
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxxxx&method=slb.template.cache.update&name=cache_template_name
1&age=3600&max_cache=80&veri_host=1&policy_list=policy1%02policy2&policy1=uri%03aaa%02ac
tion%031&policy2=uri%03bbb%02action%030%02duration%033600
```

CLI configuration after aXAPI call:

```
!
slb template cache cache_template_name1
    verify-host
    policy uri aaa nocache
    policy uri bbb cache
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> Template >> Application >> RAM Caching

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.9.5 “slb.template.cache.delete” Method

This method is used to delete a cache template.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.template.cache.delete	String		
name (*)	cache template name	String		

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
slb template cache cache_template_name1
  verify-host
  policy uri aaa nocache
  policy uri bbb cache
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.template.cache.delete&name=cache_template_name1

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> Template >> Application >> RAM Caching

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.10 PBSLB Template

7.10.1 “**slb.template.pbslb.getAll**” Method

This method is used to get configuration information for all PBSLB templates configured on the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.template.pbslb.getAll	String		

Notes: (*) parameter is required in the API.

Response Fields

pbslb_template_list	XML tag for the collection of PBSLB template
pbslb_template	XML tag for PBSLB template
name	PBSLB template name
pbslb	XML tag for PBSLB
name	PBSLB name
id_list	XML tag for PBSLB groups
id_entry	XML tag for PBSLB group
id	PSBLB group ID

action	DROP, RESET or other service group name
logging_enabled	logging status, enabled(1). disabled(0)
logging_interval	logging interval
logging_failure_only	enabled(1). disabled(0)
over_limit_action	over limit action, drop(0), reset(1)
over_limit_lockup_duration	over limit lockup duration
over_limit_log_interval	over limit log interval
timeout	timeout
ip_limit	XML tag for the collection of IP limiting
class_list	class list name
lid_list	XML tag for the collection of LID
lid	XML tag for LID
id	LID ID
conn_limit	connection limit
conn_rate_limit	connection rate limit
conn_rate_limit_per	connection rate limit interval
req_limit	request limit
req_rate_limit	request rate limit
req_rate_limit_per	request rate limit interval
over_limit_action	over limit action
lockout	lockout
log_status	log status, enabled(1). disabled(0)
log_interval	log interval
client_ip	L3 source IP(0), L3 destination IP(1) or L7 header name(2)
17_head_name	extract client IP address from L7 header
use_dst_ip	use destination IP, enabled(1). disabled(0)
overlap	enabled(1). disabled(0)
share	enabled(1). disabled(0)
full_domain_tree	enabled(1). disabled(0)

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
class-list cl1
!
slb template policy pbslb_template_name1
    bw-list name pl
    bw-list over-limit reset lockup 34 logging 45
    bw-list timeout 67
    bw-list id 1 drop
    bw-list id 2 reset
    bw-list use-destination-ip
    class-list name cl1
    class-list lid 16
        conn-limit 1
        conn-rate-limit 1 per 1
        request-limit 1
        request-rate-limit 1 per 10
        over-limit-action lockout 1
    class-list lid 17
```

```

conn-limit 1
conn-rate-limit 1 per 1
request-limit 1
request-rate-limit 1 per 10
over-limit-action lockout 1 log 12
!
slb template policy pbslb_template_name2
  bw-list name p1
  bw-list over-limit reset lockup 34 logging 45
  bw-list timeout 67
  bw-list id 1 drop
  bw-list id 2 reset
  bw-list use-destination-ip
  class-list name cl1
  class-list lid 16
    conn-limit 1
    conn-rate-limit 1 per 1
    request-limit 1
    request-rate-limit 1 per 10
    over-limit-action lockout 1
  class-list lid 17
    conn-limit 1
    conn-rate-limit 1 per 1
    request-limit 1
    request-rate-limit 1 per 10
    over-limit-action lockout 1 log 12
!

```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.template.pbslb.getAll

Response as the HTTP body:

```

<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
  <pbslb_template_list>
    <pbslb_template>
      <name>pbslb_template_name1</name>
      <pbslb>
        <name>p1</name>
        <id_list>
          <id_entry>
            <id>1</id>
            <action>DROP</action>
            <logging_enabled>0</logging_enabled>
            <logging_interval>3</logging_interval>
            <logging_failure_only>0</logging_failure_only>
          </id_entry>
          <id_entry>
            <id>2</id>
            <action>RESET</action>
            <logging_enabled>0</logging_enabled>
            <logging_interval>3</logging_interval>
            <logging_failure_only>0</logging_failure_only>
          </id_entry>
        </id_entry_list>
        <over_limit_action>1</over_limit_action>
        <over_limit_lockup_duration>34</over_limit_lockup_duration>
        <over_limit_log_interval>45</over_limit_log_interval>
        <timeout>67</timeout>
      </pbslb>
      <ip_limit>
        <class_list>cl1</class_list>
        <lids>
          <lid>
            <id>16</id>
            <conn_limit>1</conn_limit>
            <conn_rate_limit>1</conn_rate_limit>
            <conn_rate_limit_per>1</conn_rate_limit_per>
          </lid>
        </lids>
      </ip_limit>
    </pbslb_template>
  </pbslb_template_list>
</response>

```

```

<req_limit>1</req_limit>
<req_rate_limit>1</req_rate_limit>
<req_rate_limit_per>10</req_rate_limit_per>
<over_limit_action>0</over_limit_action>
<lockout>1</lockout>
<log_status>0</log_status>
<log_interval>0</log_interval>
</lid>
<lids>
<id>17</id>
<conn_limit>1</conn_limit>
<conn_rate_limit>1</conn_rate_limit>
<conn_rate_limit_per>1</conn_rate_limit_per>
<req_limit>1</req_limit>
<req_rate_limit>1</req_rate_limit>
<req_rate_limit_per>10</req_rate_limit_per>
<over_limit_action>0</over_limit_action>
<lockout>1</lockout>
<log_status>1</log_status>
<log_interval>12</log_interval>
</lid>
</lids>
<client_ip>0</client_ip>
</ip_limit>
<use_dst_ip>1</use_dst_ip>
<overlap>0</overlap>
<share>0</share>
<full_domain_tree>0</full_domain_tree>
</pbslb_template>
<pbslb_template>
<name>pbslb_template_name2</name>
<pbslb>
<name>p1</name>
<id_list>
<id_entry>
<id>1</id>
<action>DROP</action>
<logging_enabled>0</logging_enabled>
<logging_interval>3</logging_interval>
<logging_failure_only>0</logging_failure_only>
</id_entry>
<id_entry>
<id>2</id>
<action>RESET</action>
<logging_enabled>0</logging_enabled>
<logging_interval>3</logging_interval>
<logging_failure_only>0</logging_failure_only>
</id_entry>
</id_list>
<over_limit_action>1</over_limit_action>
<over_limit_lockup_duration>34</over_limit_lockup_duration>
<over_limit_log_interval>45</over_limit_log_interval>
<timeout>67</timeout>
</pbslb>
<ip_limit>
<class_list>c11</class_list>
<lids>
<lid>
<id>16</id>
<conn_limit>1</conn_limit>
<conn_rate_limit>1</conn_rate_limit>
<conn_rate_limit_per>1</conn_rate_limit_per>
<req_limit>1</req_limit>
<req_rate_limit>1</req_rate_limit>
<req_rate_limit_per>10</req_rate_limit_per>
<over_limit_action>0</over_limit_action>
<lockout>1</lockout>
<log_status>0</log_status>
<log_interval>0</log_interval>
</lid>
<lids>

```

```
<id>17</id>
<conn_limit>1</conn_limit>
<conn_rate_limit>1</conn_rate_limit>
<conn_rate_limit_per>1</conn_rate_limit_per>
<req_limit>1</req_limit>
<req_rate_limit>1</req_rate_limit>
<req_rate_limit_per>10</req_rate_limit_per>
<over_limit_action>0</over_limit_action>
<lockout>1</lockout>
<log_status>1</log_status>
<log_interval>12</log_interval>
</lid>
</lids>
<client_ip>0</client_ip>
</ip_limit>
<use_dst_ip>1</use_dst_ip>
<overlap>0</overlap>
<share>0</share>
<full_domain_tree>0</full_domain_tree>
</pbslb_template>
</pbslb_template_list>
</response>
```

Privileges

Config Mode >> Service >> Template >> Application >> PBSLB Policy

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.10.2 “slb.template.pbslb.search” Method

This method is used to get PBSLB template information for a given PBSLB template name.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.template.pbslb.search	String		
name (*)	pbslb template name	String		

Notes: (*) parameter is required in the API.

Response Fields

pbslb_template	XML tag for PBSLB template
name	PBSLB template name
pbslb	XML tag for PBSLB
name	PBSLB name
id_list	XML tag for PBSLB groups
id_entry	XML tag for PBSLB group
id	PSBLB group ID
action	DROP, RESET or other service group name
logging_enabled	logging status, enabled(1). disabled(0)
logging_interval	logging interval
logging_failure_only	enabled(1). disabled(0)
over_limit_action	over limit action, drop(0), reset(1)
over_limit_lockup_duration	over limit lockup duration
over_limit_log_interval	over limit log interval
timeout	timeout
ip_limit	XML tag for the collection of IP limiting
class_list	class list name
lid_list	XML tag for the collection of LID
lid	XML tag for LID
id	LID ID
conn_limit	connection limit
conn_rate_limit	connection rate limit
conn_rate_limit_per	connection rate limit interval
req_limit	request limit
req_rate_limit	request rate limit
req_rate_limit_per	request rate limit interval
over_limit_action	over limit action
lockout	lockout
log_status	log status, enabled(1). disabled(0)
log_interval	log interval
client_ip	L3 source IP(0), L3 destination IP(1) or L7 header name(2)
17_head_name	extract client IP address from L7 header
use_dst_ip	use destination IP, enabled(1). disabled(0)
overlap	enabled(1). disabled(0)
share	enabled(1). disabled(0)
full_domain_tree	enabled(1). disabled(0)

Example Response

Request as:

- URL:

- [https://\[AX_IP_ADDRESS\]/services/rest/V2/](https://[AX_IP_ADDRESS]/services/rest/V2/)
- [HTTP Action:](#)
GET

CLI configuration before aXAPI call:

```
!
class-list cl1
!
slb template policy pbslb_template_name1
  bw-list name pl
    bw-list over-limit reset lockup 34 logging 45
    bw-list timeout 67
    bw-list id 1 drop
    bw-list id 2 reset
    bw-list use-destination-ip
  class-list name cl1
  class-list lid 16
    conn-limit 1
    conn-rate-limit 1 per 1
    request-limit 1
    request-rate-limit 1 per 10
    over-limit-action lockout 1
  class-list lid 17
    conn-limit 1
    conn-rate-limit 1 per 1
    request-limit 1
    request-rate-limit 1 per 10
    over-limit-action lockout 1 log 12
!
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxxxx&method=slb.template.pbslb.search&name=pbslb_template_name1

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
  <pbslb_template>
    <name>pbslb_template_name1</name>
    <pbslb>
      <name>pl</name>
      <id_list>
        <id_entry>
          <id>1</id>
          <action>DROP</action>
          <logging_enabled>0</logging_enabled>
          <logging_interval>3</logging_interval>
          <logging_failure_only>0</logging_failure_only>
        </id_entry>
        <id_entry>
          <id>2</id>
          <action>RESET</action>
          <logging_enabled>0</logging_enabled>
          <logging_interval>3</logging_interval>
          <logging_failure_only>0</logging_failure_only>
        </id_entry>
      </id_list>
      <over_limit_action>1</over_limit_action>
      <over_limit_lockup_duration>34</over_limit_lockup_duration>
      <over_limit_log_interval>45</over_limit_log_interval>
      <timeout>67</timeout>
    </pbslb>
    <ip_limit>
      <class_list>cl1</class_list>
      <lids>
        <lid>
          <id>16</id>
          <conn_limit>1</conn_limit>
          <conn_rate_limit>1</conn_rate_limit>
        </lid>
      </lids>
    </ip_limit>
  </pbslb_template>
</response>
```

```

<conn_rate_limit_per>1</conn_rate_limit_per>
<req_limit>1</req_limit>
<req_rate_limit>1</req_rate_limit>
<req_rate_limit_per>10</req_rate_limit_per>
<over_limit_action>0</over_limit_action>
<lockout>1</lockout>
<log_status>0</log_status>
<log_interval>0</log_interval>
</lid>
<lids>
  <id>17</id>
  <conn_limit>1</conn_limit>
  <conn_rate_limit>1</conn_rate_limit>
  <conn_rate_limit_per>1</conn_rate_limit_per>
  <req_limit>1</req_limit>
  <req_rate_limit>1</req_rate_limit>
  <req_rate_limit_per>10</req_rate_limit_per>
  <over_limit_action>0</over_limit_action>
  <lockout>1</lockout>
  <log_status>1</log_status>
  <log_interval>12</log_interval>
</lids>
<client_ip>0</client_ip>
</ip_limit>
<use_dst_ip>1</use_dst_ip>
<overlap>0</overlap>
<share>0</share>
<full_domain_tree>0</full_domain_tree>
</pbslb_template>
</response>

```

Privileges

Config Mode >> Service >> Template >> Application >> PBSLB Policy

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.10.3 “slb.template.pbslb.create” Method

This method is used to create an PBSLB template.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.template.pbslb.create	String		
name (*)	pbslb template name	String		
pbslb	pbslb name	String		
id_list	PBSLB group list to be added group1^B ...groupN ^B: ASCII Code 0x02, URL-encode %02	String		
id_entry<n> id(*) action(*) logging_enabled logging_interval logging_failure_only	group at element <n> group ID DROP, RESET, service group name logging status, enabled(1), disabled(0) logging interval log failure only	Int String Int Int Int	0 -31 0 or 1 0 – 60 0 or 1	0 0 3 0
over_limit_action	over limit action, drop(0), reset(1)	Int	0 or 1	0
over_limit_lockup_duration	over limit lockup duration	Int	1 - 127	

over_limit_log_interval	over limit log interval	Int	1 - 255	
timeout	timeout	Int	1 - 127	5
class_list	class list name	String		
lid_list (*)	LID list to be added lid1^B ... lidN ^B: ASCII Code 0x02, URL-encode %02	String		
lid<n> id (*) conn_limit conn_rate_limit conn_rate_limit_per req_limit req_rate_limit req_rate_limit_per over_limit_action lockout log_status log_interval	LID at element <n> LID ID connection limit connection rate limit connection rate limit per (100ms) request limit request rate limit request rate limit per (100ms) over limit action lockout disabled(0) or enabled(1) log interval	String Int Int Int Int Int Int Int Int Int Int Int Int Int	1 - 31 1 - 1048575 1-2147483647 1 - 65535 1 - 1048575 1-4294967295 1 - 65535 0, 1 or 2 1 - 1023 0 or 1 1 - 255	0 0 0 0
client_ip	I3 source IP(0), I3 destination IP(1) or I7 header name(2)	Int	0, 1 or 2	0
I7_head_name	I7 header name, only when client_ip is 2	String		
use_dst_ip	use destination IP, disabled(0) or enabled(1)	Int	0 or 1	0
overlap	disabled(0) or enabled(1)	Int	0 or 1	0
share	disabled(0) or enabled(1)	Int	0 or 1	0
full_domain_tree	disabled(0) or enabled(1)	Int	0 or 1	0

Notes: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
class-list cl1
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.template.pbslb.create&name=pbslb_temp1_name1&
class_list=c1&client_ip=0&use_des_ip=1&overlap=0&lid_list=lid1%02lid2&lid1=id%031%02conn_
limit%031%02conn_rate_limit%031%02conn_rate_limit_per%031%02req_limit%031%02req_rate_li
mit%031%02req_rate_limit_per%031%02lockout%031&lid2=id%032%02conn_limit%031%02conn_ra
te_limit%031%02conn_rate_limit_per%031%02req_limit%031%02req_rate_limit%031%02req_rate_
limit_per%031%02lockout%031
```

CLI configuration after aXAPI call:

```
!
slb template policy pbslb_temp1_name1
  class-list name cl1
  class-list lid 1
    conn-limit 1
    conn-rate-limit 1 per 1
    request-limit 1
    request-rate-limit 1 per 10
    over-limit-action lockout 1
  class-list lid 2
    conn-limit 1
    conn-rate-limit 1 per 1
```

```

request-limit 1
request-rate-limit 1 per 10
over-limit-action lockout 1
!
```

Response as the HTTP body:

```

<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Privileges

Config Mode >> Service >> Template >> Application >> PBSLB Policy
 Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.10.4 “slb.template.pbslb.update” Method

This method is used to update one or more parameters in a PBSLB template.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.template.pbslb.update	String		
name (*)	pbslb template name	String		
pbslb				
id_list	PBSLB group list to be added group1^B ...groupN ^B: ASCII Code 0x02, URL-encode %02	String		
id_entry<n> id(*) action(*) logging_enabled logging_interval logging_failure_only	group at element <n> group ID DROP, RESET, service group name logging status, enabled(1), disabled(0) logging interval log failure only	Int String Int Int Int Int	0 -31 0 or 1 0 – 60 0 or 1	0 0 3 0
over_limit_action	over limit action, drop(0), reset(1)	Int	0 or 1	0
over_limit_lockup_duration	over limit lockup duration	Int	1 - 127	
over_limit_log_interval	over limit log interval	Int	1 - 255	
timeout	timeout	Int	1 - 127	5
class_list				
lid_list (*)	LID list to be added lid1^B ...lidN ^B: ASCII Code 0x02, URL-encode %02	String		
lid<n> id (*) conn_limit conn_rate_limit conn_rate_limit_per req_limit req_rate_limit req_rate_limit_per over_limit_action lockout log_status log_interval	LID at element <n> LID ID connection limit connection rate limit connection rate limit per (100ms) request limit request rate limit request rate limit per (100ms) over limit action lockout disabled(0) or enabled(1) log interval	String Int Int Int Int Int Int Int Int Int Int Int	1 - 31 1 –1048575 1-2147483647 1 – 65535 1 –1048575 1-4294967295 1 – 65535 0, 1 or 2 1 – 1023 0 or 1 1 - 255	0 0 0 0 0 0 0 0 0 0 0 0
client_ip	I3 source IP(0), I3 destination IP(1) or I7 header	Int	0, 1 or 2	0

	name(2)			
I7_head_name	I7 header name, only when client_ip is 2	String		
use_dst_ip	use destination IP, disabled(0) or enabled(1)	Int	0 or 1	0
overlap	disabled(0) or enabled(1)	Int	0 or 1	0
share	disabled(0) or enabled(1)	Int	0 or 1	0
full_domain_tree	disabled(0) or enabled(1)	Int	0 or 1	0

Notes: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
class-list cl1
!
slb template policy pbslb_tmpl_name1
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxxxx&method=slb.template.pbslb.update&name=pbslb_tmpl_name1&class_list=c1&client_ip=0&use_des_ip=1&overlap=0&lid_list=lid1%02lid2&lid1=id%031%02conn_limit%031%02conn_rate_limit%031%02conn_rate_limit_per%031%02req_limit%031%02req_rate_limit%031%02req_rate_limit_per%0310%02lockout%031&lid2=id%032%02conn_limit%031%02conn_rate_limit%031%02conn_rate_limit_per%031%02req_limit%031%02req_rate_limit%031%02req_rate_limit_per%0310%02lockout%031
```

CLI configuration after aXAPI call:

```
class-list cl1
!
slb template policy pbslb_tmpl_name1
  class-list name c1
    class-list lid 1
      conn-limit 1
      conn-rate-limit 1 per 1
      request-limit 1
      request-rate-limit 1 per 10
      over-limit-action lockout 1
    class-list lid 2
      conn-limit 1
      conn-rate-limit 1 per 1
      request-limit 1
      request-rate-limit 1 per 10
      over-limit-action lockout 1
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Privileges

Config Mode >> Service >> Template >> Application >> PBSLB Policy

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.10.5 “slb.template.pbslb.delete” Method

This method is used to delete a PBSLB template.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.template.pbslb.delete	String		
name (*)	pbslb template name	String		

Notes: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
slb template policy pbslb_tmpl_name1
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.template.pbslb.delete&name=pbslb_tmpl_name1

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Privileges

Config Mode >> Service >> Template >> Application >> PBSLB Policy

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.11 SMTP Template

7.11.1 “slb.template.smtp.getAll” Method

This method is used to get all configuration information for all SMTP templates configured on the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.template.smtp.getAll	String		

Note: (*) parameter is required in the API.

Response Fields

smtp_template_list	XML tag for the collection of smtp templates
smtp_template	XML tag for smtp template
name	SMTP template name
starttls	STARTTLS, disabled(0), enforced(2) or optional(1)
EXPN	SMTP command EXPN, enabled(0) or disabled(1)
TURN	SMTP command TURN, enabled(0) or disabled(1)
VRFY	SMTP command VRFY, enabled(0) or disabled(1)
server_domain	server domain
service_ready_message	service ready message
client_domain_switching_list	XML tag for the collection of client switchings
client_domain_switching	XML tag for client domain switching
client_domain	client domain
service_group	service group name
match_type	match type, contains(0), starts with(1), ends with(2)

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
slb template smtp smtp_template_name1
    client-domain-switching contains domain1 service-group src
    client-domain-switching contains domain2 service-group src
!
slb template smtp smtp_template_name2
    starttls optional
    command-disable expn turn vrfy
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.template.smtp.getAll

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
    <smtp_template_list>
```

```

<smtp_template>
  <name>smtp_template_name1</name>
  <starttls>0</starttls>
  <EXPN>0</EXPN>
  <TURN>0</TURN>
  <VRFY>0</VRFY>
  <server_domain>abc</server_domain>
  <service_ready_message>123</service_ready_message>
  <client_domain_switching_list>
    <client_domain_switching>
      <client_domain>1</client_domain>
      <service_group>src</service_group>
      <match_type>0</match_type>
    </client_domain_switching>
    <client_domain_switching>
      <client_domain>2</client_domain>
      <service_group>src</service_group>
      <match_type>0</match_type>
    </client_domain_switching>
  </client_domain_switching_list>
</smtp_template>
<smtp_template>
  <name>smtp_template_name2</name>
  <starttls>1</starttls>
  <EXPN>1</EXPN>
  <TURN>1</TURN>
  <VRFY>1</VRFY>
  <server_domain></server_domain>
  <service_ready_message></service_ready_message>
  <client_domain_switching_list></client_domain_switching_list>
</smtp_template>
</smtp_template_list>
</response>

```

Menus Privilege

Config Mode >> Service >> Template >> Application >> SMTP

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.11.2 “slb.template.smtp.search” Method

This method is used to get SMTP template information for a given SMTP template name.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.template.smtp.search	String		
name (*)	smtp template name	String		

Note: (*) parameter is required in the API.

Response Fields

smtp_template	XML tag for smtp template
name	SMTP template name
starttls	STARTTLS, disabled(0), enforced(2) or optional(1)
EXPN	SMTP command EXPN, enabled(0) or disabled(1)
TURN	SMTP command TURN, enabled(0) or disabled(1)
VRFY	SMTP command VRFY, enabled(0) or disabled(1)
server_domain	server domain
service_ready_message	service ready message

client_domain_switching_list	XML tag for the collection of client switchings
client_domain_switching	XML tag for client domain switching
client_domain	client domain
service_group	service group name
match_type	match type, contains(0), starts with(1), ends with(2)

Example Response

Request as:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
- **HTTP Action:**
GET

CLI configuration before aXAPI call:

```
!
slb template smtp smtp_template_name1
    client-domain-switching contains domain1 service-group src
    client-domain-switching contains domain2 service-group src
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxxxx&method=slb.template.smtp.search&name=smtp_template_name1

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
    <smtp_template>
        <name>smtp_template_name1</name>
        <starttls>0</starttls>
        <EXPN>0</EXPN>
        <TURN>0</TURN>
        <VRFY>0</VRFY>
        <server_domain>abc</server_domain>
        <service_ready_message>123</service_ready_message>
        <client_domain_switching_list>
            <client_domain_switching>
                <client_domain>1</client_domain>
                <service_group>src</service_group>
                <match_type>0</match_type>
            </client_domain_switching>
            <client_domain_switching>
                <client_domain>2</client_domain>
                <service_group>src</service_group>
                <match_type>0</match_type>
            </client_domain_switching>
        </client_domain_switching_list>
    </smtp_template>
</response>
```

Menus Privilege

Config Mode >> Service >> Template >> Application >> SMTP

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.11.3 “slb.template.smtp.create” Method

This method is used to create an SMTP template.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.template.smtp.create	String		
name (*)	smtp template name	String		
starttls	Disabled(0), Enforced(2) or Optional(1)	Int	0, 1 or 2	0
EXPN	SMTP command EXPN, enabled(0), or disabled(1)	Int	0 or 1	0
TURN	SMTP command TURN, enabled(0), or disabled(1)	int	0 or 1	0
VRFY	SMTP command VRFY, enabled(0), or disabled(1)	int	0 or 1	0
server_domain	server domain	String		
service_ready_message	service ready message	String		
client_domain_switching_list (*)	switch list to be added switch1^B ...switchN ^B: ASCII Code 0x02, URL-encode %02	String		
client_domain_switching<n> client_domain (*) service_group match_type	switch at element <n> client domain service group name match type, contains(0), starts with(1), ends with(2)	String String Int	0, 1 or 2	0

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
slb service-group srvgrp1 tcp
!
slb service-group srvgrp2 tcp
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.template.smtp.create&name=smtp_template_name1&
starttls=1&EXPN=1&TURN=1&VRFY=1&client_domain_switching_list=switch1%02switch2&switch1=cl
ient_domain%03domain1%02service_group%03srvgrp1%02match_type%030&switch2=client_domain%
03domain2%02service_group%03srvgrp2%02match_type%031
```

CLI configuration after aXAPI call:

```
!
slb service-group srvgrp1 tcp
!
slb service-group srvgrp2 tcp
!
slb template smtp smtp_template_name1
  starttls optional
  command-disable expn turn vrfy
  client-domain-switching contains domain1 service-group srvgrp1
  client-domain-switching contains domain2 service-group srvgrp2
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> Template >> Application >> SMTP

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.11.4 “slb.template.smtp.update” Method

This method is used to update one or more parameters in an SMTP template.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.template.smtp.update	String		
name (*)	smtp template name	String		
starttls	Disabled(0), Enforced(2) or Optional(1)	Int	0, 1 or 2	0
EXPN	SMTP command EXPN, enabled(0), or disabled(1)	Int	0 or 1	0
TURN	SMTP command TURN, enabled(0), or disabled(1)	int	0 or 1	0
VRFY	SMTP command VRFY, enabled(0), or disabled(1)	int	0 or 1	0
server_domain	server domain	String		
service_ready_message	service ready message	String		
client_domain_switching_list (*)	switch list to be added switch1^B ...switchN ^B: ASCII Code 0x02, URL-encode %02	String		
client_domain_switching<n> client_domain (*) service_group match_type	switch at element <n> client domain service group name match type, contains(0), starts with(1), ends with(2)	String String Int	0, 1 or 2	0

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
- **HTTP Action:**
POST

CLI configuration before aXAPI call:

```
!
slb service-group srvgrp1 tcp
!
slb service-group srvgrp2 tcp
!
!
```

```

slb template smtp smtp_template_name1
  starttls optional
  command-disable expn turn vrfy
!
```

HTTP Body

```

session_id=xxxxxxxxxxxxxxxxxxxxxx&method=slb.template.smtp.update&name=smtp_template_name1&
starttls=1&EXPN=1&TURN=1&VRFY=1&client_domain_switching_list=switch1%02switch2&switch1=cl
ient_domain%03domain1%02service_group%03srvgrp1%02match_type%030&switch2=client_domain%
03domain2%02service_group%03srvgrp2%02match_type%031
```

CLI configuration after aXAPI call:

```

!
slb service-group srvgrp1 tcp
!
slb service-group srvgrp2 tcp
!
slb template smtp smtp_template_name1
  starttls optional
  command-disable expn turn vrfy
  client-domain-switching contains domain1 service-group srvgrp1
  client-domain-switching contains domain2 service-group srvgrp2
!
```

Response as the HTTP body:

```

<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> Template >> Application >> SMTP

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.11.5 “slb.template.smtp.delete” Method

This method is used to delete an SMTP template.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.template.smtp.delete	String		
name (*)	smtp template name	String		

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
slb template smtp smtp_template_name1
  starttls optional
  command-disable expn turn vrfy
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.template.smtp.delete&name=smtp_template_name1
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> Template >> Application >> SMTP

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.12 SIP Template

7.12.1 “**slb.template.sip.getAll**” Method

This method is used to get configuration information for all SIP templates configured on the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.template.sip.getAll	String		

Note: (*) parameter is required in the API.

Response Fields

sip_template_list	XML tag for the collection of sip templates
sip_template	XML tag for sip template
name	sip template name
header_erase	the header to erase
header_insert	the header to insert
header_replace_from	header replace from
header_replace_to	header replace to
registrar_service_group	service group name
timeout	timeout
pass_server_ip_for_acl	pass real server IP for ACL
client_keep_alive	client keep alive, disabled(0) or enabled(1)
server_keep_alive	server keep alive, disabled(0) or number > 0

insert_client_ip	insert client IP, disabled(0) or enabled(1)
select_client_fail_action	client fail action, do nothing(0), drop(1), send message(2)
select_client_fail_message	message to send, when select_client_fail_action is 2
select_server_fail_action	server fail action, do nothing(0), drop(1), send message(2)
select_server_fail_message	message to send when select server failed. (only when select_server_fail_action is 2)
exclude_translation_body	exclude translation body, disabled(0) or enabled(1)
exclude_translation_start_line	exclude translation start line, disabled(0) or enabled(1)
exclude_translation_header_list	XML tag for the collection of headers, only when status is enable(1)
exclude_translation_header	header name

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
slb service-group udp udp
!
slb template sip sip_template_name1
  registrar service-group udp
  header-insert client_ip-address:sddssdds
  header-erase sdadsadadasdadsadsadsa
  header-replace SSSSS AAAAA
  insert-client-ip
  client-keep-alive
  server-keep-alive
  exclude-translation start-line
  exclude-translation header AAAAA
  exclude-translation body
  select-client-fail sdsds
  select-server-fail sddsdss
slb template sip sip_template_name2
  registrar service-group udp
  header-insert TEST:UVWXYZ
  header-erase ABCDEF
  header-replace ABC UDF
  insert-client-ip
  select-client-fail drop
  select-server-fail drop
  exclude-translation start-line
  exclude-translation header CLIENT
  exclude-translation header NAME
  exclude-translation body
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.template.sip.getAll

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
  <sip_template_list>
    <sip_template>
      <name> sip_template_name1</name>
```

```

<header_erase>sadadsadadasdadsadsa</header_erase>
<header_insert>client_ip-address:sddssdds</header_insert>
<header_replace_from>SSSSS</header_replace_from>
<header_replace_to>AAAAA</header_replace_to>
<registrar_service_group>udp</registrar_service_group>
<timeout>30</timeout>
<pass_server_ip_for_acl>0</pass_server_ip_for_acl>
<client_keep_alive>1</client_keep_alive>
<server_keep_alive>30</server_keep_alive>
<insert_client_ip>1</insert_client_ip>
<select_client_fail_action>2</select_client_fail_action>
<select_client_fail_message>sdsds</select_client_fail_message>
<select_server_fail_action>2</select_server_fail_action>
<select_server_fail_message>sddsd</select_server_fail_message>
<exclude_translation_body>1</exclude_translation_body>
<exclude_translation_start_line>1</exclude_translation_start_line>
<exclude_translation_header_list>
    <exclude_translation_header>AAAAA</exclude_translation_header>
</exclude_translation_header_list>
</sip_template>
</sip_template>
    <name> sip_template_name2</name>
    <header_erase>ABCDEF</header_erase>
    <header_insert>TEST:UVWXYZ</header_insert>
    <header_replace_from>ABC</header_replace_from>
    <header_replace_to>UDF</header_replace_to>
    <registrar_service_group>udp</registrar_service_group>
    <timeout>30</timeout>
    <pass_server_ip_for_acl>0</pass_server_ip_for_acl>
    <client_keep_alive>0</client_keep_alive>
    <server_keep_alive>0</server_keep_alive>
    <insert_client_ip>1</insert_client_ip>
    <select_client_fail_action>1</select_client_fail_action>
    <select_server_fail_action>1</select_server_fail_action>
    <exclude_translation_body>1</exclude_translation_body>
    <exclude_translation_start_line>1</exclude_translation_start_line>
    <exclude_translation_header_list>
        <exclude_translation_header>CLIENT</exclude_translation_header>
        <exclude_translation_header>NAME</exclude_translation_header>
    </exclude_translation_header_list>
</sip_template>
</sip_template_list>
</response>

```

Menus Privilege

Config Mode >> Service >> Template >> Application >> SIP

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.12.2 “slb.template.sip.search” Method

This method is used to get SIP template information by given SIP template name.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.template.sip.search	String		
name (*)	sip template name	String		

Note: (*) parameter is required in the API.

Response Fields

sip_template	XML tag for sip template
name	sip template name
header_erase	the header to erase
header_insert	the header to insert
header_replace_from	header replace from
header_replace_to	header replace to
registrar_service_group	service group name
timeout	timeout
pass_server_ip_for_acl	pass real server IP for ACL
client_keep_alive	client keep alive, disabled(0) or enabled(1)
server_keep_alive	server keep alive, disabled(0) or number > 0
insert_client_ip	insert client IP, disabled(0) or enabled(1)
select_client_fail_action	client fail action, do nothing(0), drop(1), send message(2)
select_client_fail_message	message to send, when select_client_fail_action is 2
select_server_fail_action	server fail action, do nothing(0), drop(1), send message(2)
select_server_fail_message	message to send when select server failed. (only when select_server_fail_action is 2)
exclude_translation_body	exclude translation body, disabled(0) or enabled(1)
exclude_translation_start_line	exclude translation start line, disabled(0) or enabled(1)
exclude_translation_header_list	XML tag for the collection of headers, only when status is enabled(1)
exclude_translation_header	header name

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
slb service-group udp udp
!
slb template sip sip_template_name2
    registrar service-group udp
    header-insert TEST:UVWXYZ
    header-erase ABCDEF
    header-replace ABC UDF
    insert-client-ip
    select-client-fail drop
    select-server-fail drop
    exclude-translation start-line
    exclude-translation header CLIENT
    exclude-translation header NAME
    exclude-translation body
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxxxx&method=slb.template.sip.search&name=sip_02

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
```

```

<sip_template>
  <name>sip_02</name>
  <header_erase>ABCDEF</header_erase>
  <header_insert>TEST:UVWXYZ</header_insert>
  <header_replace_from>ABC</header_replace_from>
  <header_replace_to>UDF</header_replace_to>
  <registrar_service_group>udp</registrar_service_group>
  <timeout>30</timeout>
  <pass_server_ip_for_acl>0</pass_server_ip_for_acl>
  <client_keep_alive>0</client_keep_alive>
  <server_keep_alive>0</server_keep_alive>
  <insert_client_ip>1</insert_client_ip>
  <select_client_fail_action>1</select_client_fail_action>
  <select_server_fail_action>1</select_server_fail_action>
  <exclude_translation_body>1</exclude_translation_body>
  <exclude_translation_start_line>1</exclude_translation_start_line>
  <exclude_translation_header_list>
    <exclude_translation_header>CLIENT</exclude_translation_header>
    <exclude_translation_header>NAME</exclude_translation_header>
  </exclude_translation_header_list>
</sip_template>
</response>

```

Menus Privilege

Config Mode >> Service >> Template >> Application >> SIP

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.12.3 “slb.template.sip.create” Method

This method is used to create a SIP template.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.template.sip.create	String		
name (*)	sip template name	String		
header_erase	header erase	String		
header_insert	header insert	String		
header_replace_from	header replace from	String		
header_replace_to	header replace to	String		
registrar_service_group	registrar service group name	String		
timeout	timeout (minute)	Int	1 – 250	30
pass_server_ip_for_acl	pass real server IP for ACL	int	100 - 199	
client_keep_alive	client keep alive, disabled(0) or enabled(1)	Int	0 or 1	0
server_keep_alive	server keep alive, disabled(0) or enabled(1)	Int	0,1 and 5-300	0
insert_client_ip	insert client IP, disabled(0) or enabled(1)	Int	0 or 1	0
select_client_fail_action	client fail action, drop(0) or send message(1)	Int	0, 1 or 2	0
select_client_fail_message	client fail action send message	String		
select_server_fail_action	server fail action, drop(0) or send message(1)	Int	0, 1 or 2	0
select_server_fail_message	server fail action send message	String		
exclude_translation_body	exclude translation body, disabled(0) or enabled(1)	Int	0 or 1	0
exclude_translation_start_line	exclude translation start line, disabled(0) or enabled(1)	Int	0 or 1	0
exclude_translation_header_list (*)	header list to be added header1^B ...headerN ^B: ASCII Code 0x02, URL-encode %02	String		
exclude_translation_header(*)	header name at element <n> header name	String		

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
slb service-group udp udp
!
slb template sip sip_template_name1
    registrar service-group udp
    header-insert aaa:aa
    header-erase 12345678901234567890
    header-replace asdfghjkl qwertyuiop
    timeout 60
    insert-client-ip
    client-keep-alive
    server-keep-alive 6
    select-client-fail cccccccccccc
    select-server-fail dddddddddd
    exclude-translation start-line
    exclude-translation header aaaaaaaa
    exclude-translation header rrrrrrrrrr
    exclude-translation body
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxxxx&method=slb.template.sip.create&name=sip_template_name1
&header_erase=12345678901234567890&header_insert=aaa:aa&header_replace_from=asdfghjkl&
header_replace_to=qwertyuiop&registrar_service_group=udp&timeout=60&pass_server_ip_for
_acl=101&client_keep_alive=1&server_keep_alive=6&insert_client_ip=1&select_client_fail
_action=2&select_server_fail_action=2&select_server_fail_message=ddddddddd&select_cli
ent_fail_message=ccccccccc&exclude_translation_body=1&exclude_translation_start_lin
e=1&exclude_translation_header_list=a1%02a2&a1=exclude_translation_header%03aaaaaaaa&a2
=exclude_translation_header%03rrrrrrrrrrr
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> Template >> Application >> SIP

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.12.4 “slb.template.sip.update**” Method**

This method is used to update one or more parameters in a Session Initiation Protocol (SIP) template.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.template.sip.update	String		
name (*)	sip template name	String		
header_erase	header erase	String		
header_insert	header insert	String		
header_replace_from	header replace from	String		
header_replace_to	header replace to	String		
registrar_service_group	registrar service group name	String		
Timeout	timeout (minute)	Int	1 – 250	30
pass_server_ip_for_acl	pass real server IP for ACL	int	100 -199	
client_keep_alive	client keep alive, disabled(0) or enabled(1)	Int	0 or 1	0
server_keep_alive	server keep alive, disabled(0) or enabled(1)	Int	0 or 1	0
insert_client_ip	insert client IP, disabled(0) or enabled(1)	Int	0 or 1	0
select_client_fail_action	client fail action, drop(0) or send message(1)	Int	0, 1 or 2	0
select_client_fail_message	client fail action send message	String		
select_server_fail_action	server fail action, drop(0) or send message(1)	Int	0, 1 or 2	0
select_server_fail_message	server fail action send message	String		
Exclude_translation_body	exclude translation body, disabled(0) or enabled(1)	Int	0 or 1	0
Exclude_translation_start_line	exclude translation start line, disabled(0) or enabled(1)	Int	0 or 1	0
Exclude_translation_header_list (*)	header list to be added header1^B ...headerN ^B: ASCII Code 0x02, URL-encode %02	String		
Exclude_translation_header(*)	header name at element <n> header name	String		

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
slb service-group udp udp
!
slb template sip sip_template_name1
  registrar service-group udp
  header-insert TEST:UVWXYZ
  header-erase ABCDEF
  header-replace ABC UDF
  insert-client-ip
  select-client-fail drop
  select-server-fail drop
  exclude-translation start-line
  exclude-translation header CLIENT
  exclude-translation header NAME
  exclude-translation body
!
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.template.sip.update&name=sip_template_name1&header_erase=&header_insert=&header_replace_from=&header_replace_to=&registrar_service_group=&timeout=1&pass_server_ip_for_acl=&client_keep_alive=0&server_keep_alive=0&insert_client_ip=0&select_client_fail_action=2&select_server_fail_action=2&select_server_fail_message=ddddddddd&select_client_fail_message=ccccccccccc&exclude_translation_body=0&exclude_tr
```

```
anslation_start_line=0&exclude_translation_header_list=a1%02a2%02a3&a1=exclude_translatio
n_header%03aaaaaaaa&a2=exclude_translation_header%03rrrrrrrrrrr&a3=exclude_translation_he
ader%03qwerty
```

CLI configuration after aXAPI call:

```
!
slb service-group udp udp
!
slb template sip sip_template_name1
    timeout 1
    select-client-fail cccccccccccc
    select-server-fail dddddddddd
    exclude-translation header aaaaaaa
    exclude-translation header rrrrrrrrrrr
    exclude-translation header qwerty
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> Template >> Application >> SIP

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.12.5 “slb.template.sip.delete” Method

This method is used to delete a Session Initiation Protocol (SIP) template.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.template.sip.delete	String		
name (*)	sip template name	String		

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
- **HTTP Action:**
POST

CLI configuration before aXAPI call:

```
!
slb service-group udp udp
!
slb template sip test26
```

```

timeout 1
select-client-fail cccccccccc
select-server-fail dddddddddd
exclude-translation header aaaaaaa
exclude-translation header rrrrrrrrrr
exclude-translation header qwerty
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.template.sip.delete&name=sip_tmpl_name1

Response as the HTTP body:

```

<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> Template >> Application >> SIP

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.13 RTSP Template

7.13.1 “**slb.template.rtsp.getAll**” Method

This method is used to get configuration information for all RTSP templates configured on the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.template.rtsp.getAll	String		

Note: (*) parameter is required in the API.

Response Fields

rtsp_template_list	XML tag for the collection of rtsp templates
rtsp_template	XML tag for rtsp template
name	rtsp template name
uri_list	XML tag for the collection of URIs
uri	XML tag for URI
uri	URI name
service_group	service group name

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```

!
slb service-group http tcp
!
slb service-group http http_srv_grp1
!
slb template streaming-media rtsp_template_name1
    uri-switching stream sdddsds service-group http
slb template streaming-media rtsp_template_name2
    uri-switching stream FFFF service-group http
    uri-switching stream VVVVV service-group http_srv_grp1
!

```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxxxx&method=slb.template.rtsp.getAll

Response as the HTTP body:

```

<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
    <rtsp_template_list>
        <rtsp_template>
            <name>rtsp_template_name1</name>
            <uri_list>
                <uri>
                    <uri>sdddsds</uri>
                    <service_group>http</service_group>
                </uri>
            </uri_list>
        </rtsp_template>
        <rtsp_template>
            <name>rtsp_template_name2</name>
            <uri_list>
                <uri>
                    <uri>FFFF</uri>
                    <service_group>http</service_group>
                </uri>
                <uri>
                    <uri>VVVV</uri>
                    <service_group>http_srv_grp1</service_group>
                </uri>
            </uri_list>
        </rtsp_template>
    </rtsp_template_list>
</response>

```

Menus Privilege

Config Mode >> Service >> Template >> Application >> RTSP

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.13.2 “slb.template.rtsp.search” Method

This method is used to get RTSP template information using a given RTSP template name.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.template.rtsp.search	String		
name(*)	rtsp template name	String		

Note: (*) parameter is required in the API.

Response Fields

rtsp_template	XML tag for rtsp template
name	rtsp template name
uri_list	XML tag for the collection of URIs
uri	XML tag for URI
uri	URI name
service_group	service group name

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
slb service-group http tcp
!
slb service-group http http_srv_grp1
!
slb template streaming-media rtsp_template_name1
  uri-switching stream sdddsds service-group http
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.template.rtsp.search&name=rtsp_template_name1
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
  <rtsp_template>
    <name>rtsp_template_name1</name>
    <uri_list>
      <uri>
        <uri>sdddsds</uri>
        <service_group>http</service_group>
      </uri>
    </uri_list>
  </rtsp_template>
</response>
```

Menus Privilege

Config Mode >> Service >> Template >> Application >> RTSP

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.13.3 “slb.template.rtsp.create” Method

This method is used to create an RTSP template.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.template.rtsp.create	String		

name (*)	rtsp template name	String		
uri_list (*)	uri list to be added uri1^B ...uriN ^B: ASCII Code 0x02, URL-encode %02	String		
uri<n> uri (*) service_group (*)	uri at element <n> uri name service group name	String String		

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.template.rtsp.create&name=rtsp_template_name1
e1&uri_list=11%0212&11=uri%03aaa%02service_group%03http&l2=uri%03bbb%02service_group%03http
```

CLI configuration after aXAPI call:

```
!
slb service-group http tcp
!
slb template streaming-media rtsp_template_name1
    uri-switching stream aaa service-group http
    uri-switching stream bbb service-group http
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> Template >> Application >> RTSP

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.13.4 “**slb.template.rtsp.update**” Method

This method is used to update one or more parameters in a RTSP template.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.template.rtsp.update	String		
name (*)	rtsp template name	String		
uri_list (*)	uri list to be added uri1^B ...uriN ^B: ASCII Code 0x02, URL-encode %02	String		

uri<n> uri (*) service_group (*)	uri at element <n> uri name service group name	String String		
---	--	------------------	--	--

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
slb service-group http tcp
!
slb template streaming-media rtsp_template_name1
    uri-switching stream sdddsds service-group http
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxxxx&method=slb.template.rtsp.update&name=rtsp_template_name1&
uri_list=l1%0212&l1=uri%03aaa%02service_group%03http&l2=uri%03bbb%02service_group%03http
```

CLI configuration after aXAPI call:

```
!
slb service-group http tcp
!
slb template streaming-media rtsp_template_name1
    uri-switching stream aaa service-group http
    uri-switching stream bbb service-group http
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> Template >> Application >> RTSP

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.13.5 “slb.template.rtsp.delete” Method

This method is used to delete an RTSP template.

Parameters

Parameter Name	Description	Data Type	Range	Default
----------------	-------------	-----------	-------	---------

session_id (*)	user authenticated session id	String		
method (*)	slb.template.rtsp.delete	String		
name (*)	rtsp template name	String		

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
- **HTTP Action:**
POST

CLI configuration before aXAPI call:

```
!
slb service-group http tcp
!
slb template streaming-media rtsp_template_name1
    uri-switching stream sdddsds service-group http
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.template.rtsp.delete&name=rtsp_template_name1

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> Template >> Application >> RTSP

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.14 Connection-Reuse Template

7.14.1 “slb.template.conn_reuse.getAll**” Methods**

This method is used to get configuration information for all connection reuse templates configured on the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.template.conn_reuse.getAll	String		

Note: (*) parameter is required in the API.

Response Fields

conn_reuse_template_list	XML tag for the collection of connection reuse templates
conn_reuse_template	XML tag for connection reuse template

name	connection reuse template name
limit_per_srv	limit per server
timeout	timeout
keep_alive_conn	keep alive connection

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
slb template connection-reuse conn_reuse_template_name1
!
slb template connection-reuse conn_reuse_template_name2
    timeout 240
    limit-per-server 100
    keep-alive-conn 10
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxxxx&method=slb.template.conn_reuse.getAll

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
<conn_reuse_template_list>
    <conn_reuse_template>
        <name>conn_reuse_template_name1</name>
        <limit_per_srv>1000</limit_per_srv>
        <timeout>2400</timeout>
        <keep_alive_conn>0</keep_alive_conn>
    </conn_reuse_template>
    <conn_reuse_template>
        <name>conn_reuse_template_name2</name>
        <limit_per_srv>100</limit_per_srv>
        <timeout>240</timeout>
        <keep_alive_conn>10</keep_alive_conn>
    </conn_reuse_template>
</conn_reuse_template_list>
</response>
```

Menus Privilege

Config Mode >> Service >> Template >> Connection Reuse

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.14.2 “`slb.template.conn_reuse.search`” Method

This method is used to get connection reuse template information by given connection reuse template name.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		

method (*)	slb.template.conn_reuse.search	String		
name	connection reuse template name	String		

Note: (*) parameter is required in the API.

Response Fields

conn_reuse_template	XML tag for connection reuse template
name	connection reuse template name
limit_per_srv	limit per server
timeout	timeout
keep_alive_conn	keep alive connection

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
slb template connection-reuse conn_reuse_template_name1
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.template.conn_reuse.search&name=conn_reuse_template_name1
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
  <conn_reuse_template>
    <name>conn_reuse_template_name1</name>
    <limit_per_srv>1000</limit_per_srv>
    <timeout>2400</timeout>
    <keep_alive_conn>0</keep_alive_conn>
  </conn_reuse_template>
</response>
```

Menus Privilege

Config Mode >> Service >> Template >> Connection Reuse

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.14.3 “slb.template.conn_reuse.create” Method

This method is used to create a connection reuse template.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.template.conn_reuse.create	String		
name (*)	connection reuse template name	String		
limit_per_srv	limit per server	Int	0 – 65535	1000
timeout	Timeout	Int	1 – 3600	2400

<code>keep_alive_conn</code>	keep alive connection	Int	0 – 1024	0
------------------------------	-----------------------	-----	----------	---

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
`https://[AX_IP_ADDRESS]/services/rest/V2/`
- HTTP Action:
POST

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxxxx&method=slb.template.conn_reuse.create&name=conn_reuse_
templ_name1&limit_per_srv=1000&timeout=2400&keep_alive_conn=100
```

CLI configuration after aXAPI call:

```
!
slb template connection-reuse conn_reuse_template_name1
    keep-alive-conn
!!
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> Template >> Connection Reuse

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.14.4 “`slb.template.conn_reuse.update`” Method

This method is used to update one or more parameters in a connection reuse template.

Parameters

Parameter Name	Description	Data Type	Range	Default
<code>session_id</code> (*)	user authenticated session id.	String		
<code>method</code> (*)	<code>slb.template.conn_reuse.update</code>	String		
<code>name</code> (*)	connection reuse template name	String		
<code>limit_per_srv</code>	limit per server	Int	0 – 65535	
<code>timeout</code>	Timeout	Int	1 – 3600	
<code>keep_alive_conn</code>	keep alive connection	Int	0 – 1024	

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
slb template connection-reuse conn_reuse_template_name1
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxxxx&method=slb.template.conn_reuse.update&name=conn_reuse_template_name1&limit_per_srv=1000&timeout=2400&keep_alive_conn=100
```

CLI configuration after aXAPI call:

```
!
slb template connection-reuse conn_reuse_template_name1
    keep-alive-conn
!!
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> Template >> Connection Reuse

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.14.5 “**slb.template.conn_reuse.delete**” Method

This method is used to delete a connection reuse template.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.template.conn_reuse.delete	String		
name (*)	connection reuse template name	String		

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
```

```
slb template connection-reuse conn_reuse_template_name1  
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxxxx&method=slb.template.conn_reuse.delete&name=conn_reuse_template_name1
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>  
<response status="ok">  
</response>
```

Menus Privilege

Config Mode >> Service >> Template >> Connection Reuse

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.15 TCP Template

7.15.1 “**slb.template.tcp.getAll**” Method

This method is used to get configuration information for all TCP templates configured on the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.template.tcp.getAll	String		

Note: (*) parameter is required in the API.

Response Fields

tcp_template_list	XML tag for the collection of tcp templates
tcp_template	XML tag for tcp template
name	tcp template name
idle_timeout	idle timeout
force_del_timeout	force delete timeout (Second)
init_win_size	initial window size
half_close_idle_timeout	disabled(0) or number > 0 (Second)
reset_fwd	reset forward, disabled(0) or enabled(1)
reset_rec	reset receive, disabled(0) or enabled(1)

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```

!
slb template tcp tcp_template_name1
    force-delete-timeout 8
    half-close-idle-timeout 100
!
slb template tcp tcp_template_name 2
    idle-timeout 240
    half-close-idle-timeout 100
    force-delete-timeout 8
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxxxx&method=slb.template.tcp.getAll

Response as the HTTP body:

```

<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
    <tcp_template_list>
        <tcp_template>
            <name>tcp_template_name1</name>
            <idle_timeout>120</idle_timeout>
            <force_del_timeout>8</force_del_timeout>
            <init_win_size>0</init_win_size>
            <half_close_idle_timeout>100</half_close_idle_timeout>
            <reset_fwd>0</reset_fwd>
            <reset_rec>0</reset_rec>
        </tcp_template>
        <tcp_template>
            <name>tcp_template_name 2</name>
            <idle_timeout>240</idle_timeout>
            <force_del_timeout>8</force_del_timeout>
            <init_win_size>0</init_win_size>
            <half_close_idle_timeout>100</half_close_idle_timeout>
            <reset_fwd>1</reset_fwd>
            <reset_rec>1</reset_rec>
        </tcp_template>
    </tcp_template_list>
</response>
```

Menus Privilege

Config Mode >> Service >> Template >> L4 >> TCP

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.15.2 “**slb.template.tcp.search**” Method

This method is used to get TCP template information by given TCP template name.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.template.tcp.search	String		
name	tcp template name	String		

Note: (*) parameter is required in the API.

Response Fields

tcp_template name	XML tag for tcp template tcp template name
----------------------	---

idle_timeout	idle timeout
force_del_timeout	force delete timeout, disabled(0) or number > 0 (Second)
init_win_size	initial window size
half_close_idle_timeout	disabled(0) or number > 0 (Second)
reset_fwd	reset forward, disabled(0) or enabled(1)
reset_rec	reset receive, disabled(0) or enabled(1)

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
slb template tcp tcp_template_name1
  force-delete-timeout 8
  half-close-idle-timeout 100
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxxxx&method=slb.template.tcp.search&name=tcp_template_name1

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
  <tcp_template>
    <name>tcp_template_name1</name>
    <idle_timeout>120</idle_timeout>
    <force_del_timeout>8</force_del_timeout>
    <init_win_size>0</init_win_size>
    <half_close_idle_timeout>100</half_close_idle_timeout>
    <reset_fwd>0</reset_fwd>
    <reset_rec>0</reset_rec>
  </tcp_template>
</response>
```

Menus Privilege

Config Mode >> Service >> Template >> L4 >> TCP

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.15.3 “slb.template.tcp.create” Method

This method is used to create a TCP template.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.template.tcp.create	String		
name (*)	tcp template name	String		
idle_timeout	Idle timeout	Int	60 – 120000	120
force_del_timeout	force delete timeout, disabled(0) or number > 0	Int	0 - 31	0
init_win_size	Initial window size	Int	1 – 65535	
half_close_idle_timeout	half close idle timeout, disabled(0) or number > 0	Int	60 - 15000	0

<code>reset_fwd</code>	reset forward, disabled(0) or enabled(1)	Int	0 or 1	0
<code>reset_rec</code>	reset receive, disabled(0) or enabled(1)	Int	0 or 1	0

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
`https://[AX_IP_ADDRESS]/services/rest/V2/`
- HTTP Action:
`POST`

CLI configuration before aXAPI call:

```
!
slb template tcp tcp_template_name1
    initial-window-size 100
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.template.tcp.create&name=tcp_template_name1&
idle_timeout=120&init_win_size=100&reset_fwd=0&reset_rec=0
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> Template >> L4 >> TCP

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.15.4 “`slb.template.tcp.update`” Method

This method is used to update one or more parameters in a TCP template.

Parameters

Parameter Name	Description	Data Type	Range	Default
<code>session_id</code> (*)	user authenticated session id	String		
<code>method</code> (*)	<code>slb.template.tcp.update</code>	String		
<code>name</code> (*)	tcp template name	String		
<code>idle_timeout</code>	Idle timeout	Int	60 – 120000	
<code>force_del_timeout</code>	force delete timeout, disabled(0) or number > 0	Int	0 - 31	0
<code>init_win_size</code>	Initial window size	Int	1 – 65535	
<code>half_close_idle_timeout</code>	half close idle timeout, disabled(0) or number > 0	Int	60 - 15000	0
<code>reset_fwd</code>	reset forward, disabled(0) or enabled(1)	Int	0 or 1	
<code>reset_rec</code>	reset receive, disabled(0) or enabled(1)	Int	0 or 1	

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
slb template tcp tcp_template_name1
  force-delete-timeout 8
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxxxx&method=slb.template.tcp.update&name=tcp_template_name1&idle_timeout=120&init_win_size=100&reset_fwd=0&reset_rec=0

CLI configuration before aXAPI call:

```
!
slb template tcp tcp_template_name1
  initial-window-size 100
  force-delete-timeout 8
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> Template >> L4 >> TCP

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.15.5 “slb.template.tcp.delete” Method

This method is used to delete a TCP template.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.template.tcp.delete	String		
name (*)	tcp template name	String		

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
slb template tcp tcp_template_name1
    force-delete-timeout 8
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxxxx&method=slb.template.tcp.delete&name=tcp_template_name1

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> Template >> L4 >> TCP

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.16 UDP Template

7.16.1 “slb.template.udp.getAll” Method

This method is used to get configuration information for all UDP templates configured on the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.template.udp.getAll	String		

Note: (*) parameter is required in the API.

Response Fields

udp_template_list	XML tag for the collection of udp templates
udp_template	XML tag for udp template
name	udp template name
idle_timeout	idle timeout
aging_type	none(0), immediate(1) or short-lived(2)
short_lived	short live age. only when aging_type is 2
reselect	select another server if server is down, disabled(0) or enabled(1)

Example Response

Request as:

- URL:

- [https://\[AX_IP_ADDRESS\]/services/rest/V2/](https://[AX_IP_ADDRESS]/services/rest/V2/)
- **HTTP Action:**
GET

CLI configuration before aXAPI call:

```
!
slb template udp udp_template_name 1
    idle-timeout 240
    aging short 5
    re-select-if-server-down
!
slb template udp udp_template_name 2
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.template.udp.getAll

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
  <udp_template_list>
    <udp_template>
      <name>udp_template_name1</name>
      <idle_timeout>240</idle_timeout>
      <aging_type>2</aging_type>
      <short_lived>5</short_lived>
      <reselect>1</reselect>
    </udp_template>
    <udp_template>
      <name>udp_template_name2</name>
      <idle_timeout>120</idle_timeout>
      <aging_type>0</aging_type>
      <reselect>0</reselect>
    </udp_template>
  </udp_template_list>
</response>
```

Menus Privilege

Config Mode >> Service >> Template >> L4 >> UDP

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.16.2 “slb.template.udp.search” Method

This method is used to get UDP template information by a given UDP template name.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.template.udp.search	String		
name	udp template name	String		

Note: (*) parameter is required in the API.

Response Fields

udp_template	XML tag for udp template
name	udp template name
idle_timeout	idle timeout

aging_type	none(0), immediate(1) or short-lived(2)
short_lived	short live age. only when aging_type is 2
reselect	select another server if server is down, disabled(0) or enabled(1)

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
slb template udp udp_template_name 1
  idle-timeout 240
  aging short 5
  re-select-if-server-down
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.template.udp.search&name=udp_template_name1
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
  <udp_template>
    <name>udp_template_name1</name>
    <idle_timeout>240</idle_timeout>
    <aging_type>2</aging_type>
    <short_lived>5</short_lived>
    <reselect>1</reselect>
  </udp_template>
</response>
```

Menus Privilege

Config Mode >> Service >> Template >> L4 >> UDP

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.16.3 “slb.template.udp.create” Method

This method is used to create a UDP template.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.template.udp.create	String		
name (*)	udp template name	String		
idle_timeout	Idle timeout	Int	60 – 120000	120
aging_type	none(0), immediate (1), short-lived (2)	Int	0, 1 or 2	0
short_lived	short live age. only when aging_type is 2	Int	1 - 6	
reselect	Select another server if down, disabled(0) or enable(1)	Int	0 or 1	0

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxxxx&method=slb.template.udp.create&name=udp_template_name1
&idle_timeout=120&aging_type=0&reselect=0
```

CLI configuration after aXAPI call:

```
!
slb template udp udp_template_name 1
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> Template >> L4 >> UDP

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.16.4 “slb.template.udp.update” Method

This method is used to update one or more parameters in a UDP template.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.template.udp.update	String		
name (*)	udp template name	String		
idle_timeout	Idle timeout	Int	60 – 120000	
aging_type	none(0), immediate (1), short-lived (2)	Int	0, 1 or 2	
short_lived	short live age. only when aging_type is 2	Int	1 - 6	
reselect	Select another server if down, disabled(0) or enable(1)	Int	0 or 1	

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
slb template udp udp_template_name1
    idle-timeout 240
    aging short 5
    re-select-if-server-down
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxxxx&method=slb.template.udp.update&name=udp_template_name1&idle_timeout=120&aging_type=0&reselect=0

CLI configuration after aXAPI call:

```
!
slb template udp udp_template_name 1
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> Template >> L4 >> UDP

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.16.5 “`slb.template.udp.delete`” Method

This method is used to delete a UDP template.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	<code>slb.template.udp.delete</code>	String		
name (*)	udp template name	String		

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
slb template udp udp_template_name1
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxxxx&method=slb.template.udp.delete&name=udp_template_name1

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> Template >> L4 >> UDP

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.17 Cookie Persistence Template

7.17.1 “slb.template.cookie_persistence.getAll” Method

This method is used to get configuration information for all cookie persistence template configured on the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.template.cookie_persistence.getAll	String		

Note: (*) parameter is required in the API.

Response Fields

cookie_persistence_template_list	XML tag for the collection of cookie persistence templates
cookie_persistence_template	XML tag for cookie persistence template
name	cookie persistence template name
expire	(second)
domain	
path	
match_type	port(0), server(1) or service group(2)
match_all	match scan all members, disabled(0) or enabled(0)
	only when type is server(1) or service group(2)
insert_always	disabled(0) or enabled(1)
no_honor_conn	don't honor conn rules, disabled(0) or enabled(1)

Example Response

Request as:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
- **HTTP Action:**
GET

CLI configuration before aXAPI call:

```
!
slb template persist cookie cookie_persistence_template name1
```

```
!
slb template persist cookie cookie_persistence_template_name 2
  expire 100
  dont-honor-conn-rules
  insert-always
  match-type server
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.template.cookie_persistence.getAll

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
  <cookie_persistence_template_list>
    <cookie_persistence_template>
      <name>cookie_persistence_template_name1</name>
      <expire>0</expire>
      <domain/>
      <path/>
      <match_type>0</match_type>
      <match_all>0</match_all>
      <insert_always>0</insert_always>
      <no_honor_conn>0</no_honor_conn>
    </cookie_persistence_template>
    <cookie_persistence_template>
      <name>cookie_persistence_template_name2</name>
      <expire>100</expire>
      <domain/>
      <path/>
      <match_type>1</match_type>
      <match_all>1</match_all>
      <insert_always>1</insert_always>
      <no_honor_conn>1</no_honor_conn>
    </cookie_persistence_template>
  </cookie_persistence_template_list>
</response>
```

Menus Privilege

Config Mode >> Service >> Template >> Persistent >> Cookie Persistence

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.17.2 “slb.template.cookie_persistence.search” Method

This method is used to get cookie persistence template information by given cookie persistence template name.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.template.cookie_persistence.search	String		
name (*)	cookie persistence template name	String		

Note: (*) parameter is required in the API.

Response Fields

cookie_persistence_template
name

XML tag for cookie persistence template
cookie persistence template name

expire	(second)
domain	
path	
match_type	port(0), server(1) or service group(2)
match_all	match scan all members, disabled(0) or enabled(0)
insert_always	only when type is server(1) or service group(2)
no_honor_conn	disabled(0) or enabled(1)
	don't honor conn rules, disabled(0) or enabled(1)

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
slb template persist cookie cookie_persistence_template_name1
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.template.cookie_persistence.search&name=cookie_persistence_template_name1
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
  <cookie_persistence_template>
    <name>cookie_persistence_template_name1</name>
    <expire>0</expire>
    <domain/>
    <path/>
    <match_type>0</match_type>
    <match_all>0</match_all>
    <insert_always>0</insert_always>
    <no_honor_conn>0</no_honor_conn>
  </cookie_persistence_template>
</response>
```

Menus Privilege

Config Mode >> Service >> Template >> Persistent >> Cookie Persistence
 Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.17.3 “slb.template.cookie_persistence.create” Method

This method is used to create a cookie persistence template.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.template.cookie_persistence.create	String		
name (*)	cookie persistence template name	String		
expire	disabled(0)	Int	0 – 31536000	0

domain	domain name	String		
path	path name	String		
match_type	match port(0), server (1), service group (2)	Int	0, 1 or 2	0
match_all	match scan all members, disabled(0) or enabled(1)	Int	0 or 1	0
insert_always	disabled(0) or enabled(1)	Int	0 or 1	0
no_honor_conn	don't honor conn rules	Int	0 or 1	0

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.template.cookiepersist.create&name=cookie_persistence_template_name1&expire=120
```

CLI configuration after aXAPI call:

```
!
slb template persist cookie cookie_persistence_template_name1
    expire 120
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> Template >> Persistent >> Cookie Persistence
Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.17.4 “slb.template.cookie_persistence.update” Method

This method is used to update one or more parameters in a cookie persistence template.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.template.cookie_persistence.update	String		
name (*)	cookie persistence template name	String		
expire	disabled(0)	Int	0 – 31536000	
domain	domain name	String		
path	path name	String		
match_status	disabled(0) or enabled(1)	Int	0 or 1	
match_type	match port(0) server (1)	Int	0 or 1	
match_all	match scan all members	Int	0 or 1	
insert_always	disabled(0) or enabled(1)	Int	0 or 1	

no_honor_conn	don't honor conn rules	Int	0 or 1	
----------------------	------------------------	-----	--------	--

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
slb template persist cookie cookie_persistence_template_name1
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.template.cookie_persistence.update&name=cookie_persistence_template_name1&expire=120
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> Template >> Persistent >> Cookie Persistence

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.17.5 “slb.template.cookie_persistence.delete” Method

This method is used to delete a cookie persistence template.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.template.cookie_persistence.delete	String		
name (*)	cookie persistence template name	String		

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
slb template persist cookie cookie_persistence_template_name1
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.template.cookie_persistence.delete&name=cookie_persistence_template_name1
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> Template >> Persistent >> Cookie Persistence

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.18 Source-IP Persistence Template

7.18.1 “**slb.template.src_ip_persistence.getAll**” Method

This method is used to get configuration information for all source IP persistence templates configured on the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.template.src_ip_persistence.getAll	String		

Note: (*) parameter is required in the API.

Response Fields

src_ip_persistence_template_list	XML tag for the collection of source IP persistence templates
src_ip_persistence_template	XML tag for source IP persistence template
name	source persistence template name
match_type	port(0), server(1) or service group(2)
match_all	match scan all members, disabled(0) or enabled(0)
only when type is server(1) or service group(2)	
timeout	(Minutes)
no_honor_conn	don't honor conn rules, disabled(0) or enabled(1)
incl_sport	include source port on persist, disable(0), enable(1)
netmask	
netmask6	

Example Response

Request as:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
- **HTTP Action:**
GET

CLI configuration before aXAPI call:

```
!
slb template persist source-ip src_ip_persistence_template_name1
    netmask 255.255.255.0
    match-type service-group
    timeout 200
    dont-honor-conn-rules
!
slb template persist source-ip src_ip_persistence_template_name2
    netmask 255.255.255.0
    timeout 300
    dont-honor-conn-rules
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.template.src_ip_persistence.getAll
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
  <src_ip_persistence_template_list>
    <src_ip_persistence_template>
      <name>src_ip_persistence_template_name1</name>
      <match_type>2</match_type>
      <match_all>0</match_all>
      <timeout>200</timeout>
      <no_honor_conn>1</no_honor_conn>
      <incl_sport>0</incl_sport>
      <netmask>255.255.255.0</netmask>
      <netmask6 />
    </src_ip_persistence_template>
    <src_ip_persistence_template>
      <name>src_ip_persistence_template_name2</name>
      <match_type>0</match_type>
      <match_all>0</match_all>
      <timeout>300</timeout>
      <no_honor_conn>1</no_honor_conn>
      <incl_sport>0</incl_sport>
      <netmask>255.255.0.0</netmask>
      <netmask6 />
    </src_ip_persistence_template>
  </src_ip_persistence_template_list>
</response>
```

Menus Privilege

Config Mode >> Service >> Template >> Persistent >> Source IP Persistence

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.18.2 “slb.template.src_ip_persistence.search” Method

This method is used to get source IP persistence template information by given source IP persistence template name.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.template.src_ip_persistence.search	String		
name (*)	source IP persistence template name	String		

Note: (*) parameter is required in the API.

Response Fields

src_ip_persistence_template	XML tag for source IP persistence template
name	source persistence template name
match_type	port(0), server(1) or service group(2)
match_all	match scan all members, disabled(0) or enabled(0)
	only when type is server(1) or service group(2)
timeout	(minutes)
no_honor_conn	don't honor conn rules, disabled(0) or enabled(1)
incl_sport	include source port on persist, disable(0), enable(1)
netmask	
netmask6	

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
slb template persist source-ip src_ip_persistence_template_name1
  netmask 255.255.255.0
  match-type service-group
  timeout 200
  dont-honor-conn-rules
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxxxx&method=slb.template.src_ip_persistence.search&name=src_ip_persistence_template_name1
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
  <src_ip_persistence_template>
    <name>src_ip_persistence_template_name1</name>
    <match_type>2</match_type>
    <match_all>0</match_all>
    <timeout>200</timeout>
    <no_honor_conn>1</no_honor_conn>
    <incl_sport>0</incl_sport>
    <netmask>255.255.255.0</netmask>
    <netmask6 />
  </src_ip_persistence_template>
</response>
```

Menus Privilege

Config Mode >> Service >> Template >> Persistent >> Source IP Persistence
 Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.18.3 “slb.template.src_ip_persistence.create” Method

This method is used to create a source IP persistence template.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.template.src_ip_persistence.create	String		
name (*)	Source IP persistence template name	String		
match_type	match port(0) server (1) or service group(2)	Int	0, 1 or 2	0
match_all	match scan all members, only when match_type is 1, 2	Int	0 or 1	0
timeout		Int	1 – 2000	5
no_honor_conn	don't honor conn rules	Int	0 or 1	0
incl_sport	include source port on the persist	Int	0 or 1	0
netmask		String		
netmask6		Int	1 – 128	

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
slb template persist source-ip src_ip_persistence_template_name1
  timeout 100
  dont-honor-conn-rules
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.template.src_ip_persistence.create&name=src
_ip_persistence_template_name1&timeout=100&no_honor_conn=1
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> Template >> Persistent >> Source IP Persistence
 Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.18.4 “slb.template.src_ip_persistence.update” Method

This method is used to update one or more parameters in a source IP persistence template.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.template.src_ip_persistence.update	String		
name (*)	Source IP persistence template name	String		
match_type	match port(0) server (1) or service group(2)	Int	0, 1 or 2	0
match_all	match scan all members, only when match_type is 1, 2	Int	0 or 1	0
timeout		Int	1 – 2000	5
no_honor_conn	do not honor conn rules	Int	0 or 1	0
incl_sport	include source port on the persist	Int	0 or 1	0
netmask		String		255.255.2 55.255
netmask6		Int	1 – 128	

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
slb template persist source-ip src_ip_persistence_template_name1
  netmask 255.255.255.0
  match-type service-group
  timeout 200
  dont-honor-conn-rules
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.template.src_ip_persistence.update&name=src_ip
_persistence_template_name1&timeout=100&no_honor_conn=1
```

CLI configuration after aXAPI call:

```
!
slb template persist source-ip src_ip_persistence_template_name1
  netmask 255.255.255.0
  match-type service-group
  timeout 100
  dont-honor-conn-rules
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> Template >> Persistent >> Source IP Persistence
 Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.18.5 “slb.template.src_ip_persistence.delete” Method

This method is used to delete a source IP persistence template.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.template.src_ip_persistence.delete	String		
name (*)	source IP persistence template name	String		

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
slb template persist source-ip src_ip_persistence_template_name1
  netmask 255.255.255.0
  match-type service-group
  timeout 200
  dont-honor-conn-rules
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.template.src_ip_persistence.delete&name=src_ip
_persistence_template_name1
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> Template >> Persistent >> Source IP Persistence
 Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.19 Destination-IP Template

7.19.1 “**slb.template.dst_ip_persistence.getAll**” Method

This method is used to get configuration information for all destination persistence template configured on the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.template.dst_ip_persistence.getAll	String		

Note: (*) parameter is required in the API.

Response Fields

dst_ip_persistence_template_list	XML tag for the collection of destination IP persistence templates
dst_ip_persistence_template	XML tag for destination IP persistence template
name	destination IP persistence template name
match_type	port(0), server(1) or service group(2)
match_all	match scan all members, disabled(0) or enabled(0)
timeout	only when type is server(1) or service group(2) (minutes)
no_honor_conn	don't honor conn rules, disabled(0) or enabled(1)
netmask	
netmask6	

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
slb template persist destination-ip dst_ip_persistence_template_name1
    netmask 255.255.255.0
    timeout 3
    dont-honor-conn-rules
!
slb template persist destination-ip dst_ip_persistence_template_name2
    dont-honor-conn-rules
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.template.dst_ip_persistence.getAll

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
  <dst_ip_persistence_template_list>
    <dst_ip_persistence_template>
      <name>dst_ip_persistence_template_name1</name>
      <match_type>0</match_type>
      <match_all>0</match_all>
```

```

<timeout>3</timeout>
<no_honor_conn>1</no_honor_conn>
<netmask>255.255.255.0</netmask>
<netmask6 />
</dst_ip_persistence_template>
<dst_ip_persistence_template>
<name>dst_ip_persistence_template_name2</name>
<match_type>0</match_type>
<match_all>0</match_all>
<timeout>5</timeout>
<no_honor_conn>1</no_honor_conn>
<netmask>255.255.255</netmask>
<netmask6 />
</dst_ip_persistence_template>
</dst_ip_persistence_template_list>
</response>

```

Menus Privilege

Config Mode >> Service >> Template >> Persistent >> Destination IP Persistence
Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.19.2 “slb.template.dst_ip_persistence.search” Method

This method is used to get destination IP persistence template information by given destination IP persistence template name.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.template.dst_ip_persistence.search	String		
name	destination IP persistence template name	String		

Note: (*) parameter is required in the API.

Response Fields

dst_ip_persistence_template	XML tag for destination IP persistence template
name	destination IP persistence template name
match_type	port(0), server(1) or service group(2)
match_all	match scan all members, disabled(0) or enabled(0)
timeout	only when type is server(1) or service group(2) (minutes)
no_honor_conn	don't honor conn rules, disabled(0) or enabled(1)
netmask	
netmask6	

Example Response

Request as:

- **URL:**
[https://\[AX_IP_ADDRESS\]/services/rest/V2/](https://[AX_IP_ADDRESS]/services/rest/V2/)
- **HTTP Action:**
GET

CLI configuration before aXAPI call:

```
!
slb template persist destination-ip dst_ip_persistence_template_name1
    netmask 255.255.255.0
```

```

    timeout 3
    dont-honor-conn-rules
!
slb template persist destination-ip dst_ip_persistence_template_name2
    dont-honor-conn-rules
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxxxx&method=slb.template.dst_ip_persistence.search&name=dst_ip_persistence_template_name1

Response as the HTTP body:

```

<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
    <dst_ip_persistence_template>
        <name>dst_ip_persistence_template_name1</name>
        <match_type>0</match_type>
        <match_all>0</match_all>
        <timeout>3</timeout>
        <no_honor_conn>1</no_honor_conn>
        <netmask>255.255.255.0</netmask>
        <netmask6 />
    </dst_ip_persistence_template>
</response>
```

Menus Privilege

Config Mode >> Service >> Template >> Persistent >> Destination IP Persistence
Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.19.3 “slb.template.dst_ip_persistence.create” Method

This method is used to create a destination IP persistence template.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.template.dst_ip_persistence.create	String		
name (*)	dest IP persistence template name	String		
match_type	match port(0) server (1) or service group(2)	Int	0, 1 or 2	0
match_all	match scan all members, only when match_type is 1, 2	Int	0 or 1	0
timeout	timeout	Int	1 – 2000	5
no_honor_conn	do not honor conn rules	Int	0 or 1	0
netmask		String		255.255.255.0
netmask6		Int	1 – 128	55.255

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxxxx&method=slb.template.dst_ip_persistence.create&name=dst_ip_persistence_template_name1&timeout=100&no_honor_conn=1
```

CLI configuration after aXAPI call:

```
!
slb template persist destination-ip dst_ip_persistence_template_name1
    timeout 100
    dont-honor-conn-rules
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> Template >> Persistent >> Destination IP Persistence
Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.19.4 “slb.template.dst_ip_persistence.update” Method

This method is used to update one or more parameters in a destination IP persistence template.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.template.dst_ip_persistence.update	String		
name (*)	dest IP persistence template name	String		
match_type	match port(0) server (1) or service group(2)	Int	0, 1 or 2	0
match_all	match scan all members, only when match_type is 1, 2	Int	0 or 1	0
timeout	timeout	Int	1 – 2000	5
no_honor_conn	don't honor conn rules	Int	0 or 1	0
netmask		String		
netmask6		Int	1 – 128	

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
slb template persist destination-ip dst_ip_persistence_template_name1
    netmask 255.255.255.0
    timeout 3
```

```

dont-honor-conn-rules
!
```

HTTP Body

```

session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.template.dst_ip_persistence.update&name=dst_ip
_persistence_template_name1&timeout=100&no_honor_conn=1
```

CLI configuration after aXAPI call:

```

!
slb template persist destination-ip dst_ip_persistence_template_name1
    netmask 255.255.255.0
    timeout 100
    dont-honor-conn-rules
!
```

Response as the HTTP body:

```

<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> Template >> Persistent >> Destination IP Persistence
Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.19.5 “slb.template.dst_ip_persistence.delete” Method

This method is used to delete a destination IP persistence template.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.template.dest_ip_persistence.delete	String		
name (*)	dest IP persistence template name	String		

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```

!
slb template persist destination-ip dst_ip_persistence_template_name1
    netmask 255.255.255.0
    timeout 3
    dont-honor-conn-rules
```

```
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxxxx&method=slb.template.dst_ip_persistence.delete&name=dst_ip_persistence_template_name1
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> Template >> Persistent >> Destination IP Persistence
Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.20 SSL Persistence Template

7.20.1 “**slb.template.ssl_sid_persistence.getAll**” Method

This method is used to get configuration information for all SSL session ID persistence templates configured on the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.template.ssl_sid_persistence.getAll	String		

Note: (*) parameter is required in the API.

Response Fields

ssl_sid_persistence_template_list	XML tag for the collection of SSL session id persistence templates
ssl_sid_persistence_template	XML tag for SSL session id persistence template
name	SSL session id persistence template name
timeout	(minutes)
no_honor_conn	don't honor conn rules, disabled(0) or enabled(1)

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
slb template persist ssl-sid ssl_sid_persistence_template_name1
!
```

```

slb template persist ssl-sid ssl_sid_persistence_template_name2
    timeout 10
    dont-honor-conn-rules
!

```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxxxx&method=slb.template.ssl_sid_persistence.getAll

Response as the HTTP body:

```

<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
  <ssl_sid_persistence_template_list>
    <ssl_sid_persistence_template>
      <name>ssl_sid_persistence_template_name1</name>
      <timeout>5</timeout>
      <no_honor_conn>0</no_honor_conn>
    </ssl_sid_persistence_template>
    <ssl_sid_persistence_template>
      <name>ssl_sid_persistence_template_name2</name>
      <timeout>10</timeout>
      <no_honor_conn>1</no_honor_conn>
    </ssl_sid_persistence_template>
  </ssl_sid_persistence_template_list>
</response>

```

Menus Privilege

Config Mode >> Service >> Template >> Persistent >> SSL Session ID Persistence
Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.20.2 “`slb.template.ssl_sid_persistence.search`” Method

This method is used to get SSL session ID persistence template information by given SSL session ID persistence template name.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	<code>slb.template.ssl_sid_persistence.search</code>	String		
name (*)	SSL session id persistence template name	String		

Note: (*) parameter is required in the API.

Response Fields

ssl_sid_persistence_template	XML tag for SSL session id persistence template
name	SSL session id persistence template name
timeout	(minutes)
no_honor_conn	don't honor conn rules, disabled(0) or enabled(1)

Example Response

Request as:

- URL:
`https://[AX_IP_ADDRESS]/services/rest/V2/`
- HTTP Action:
`GET`

CLI configuration before aXAPI call:

```
!
slb template persist ssl-sid ssl_sid_persistence_template_name1
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.template.ssl_sid_persistence.search&name=ssl_sid_persistence_template_name1
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
    <ssl_sid_persistence_template>
        <name>ssl_sid_persistence_template_name1</name>
        <timeout>5</timeout>
        <no_honor_conn>0</no_honor_conn>
    </ssl_sid_persistence_template>
</response>
```

Menus Privilege

Config Mode >> Service >> Template >> Persistent >> SSL Session ID Persistence
Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.20.3 “`slb.template.ssl_sid_persistence.create`” Method

This method is used to create an SSL session ID persistence template.

Parameters

Parameter Name	Description	Data Type	Range	Default
<code>session_id (*)</code>	user authenticated session id.	String		
<code>method (*)</code>	<code>slb.template.ssl_sid_persistence.create</code>	String		
<code>name (*)</code>	SSL session id persistence template name	String		
<code>timeout</code>	timeout	Int	1 – 2000	5
<code>no_honor_conn</code>	don't honor conn rules	Int	0 or 1	0

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
`https://[AX_IP_ADDRESS]/services/rest/V2/`
- HTTP Action:
POST

CLI configuration after aXAPI call:

```
!
slb template persist ssl-sid ssl_sid_persistence_template_name1
    timeout 100
    dont-honor-conn-rules
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.template.ssl_sid_persistence.create&name=ssl_sid_persistence_template_name1&timeout=100&no_honor_conn=1
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> Template >> Persistent >> SSL Session ID Persistence
Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.20.4 “slb.template.ssl_sid_persistence.update” Method

This method is used to update one or more parameters in an SSL session ID persistence template.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.template.ssl_sid_persistence.update	String		
name (*)	SSL session id persistence template name	String		
timeout	timeout	Int	1 – 2000	5
no_honor_conn	don't honor conn rules	Int	0 or 1	0

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
- **HTTP Action:**
POST

CLI configuration before aXAPI call:

```
!
slb template persist ssl-sid ssl_sid_persistence_template_name1
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.template.ssl_sid_persistence.update&name=ssl_sid_persistence_template_name1&timeout=100&no_honor_conn=1

CLI configuration after aXAPI call:

```
!
slb template persist ssl-sid ssl_sid_persistence_template_name1
    timeout 100
    dont-honor-conn-rules
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
```

```
</response>
```

Menus Privilege

Config Mode >> Service >> Template >> Persistent >> SSL Session ID Persistence
Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.20.5 “slb.template.ssl_sid_persistence.delete” Method

This method is used to delete an SSL session ID persistence template.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.template.ssl_sid_persistence.delete	String		
name (*)	SSL session id persistence template name	String		

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
[https://\[AX_IP_ADDRESS\]/services/rest/V2/](https://[AX_IP_ADDRESS]/services/rest/V2/)
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
slb template persist ssl-sid ssl_sid_persistence_template_name1
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.template.ssl_sid_persistence.delete&name=ssl_sid_persistence_template_name1
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> Template >> Persistent >> SSL Session ID Persistence
Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.21 Client-SSL Template

7.21.1 “slb.template.client_ssl.getAll” Method

This method is used to get configuration information for all client SSL templates configured on the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	slb.template.client_ssl.getAll	String		

Note: (*) parameter is required in the API.

Response Fields

client_ssl_template_list	XML tag for the collection of client SSL templates
client_ssl_template	XML tag for client SSL template
name	client SSL template name
cert_name	certificate name
chain_name	chain certificate name, only when cert_name not none.
key_name	key name
phrase	only when key_name not none
cache_size	
ssl_false_stat	disabled(0) or enabled(1)
client_check_mode	require(0), request(1) or ignore(2)
client_close_notify	disabled(0) or enabled(1)
client_ca_cert_name	client CA certificate name
client_cert_rev_list	client cert-revocation list name
ca_cert_list	XML tag for the collection of CA certificate
ca_cert	CA certificate
cipher_list	XML tag for the collection of cipher_list
cipher	 SSL3_RSA_RC4_40_MD5 SSL3_RSA_RC4_128_MD5 SSL3_RSA_RC4_128_SHA SSL3_RSA_DES_40_CBC_SHA SSL3_RSA_DES_64_CBC_SHA SSL3_RSA_DES_192_CBC3_SHA TLS1_RSA_AES_128_SHA TLS1_RSA_AES_256_SHA TLS1_RSA_EXPORT1024_RC4_56_MD5 TLS1_RSA_EXPORT1024_RC4_56_SHA

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
slb template client-ssl client_ssl_template_name1
  cert cert2
  chain-cert cert2
  key cert2 pass-phrase encrypted
    eGzct2y1ODU8EIy41dsA5zwQjLjV2wDnPBCMuNXbAOc8EIy41dsA5zwQjLjV2wDn
```

```

cipher TLS1_RSA_EXPORT1024_RC4_56_MD5
cipher TLS1_RSA_EXPORT1024_RC4_56_SHA
cipher SSL3_RSA_RC4_128_MD5
cipher SSL3_RSA_RC4_128_SHA
cipher SSL3_RSA_DES_40_CBC_SHA
cipher SSL3_RSA_DES_64_CBC_SHA
cipher SSL3_RSA DES_192_CBC3_SHA
cipher TLS1_RSA_AES_128_SHA
cipher TLS1_RSA_AES_256_SHA
client-certificate Request
ssl-false-start-disable
ca-cert ca_cert_name1
ca-cert ca_cert_name2
crl crl1
!
slb template client-ssl cl2
cipher TLS1_RSA_EXPORT1024_RC4_56_MD5
cipher TLS1_RSA_EXPORT1024_RC4_56_SHA
ca-cert ca_cert_name1
ca-cert ca_cert_name2
!

```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxxxx&method=slb.template.client_ssl.getAll

Response as the HTTP body:

```

<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
  <client_ssl_template_list>
    <client_ssl_template>
      <name>client_ssl_template_name1</name>
      <cert_name>cert2</cert_name>
      <chain_name>cert2</chain_name>
      <key_name>cert2</key_name>
      <phrase>123</phrase>
      <cache_size>0</cache_size>
      <ssl_false_stat>0</ssl_false_stat>
      <client_check_mode>1</client_check_mode>
      <client_close_notify>0</client_close_notify>
      <client_ca_cert_name>ca_cert_name1</client_ca_cert_name>
      <client_cert_rev_list>crl1</client_cert_rev_list>
      <ca_cert_list>
        <ca_cert>ca_cert_name1</ca_cert>
        <ca_cert>ca_cert_name2</ca_cert>
      </ca_cert_list>
      <cipher_list>
        <cipher>TLS1_RSA_EXPORT1024_RC4_56_MD5</cipher>
        <cipher>TLS1_RSA_EXPORT1024_RC4_56_SHA</cipher>
        <cipher>SSL3_RSA_RC4_128_MD5</cipher>
        <cipher>SSL3_RSA_RC4_128_SHA</cipher>
        <cipher>SSL3_RSA_DES_40_CBC_SHA</cipher>
        <cipher>SSL3_RSA_DES_64_CBC_SHA</cipher>
        <cipher>SSL3_RSA DES_192_CBC3_SHA</cipher>
        <cipher>TLS1_RSA_AES_128_SHA</cipher>
        <cipher>TLS1_RSA_AES_256_SHA</cipher>
      </cipher_list>
    </client_ssl_template>
    <client_ssl_template>
      <name>client_ssl_template_name2</name>
      <cert_name></cert_name>
      <chain_name></chain_name>
      <key_name></key_name>
      <cache_size>0</cache_size>
      <ssl_false_stat>0</ssl_false_stat>
      <client_check_mode>2</client_check_mode>
      <client_close_notify>0</client_close_notify>
      <client_ca_cert_name>ca_cert_name1</client_ca_cert_name>
      <client_cert_rev_list></client_cert_rev_list>
    </client_ssl_template>
  </client_ssl_template_list>
</response>

```

```

<ca_cert_list>
    <ca_cert>ca_cert_name1</ca_cert>
    <ca_cert>ca_cert_name2</ca_cert>
</ca_cert_list>
<cipher_list>
    <cipher>TLS1_RSA_EXPORT1024_RC4_56_MD5</cipher>
    <cipher>TLS1_RSA_EXPORT1024_RC4_56_SHA</cipher>
</cipher_list>
</client_ssl_template>
</client_ssl_template_list>
</response>

```

Menus Privilege

Config Mode >> Service >> Template >> SSL >> Client SSL

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.21.2 “slb.template.client_ssl.search” Method

This method is used to get client SSL template information by given client SSL template name.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.template.client_ssl.search	String		
name (*)	client SSL template name	String		

Note: (*) parameter is required in the API.

Response Fields

client_ssl_template	XML tag for client SSL template
name	client SSL template name
cert_name	certificate name
chain_name	chain certificate name, only when cert_name not none.
key_name	key name
phrase	only when key_name not none
cache_size	
ssl_false_stat	disabled(0) or enabled(1)
client_check_mode	require(0), request(1) or ignore(2)
client_close_notify	disabled(0) or enabled(1)
client_ca_cert_name	client CA certificate name
client_cert_rev_list	client cert-revocation list name
ca_cert_list	XML tag for the collection of CA certificate
ca_cert	CA certificate
cipher_list	XML tag for the collection of cipher_list
cipher	SSL3_RSA_RC4_40_MD5 SSL3_RSA_RC4_128_MD5 SSL3_RSA_RC4_128_SHA SSL3_RSA_DES_40_CBC_SHA SSL3_RSA_DES_64_CBC_SHA SSL3_RSA_DES_192_CBC3_SHA TLS1_RSA_AES_128_SHA TLS1_RSA_AES_256_SHA TLS1_RSA_EXPORT1024_RC4_56_MD5 TLS1_RSA_EXPORT1024_RC4_56_SHA

Example Response

Request as:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
- **HTTP Action:**
GET

CLI configuration before aXAPI call:

```
!slb template client-ssl client_ssl_template_name1
cert cert2
chain-cert cert2
key cert2 pass-phrase encrypted
eGzct2ylODU8EIy41dsA5zwQjLjV2wDnPBCMuNXbAOc8EIy41dsA5zwQjLjV2wDn
cipher TLS1_RSA_EXPORT1024_RC4_56_MD5
cipher TLS1_RSA_EXPORT1024_RC4_56_SHA
cipher SSL3_RSA_RC4_128_MD5
cipher SSL3_RSA_RC4_128_SHA
cipher SSL3_RSA_DES_40_CBC_SHA
cipher SSL3_RSA_DES_64_CBC_SHA
cipher SSL3_RSA_DES_192_CBC3_SHA
cipher TLS1_RSA_AES_128_SHA
cipher TLS1_RSA_AES_256_SHA
client-certificate Request
ssl-false-start-disable
ca-cert ca_cert_name1
ca-cert ca_cert_name2
crl crl1
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.template.client_ssl.search&name=client_ssl_template_name1
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
  <client_ssl_template>
    <name>client_ssl_template_name1</name>
    <cert_name>cert2</cert_name>
    <chain_name>cert2</chain_name>
    <key_name>cert2</key_name>
    <phrase>123</phrase>
    <cache_size>0</cache_size>
    <ssl_false_stat>0</ssl_false_stat>
    <client_check_mode>1</client_check_mode>
    <client_close_notify>0</client_close_notify>
    <client_ca_cert_name>cert2</client_ca_cert_name>
    <client_cert_rev_list>crl1</client_cert_rev_list>
    <ca_cert_list>
      <ca_cert>ca_cert_name1</ca_cert>
      <ca_cert>ca_cert_name2</ca_cert>
    </ca_cert_list>
    <cipher_list>
      <cipher>TLS1_RSA_EXPORT1024_RC4_56_MD5</cipher>
      <cipher>TLS1_RSA_EXPORT1024_RC4_56_SHA</cipher>
      <cipher>SSL3_RSA_RC4_128_MD5</cipher>
      <cipher>SSL3_RSA_RC4_128_SHA</cipher>
      <cipher>SSL3_RSA_DES_40_CBC_SHA</cipher>
      <cipher>SSL3_RSA_DES_64_CBC_SHA</cipher>
      <cipher>SSL3_RSA_DES_192_CBC3_SHA</cipher>
      <cipher>TLS1_RSA_AES_128_SHA</cipher>
      <cipher>TLS1_RSA_AES_256_SHA</cipher>
    </cipher_list>
  </client_ssl_template>
</response>
```

Menus Privilege

Config Mode >> Service >> Template >> SSL >> Client SSL

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.21.3 “slb.template.client_ssl.create” Method

This method is used to create an SSL client persistence template.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.template.client_ssl.create	String		
name (*)	client SSL template name	String		
cert_name	certificate name	String		
chain_name	chain certificate name	String		
key_name	key name	String		
cache_size	cache size	Int	0 – 131072	0
ssl_false_stat	SSL false stat, disabled(0) or enabled(1)	Int	0 or 1	1
phrase	phrase	String		
client_check_mode	require(0), request(1) or ignore(2)	Int	0, 1 or 2	0
client_close_notify	client close notify, disabled(0) or enabled(1)	Int	0 or 1	0
client_ca_cert_name	CA certificate name	String		
client_cert_rev_list	cert-revocation list	String		
ca_cert_list (*)	CA certificate list to be added ca_cert1^B ...ca_certN ^B: ASCII Code 0x02, URL-encode %02	String		
ca_cert<n> name (*)	CA certificate at element <n> ca certificate name	String		
cipher_list (*)	cipher list to be added cipher1^B ...cipherN ^B: ASCII Code 0x02, URL-encode %02	String		
cipher<n> name (*)	cipher at element <n> cipher name SSL3_RSA_RC4_40_MD5 SSL3_RSA_RC4_128_MD5 SSL3_RSA_RC4_128_SHA SSL3_RSA_DES_40_CBC_SHA SSL3_RSA_DES_64_CBC_SHA SSL3_RSA_DES_192_CBC3_SHA TLS1_RSA_AES_128_SHA TLS1_RSA_AES_256_SHA TLS1_RSA_EXPORT1024_RC4_56_MD5 TLS1_RSA_EXPORT1024_RC4_56_SHA	String		

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
[https://\[AX_IP_ADDRESS\]/services/rest/V2/](https://[AX_IP_ADDRESS]/services/rest/V2/)
- HTTP Action:
POST

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.template.client_ssl.create&name=client_ssl_
template_name1&cert_name=ca1&chain_name=chain1&key_name=key1&cache_size=10000&cipher_1
```

```
ist=cipher1%02cipher2&cipher1=name%03TLS1_RSA_EXPORT1024_RC4_56_MD5&cipher2=name%03TLS1_RSA_EXPORT1024_RC4_56_SHA
```

CLI configuration after aXAPI call:

```
!
slb template client-ssl client_ssl_template_name1
  cert ca1
  chain-cert chain1
  key key1
  session-cache-size 10000
  cipher TLS1_RSA_EXPORT1024_RC4_56_MD5
  cipher TLS1_RSA_EXPORT1024_RC4_56_SHA
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> Template >> SSL >> Client SSL

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.21.4 “slb.template.client_ssl.update” Method

This method is used to update one or more parameters in a client SSL template.

Note: All cipher_list parameters must be included in the update request.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.template.client_ssl.update	String		
name (*)	client SSL template name	String		
cert_name	certificate name	String		
chain_name	chain certificate name	String		
key_name	key name	String		
cache_size	cache size	Int	0 – 131072	0
ssl_false_stat	SSL false stat, disabled(0) or enabled(1)	Int	0 or 1	1
phrase	phrase	String		
client_check_mode	require(0), request(1) or ignore(2)	Int	0, 1 or 2	0
client_close_notify	client close notify, disabled(0) or enabled(1)	Int	0 or 1	0
client_ca_cert_name	CA certificate name	String		
client_cert_rev_list	cert-revocation list	String		
ca_cert_list (*)	CA certificate list to be added ca_cert1^B ...ca_certN ^B: ASCII Code 0x02, URL-encode %02	String		
ca_cert<n> name (*)	CA certificate at element <n> ca certificate name	String		
cipher_list (*)	cipher list to be added cipher1^B ...cipherN ^B: ASCII Code 0x02, URL-encode %02	String		
cipher<n> name (*)	cipher at element <n> cipher name SSL3_RSA_RC4_40_MD5 SSL3_RSA_RC4_128_MD5	String		

	SSL3_RSA_RC4_128_SHA SSL3_RSA DES 40_CBC SHA SSL3_RSA DES 64_CBC SHA SSL3_RSA DES 192_CBC3 SHA TLS1_RSA_AES 128_SHA TLS1 RSA AES 256 SHA TLS1_RSA_EXPORT1024_RC4_56_MD5 TLS1 RSA EXPORT1024 RC4 56 SHA		
--	---	--	--

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
slb template client-ssl client_ssl_template_name1
    cipher TLS1_RSA_EXPORT1024_RC4_56_MD5
    cipher TLS1_RSA_EXPORT1024_RC4_56_SHA
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxxx&method=slb.template.client_ssl.update&name=client_ssl_template_name1&cert_name=ca1&chain_name=chain1&key_name=key1&cache_size=10000&cipher_list=cipher1%02cipher2&cipher1=name%03TLS1_RSA_EXPORT1024_RC4_56_MD5&cipher2=name%03TLS1_RSA_EXPORT1024_RC4_56_SHA
```

CLI configuration after aXAPI call:

```
!
slb template client-ssl client_ssl_template_name1
    cert ca1
    chain-cert chain1
    key key1
    session-cache-size 10000
    cipher TLS1_RSA_EXPORT1024_RC4_56_MD5
    cipher TLS1_RSA_EXPORT1024_RC4_56_SHA
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> Template >> SSL >> Client SSL

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.21.5 “slb.template.client_ssl.delete” Method

This method is used to delete a client SSL template.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.template.client_ssl.delete	String		
name (*)	client SSL template name	String		

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
slb template client-ssl client_ssl_template_name1
  cipher TLS1_RSA_EXPORT1024_RC4_56_MD5
  cipher TLS1_RSA_EXPORT1024_RC4_56_SHA
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.template.client_ssl.delete&name=client_ssl_template_name1
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> Template >> SSL >> Client SSL

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.22 Server-SSL Template

7.22.1 “slb.template.server_ssl.getAll” Method

This method is used to get configuration information for all server SSL templates configured on the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		

method (*)	slb.template.server_ssl.getAll	String		
-------------------	---------------------------------------	--------	--	--

Note: (*) parameter is required in the API.

Response Fields

server_ssl_template_list	XML tag for the collection of server SSL templates
server_ssl_template	XML tag for server SSL template
name	server SSL template name
cert_name	certificate name
key_name	key name
pass_phrase	pass phrase
ca_cert_name	CA certificate name
tls_ssl_ver	default (0), TLS1.0 (1), SSL3.0 (2)
close_notify	close notification, disabled(0) or enabled(1)
cipher_list	XML tag for the collection of cipher_list
cipher	<code>SSL3_RSA_RC4_40_MD5</code> <code>SSL3_RSA_RC4_128_MD5</code> <code>SSL3_RSA_RC4_128_SHA</code> <code>SSL3_RSA_DES_40_CBC_SHA</code> <code>SSL3_RSA_DES_64_CBC_SHA</code> <code>SSL3_RSA_DES_192_CBC3_SHA</code> <code>TLS1_RSA_AES_128_SHA</code> <code>TLS1_RSA_AES_256_SHA</code> <code>TLS1_RSA_EXPORT1024_RC4_56_MD5</code> <code>TLS1_RSA_EXPORT1024_RC4_56_SHA</code>

Example Response

Request as:

- URL:
`https://[AX_IP_ADDRESS]/services/rest/V2/`
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
slb template server-ssl server_ssl_template_name1
  cert cert_name1
  key key_name1
  version 30
  cipher TLS1_RSA_EXPORT1024_RC4_56_MD5
  cipher SSL3_RSA_RC4_40_MD5
!
slb template server-ssl cs2
  close-notify
  cert cert_name2
  key key_name2
  version 31
  cipher TLS1_RSA_EXPORT1024_RC4_56_SHA
  cipher TLS1_RSA_AES_128_SHA
!
```

HTTP Body

`session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.template.server_ssl.getAll`

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
  <server_ssl_template_list>
    <server_ssl_template>
      <name>server_ssl_template_name1</name>
      <cert_name>cert_name1</cert_name>
```

```

<key_name>key_name1</key_name>
<ca_cert_name></ca_cert_name>
<pass_phrase></pass_phrase>
<tls_ssl_ver>2</tls_ssl_ver>
<close_notify>0</close_notify>
<cipher_list>
    <cipher>TLS1_RSA_EXPORT1024_RC4_56_MD5</cipher>
    <cipher>SSL3_RSA_RC4_40_MD5</cipher>
</cipher_list>
</server_ssl_template>
<server_ssl_template>
    <name>server_ssl_template_name2</name>
    <cert_name>cert_name2</cert_name>
    <key_name>key_name2</key_name>
    <ca_cert_name></ca_cert_name>
    <pass_phrase></pass_phrase>
    <tls_ssl_ver>1</tls_ssl_ver>
    <close_notify>1</close_notify>
    <cipher_list>
        <cipher>TLS1_RSA_EXPORT1024_RC4_56_SHA</cipher>
        <cipher>"TLS1_RSA_AES_128_SHA</cipher>
    </cipher_list>
</server_ssl_template>
</server_ssl_template_list>
</response>

```

Menus Privilege

Config Mode >> Service >> Template >> SSL >> Server SSL

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.22.2 “slb.template.server_ssl.search” Method

This method is used to get server SSL template information by given server SSL template name.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.template.server_ssl.search	String		
name (*)	server SSL template name	String		

Note: (*) parameter is required in the API.

Response Fields

server_ssl_template	XML tag for server SSL template
name	server SSL template name
cert_name	certificate name
key_name	key name
pass_phrase	pass phrase
ca_cert_name	CA certificate name
tls_ssl_ver	default (0), TLS1.0 (1), SSL3.0 (2)
close_notify	close notification, disabled(0) or enabled(1)
cipher_list	XML tag for the collection of cipher_list
cipher	<code>SSL3_RSA_RC4_40_MD5</code> <code>SSL3_RSA_RC4_128_MD5</code> <code>SSL3_RSA_RC4_128_SHA</code> <code>SSL3_RSA_DES_40_CBC_SHA</code>

```

SSL3_RSA_DES_64_CBC_SHA
SSL3_RSA_DES_192_CBC3_SHA
TLS1_RSA_AES_128_SHA
TLS1_RSA_AES_256_SHA
TLS1_RSA_EXPORT1024_RC4_56_MD5
TLS1_RSA_EXPORT1024_RC4_56_SHA

```

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
slb template server-ssl server_ssl_template_name1
  cert cert_name1
  key key_name1
  version 30
  cipher TLS1_RSA_EXPORT1024_RC4_56_MD5
  cipher SSL3_RSA_RC4_40_MD5
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.template.server_ssl.search&name=server_ssl_template_name1
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
  <server_ssl_template>
    <name>server_ssl_template_name1</name>
    <cert_name>cert_name1</cert_name>
    <key_name>key_name1</key_name>
    <ca_cert_name></ca_cert_name>
    <pass_phrase></pass_phrase>
    <tls_ssl_ver>2</tls_ssl_ver>
    <close_notify>0</close_notify>
    <cipher_list>
      <cipher>TLS1_RSA_EXPORT1024_RC4_56_MD5</cipher>
      <cipher>SSL3_RSA_RC4_40_MD5</cipher>
    </cipher_list>
  </server_ssl_template>
</response>
```

Menus Privilege

Config Mode >> Service >> Template >> SSL >> Server SSL

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.22.3 “slb.template.server_ssl.create” Method

This method is used to create an SSL server persistence template.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		

method (*)	slb.template.server_ssl.create	String		
name (*)	server SSL template name	String		
cert_name	certificate name	String		
key_name	key name	String		
pass_phrase	pass phrase	String		
ca_cert_name	CA certificate name	String		
tls_ssl_ver	default (0), TLS1.0 (1) or SSL3.0 (2)	Int	0, 1 or 2	0
close_notify	disabled (0) or enabled (1)	Int	0 or 1	0
cipher_list (*)	cipher list to be added cipher1^B ...cipherN ^B: ASCII Code 0x02, URL-encode %02	String		
cipher<n> name (*)	cipher at element <n> cipher name SSL3_RSA_RC4_40_MD5 SSL3_RSA_RC4_128_MD5 SSL3_RSA_RC4_128_SHA SSL3_RSA_DES_40_CBC_SHA SSL3_RSA_DES_64_CBC_SHA SSL3_RSA_DES_192_CBC3_SHA TLS1_RSA_AES_128_SHA TLS1_RSA_AES_256_SHA TLS1_RSA_EXPORT1024_RC4_56_MD5 TLS1_RSA_EXPORT1024_RC4_56_SHA	String		

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.template.server_ssl.create&name=server_ssl_
template_name1&ca_cert_name=ca1&cipher_list=cipher1%02cipher2&cipher1=name%03TLS1_RSA_
_EXPORT1024_RC4_56_MD5&cipher2=name%03TLS1_RSA_EXPORT1024_RC4_56_SHA
```

CLI configuration after aXAPI call:

```
!
slb template server-ssl server_ssl_template_name1
  ca-cert ca-1.pem
  cipher TLS1_RSA_EXPORT1024_RC4_56_MD5
  cipher TLS1_RSA_EXPORT1024_RC4_56_SHA
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> Template >> SSL >> Server SSL

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.22.4 “slb.template.server_ssl.update” Method

The aXAPI will note update a cipher list if there is no cipher_list parameter in the request. To update a cipher list, you must send all cipher_list in cipher_list parameter. This is different with update a server port before.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.template.server_ssl.update	String		
name (*)	server SSL template name	String		
cert_name	certificate name	String		
key_name	key name	String		
pass_phrase	pass phrase	String		
ca_cert_name	CA certificate name	String		
tls_ssl_ver	default (0), TLS1.0 (1) or SSL3.0 (2)	Int	0, 1 or 2	0
close_notify	disabled (0) or enabled (1)	Int	0 or 1	0
cipher_list (*)	cipher list to be added cipher1^B ..cipherN ^B: ASCII Code 0x02, URL-encode %02	String		
cipher<n> name (*)	cipher at element <n> cipher name SSL3_RSA_RC4_40_MD5 SSL3_RSA_RC4_128_MD5 SSL3_RSA_RC4_128_SHA SSL3_RSA_DES_40_CBC_SHA SSL3_RSA_DES_64_CBC_SHA SSL3_RSA_DES_192_CBC3_SHA TLS1_RSA_AES_128_SHA TLS1_RSA_AES_256_SHA TLS1_RSA_EXPORT1024_RC4_56_MD5 TLS1_RSA_EXPORT1024_RC4_56_SHA	String		

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
slb template server-ssl server_ssl_template_name1
  cipher TLS1_RSA_EXPORT1024_RC4_56_SHA
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.template.server_ssl.update&name=server_ssl_temp_name1&ca_cert_name=ca1&cipher_list=cipher1%02cipher2&cipher1=name%03TLS1_RSA_EXPORT1024_RC4_56_MD5&cipher2=name%03TLS1_RSA_EXPORT1024_RC4_56_SHA
```

CLI configuration after aXAPI call:

```
!
slb template server-ssl server_ssl_template_name1
  ca-cert ca-1.pem
  cipher TLS1 RSA EXPORT1024 RC4 56 MD5
```

```
! cipher TLS1_RSA_EXPORT1024_RC4_56_SHA
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> Template >> SSL >> Server SSL

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.22.5 “slb.template.server_ssl.delete” Method

This method is used to delete a server SSL template.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.template.server_ssl.delete	String		
name (*)	server SSL template name	String		

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
slb template server-ssl server_ssl_template_name1
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.template.server_ssl.delete&name=server_ssl_template_name1
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> Template >> SSL >> Server SSL

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.23 TCP-Proxy Template

7.23.1 “slb.template.tcp_proxy.getAll” Method

This method is used to get configuration information for all TCP proxy templates configured on the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.template.tcp_proxy.getAll	String		

Note: (*) parameter is required in the API.

Response Fields

tcp_proxy_template_list	XML tag for the collection of TCP proxy templates
tcp_proxy_template	XML tag for TCP proxy template
name	TCP proxy template name
fin_timeout	(seconds)
idle_timeout	(seconds)
force_del_timeout	force delete timeout (seconds)
half_close_idle_timeout	(seconds)
retransmit_retries	retransmit retries
syn_retries	SYN retries
time_wait	(seconds)
receive_buff	receive buffer, (bytes)
transmit_buff	transmit buffer, (bytes)
init_window_size initial	window size, disabled(0)
nagle	disabled(0) or enabled(1)
back_win_sca	backend window scaling
mss	Responding MSS to use if client MSS is large, disabled(0)

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
slb template tcp-proxy tcp_proxy_template_name1
!
slb template tcp-proxy tcp_proxy_template_name2
    idle-timeout 60
    fin-timeout 50
    syn-retries 10
    retransmit-retries 5
    timewait 10
    nagle
```

```

initial-window-size 100
!

```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.template.tcp_proxy.getAll
```

Response as the HTTP body:

```

<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
  <tcp_proxy_template_list>
    <tcp_proxy_template>
      <name>tcp_proxy_template_name1</name>
      <fin_timeout>5</fin_timeout>
      <idle_timeout>600</idle_timeout>
      <force_del_timeout>0</force_del_timeout>
      <half_close_idle_timeout>0</half_close_idle_timeout>
      <retransmit_retries>3</retransmit_retries>
      <syn_retries>5</syn_retries>
      <time_wait>5</time_wait>
      <receive_buff>87380</receive_buff>
      <transmit_buff>16384</transmit_buff>
      <init_win_size>0</init_win_size>
      <nagle>0</nagle>
      <back_win_sca>0</back_win_sca>
      <mss>0</mss>
    </tcp_proxy_template>
    <tcp_proxy_template>
      <name>tcp_proxy_template_name2</name>
      <fin_timeout>50</fin_timeout>
      <idle_timeout>60</idle_timeout>
      <force_del_timeout>0</force_del_timeout>
      <half_close_idle_timeout>0</half_close_idle_timeout>
      <retransmit_retries>5</retransmit_retries>
      <syn_retries>10</syn_retries>
      <time_wait>10</time_wait>
      <receive_buff>87380</receive_buff>
      <transmit_buff>16384</transmit_buff>
      <init_win_size>100</init_win_size>
      <nagle>1</nagle>
      <back_win_sca>0</back_win_sca>
      <mss>0</mss>
    </tcp_proxy_template>
  </tcp_proxy_template_list>
</response>

```

Menus Privilege

Config Mode >> Service >> Template >> TCP Proxy

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.23.2 “slb.template.tcp_proxy.search” Method

This method is used to get TCP proxy template information by given TCP proxy template name.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.template.tcp_proxy.search	String		
name (*)	tcp proxy template name	String		

Note: (*) parameter is required in the API.

Response Fields

tcp_proxy_template	XML tag for TCP proxy template
name	TCP proxy template name
fin_timeout	(seconds)
idle_timeout	(seconds)
force_del_timeout	force delete timeout (seconds)
half_close_idle_timeout	(seconds)
retransmit_retries	retransmit retries
syn_retries	SYN retries
time_wait	(seconds)
receive_buff	receive buffer, (bytes)
transmit_buff	transmit buffer, (bytes)
init_window_size	initial window size, disabled(0)
nagle	disabled(0) or enabled(1)
back_win_sca	backend window scaling
mss	Responding MSS to use if client MSS is large, disabled(0)

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
slb template tcp-proxy tcp_proxy_template_name1
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.template.tcp_proxy.search&name=tcp_proxy_template_name1
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
  <tcp_proxy_template>
    <name>tcp_proxy_template_name1</name>
    <fin_timeout>5</fin_timeout>
    <idle_timeout>600</idle_timeout>
    <force_del_timeout>0</force_del_timeout>
    <half_close_idle_timeout>0</half_close_idle_timeout>
    <retransmit_retries>3</retransmit_retries>
    <syn_retries>5</syn_retries>
    <time_wait>5</time_wait>
    <receive_buff>87380</receive_buff>
    <transmit_buff>16384</transmit_buff>
    <init_win_size>0</init_win_size>
    <nagle>0</nagle>
    <back_win_sca>0</back_win_sca>
    <mss>0</mss>
  </tcp_proxy_template>
</response>
```

Menus Privilege

Config Mode >> Service >> Template >> TCP Proxy

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.23.3 “slb.template.tcp_proxy.create” Method

This method is used to create a TCP proxy template.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.template.tcp_proxy.create	String		
name (*)	tcp proxy template name	String		
fin_timeout	fin timeout (seconds)	Int	1 – 60	5
idle_timeout	idle timeout (seconds)	Int	60 – 120000	60
force_del_timeout	force delete timeout (seconds)	Int	0 – 31	0
half_close_idle_timeout	half close idle timeout (seconds)	Int	0, 60 – 15000	0
retransmit_retries	retransmit retries	Int	1 – 20	3
syn_retries	SYN tries	Int	1 – 20	5
time_wait	time wait	Int	1 – 60	5
receive_buffer	receive buffer(bytes)	Int	1-2147483647	87380
transmit_buffer	transit buffer (bytes)	Int	1-2147483647	16384
init_window_size	initial window size, disabled(0)	Int	0 – 65535	0
nagle	disabled(0) or enabled	Int	0 or 1	0
back_win_sca	backend window scaling, disabled(0)	Int	0 – 14	0
mss	mss, disabled(0)	Int	0, 128 – 4312	0

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.template.tcp_proxy.create&name=tcp_proxy_template_name1&fin_time=10&idle_timeout=10&nagle=1
```

CLI configuration after aXAPI call:

```
!
slb template tcp-proxy tcp_proxy_template_name1
    idle-timeout 10
    fin-timeout 10
    nagle
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> Template >> TCP Proxy

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.23.4 “slb.template.tcp_proxy.update” Method

This method is used to update one or more parameters in a TCP proxy template.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.template.tcp_proxy.update	String		
name (*)	tcp proxy template name	String		
fin_timeout	fin timeout (seconds)	Int	1 – 60	5
idle_timeout	idle timeout (seconds)	Int	60 – 120000	60
force_del_timeout	force delete timeout (seconds)	Int	0 – 31	0
half_close_idle_timeout	half close idle timeout (seconds)	Int	0, 60 – 15000	0
retransmit_retries	retransmit retries	Int	1 – 20	3
syn_retries	SYN tries	Int	1 – 20	5
time_wait	time wait	Int	1 – 60	5
receive_buffer	receive buffer(bytes)	Int	1-2147483647	87380
transmit_buffer	transit buffer (bytes)	Int	1-2147483647	16384
init_window_size	initial window size, disabled(0)	Int	0 – 65535	0
Nagle	disabled(0) or enabled	Int	0 or 1	0
back_win_sca	backend window scaling, disabled(0)	Int	0 – 14	0
Mss	mss, disabled(0)	Int	0, 128 – 4312	0

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
slb template tcp-proxy tcp_proxy_template_name1
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.template.tcp_proxy.update&name=tcp_proxy_template_name1&fin_time=10&idle_timeout=10&nagle=1
```

CLI configuration after aXAPI call:

```
!
slb template tcp-proxy tcp_proxy_template_name1
!
slb template tcp-proxy tcp_proxy_template_name2
  idle-timeout 10
  fin-timeout 10
  nagle
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> Template >> TCP Proxy

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.23.5 “slb.template.tcp_proxy.delete” Method

This method is used to delete a TCP proxy template.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.template.tcp_proxy.delete	String		
name (*)	tcp proxy template name	String		

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
slb template tcp-proxy tcp_proxy_template_name1
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.template.tcp_proxy.delete&name=tcp_proxy_template_name1
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> Template >> TCP Proxy

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.24 DNS Template

7.24.1 “slb.template.dns.getAll” Method

This method is used to get configuration information for all DNS templates configured on the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.template.dns.getAll	String		

Note: (*) parameter is required in the API.

Response Fields

dns_template_list	XML tag for the collection of udp templates
dns_template	XML tag for udp template
name	dns template name
malformed_query	malformed query, disabled(0), drop(1), forward to service group(2)
service_group_malformed_query	service group name, only malformed_query is 2
status	dns template status, disabled(0) or enabled(1)
def_policy	default policy, no cache(0) or cache(1)
log_period	log period (Minutes)
max_cache_size	
class_list	XML tag for the class list
name	class list name
lid_list	XML tag for the collection of LID
lid	XML tag for LID
id	LID id
dns_cache_status	DNS cache status, enabled(1) or disabled(0)
ttl	
weight	
conn_rate_limit	connection rate limit
conn_rate_limit_per	connection rate limit interval
over_limit_action	drop(0), forward(1), enable DNS cache(2) or disable dns cache(3)
lockout	lockout
log_status	log status, enabled(1). disabled(0)
log_interval	log interval

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
class-list class_list_name
!
!
slb template dns dns template name1
```

```

malformed-query forward srvgrp1
class-list name class_list_name
class-list lid 2
  dns cache-enable
  conn-rate-limit 100000 per 100
  over-limit-action dns-cache-enable lockout 10 log 100
class-list lid 3
  dns cache-enable
  conn-rate-limit 100000 per 100
  over-limit-action dns-cache-enable lockout 10 log 100
!
slb template dns dns_template_name2
  class-list name class_list_name
  class-list lid 2
    dns cache-enable
    conn-rate-limit 100000 per 100
    over-limit-action dns-cache-enable lockout 10 log 100
  class-list lid 3
    dns cache-enable
    conn-rate-limit 100000 per 100
    over-limit-action dns-cache-enable lockout 10 log 100
!

```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxxxx&method=slb.template.dns.getAll

Response as the HTTP body:

```

<response status="ok">
  <dns_template_list>
    <dns_template>
      <name>dns_template_name1</name>
      <malformed_query>1</malformed_query>
      <service_group_malformed_query>srvgrp1</service_group_malformed_query>
      <status>1</status>
      <def_policy>0</def_policy>
      <log_period>10</log_period>
      <max_cache_size>100</max_cache_size>
      <class_list>
        <name>class_list_name</name>
        <lid_list>
          <lid>
            <id>2</id>
            <dns_cache_status>1</dns_cache_status>
            <ttl>300</ttl>
            <weight>1</weight>
            <conn_rate_limit>100000</conn_rate_limit>
            <conn_rate_limit_per>100</conn_rate_limit_per>
            <over_limit_action>2</over_limit_action>
            <lockout>10</lockout>
            <log_status>1</log_status>
            <log_interval>100</log_interval>
          </lid>
          <lid>
            <id>3</id>
            <dns_cache_status>1</dns_cache_status>
            <ttl>300</ttl>
            <weight>1</weight>
            <conn_rate_limit>100000</conn_rate_limit>
            <conn_rate_limit_per>100</conn_rate_limit_per>
            <over_limit_action>2</over_limit_action>
            <lockout>10</lockout>
            <log_status>1</log_status>
            <log_interval>100</log_interval>
          </lid>
        </lid_list>
      </class_list>
    </dns_template>
    <dns_template>
      <name>dns_template_name2</name>

```

```

<malformed_query>0</malformed_query>
<status>1</status>
<def_policy>0</def_policy>
<log_period>10</log_period>
<max_cache_size>100</max_cache_size>
<class_list>
  <name>class_list_name</name>
  <lid_list>
    <lid>
      <id>2</id>
      <dns_cache_status>1</dns_cache_status>
      <ttl>300</ttl>
      <weight>1</weight>
      <conn_rate_limit>100000</conn_rate_limit>
      <conn_rate_limit_per>100</conn_rate_limit_per>
      <over_limit_action>2</over_limit_action>
      <lockout>10</lockout>
      <log_status>1</log_status>
      <log_interval>100</log_interval>
    </lid>
    <lid>
      <id>3</id>
      <dns_cache_status>1</dns_cache_status>
      <ttl>300</ttl>
      <weight>1</weight>
      <conn_rate_limit>100000</conn_rate_limit>
      <conn_rate_limit_per>100</conn_rate_limit_per>
      <over_limit_action>2</over_limit_action>
      <lockout>10</lockout>
      <log_status>1</log_status>
      <log_interval>100</log_interval>
    </lid>
  </lid_list>
</class_list>
</dns_template>
</dns_template_list>
</response>

```

Menus Privilege

Config Mode >> Service >> Template >> Application >> DNS

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.24.2 “slb.template.dns.search” Method

This method is used to get DNS template information by given DNS template name.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.template.dns.search	String		
name	dns template name	String		

Note: (*) parameter is required in the API.

Response Fields

dns_template	XML tag for UDP template
name	DNS template name
malformed_query	malformed query, disabled(0), drop(1), forward to service group(2)
service_group_malformed_query	service group name, only malformed_query is 2

status	dns template status, disabled(0) or enabled(1)
def_policy	default policy, no cache(0) or cache(1)
log_period	log period (Minutes)
max_cache_size	
class_list	XML tag for the class list
name	class list name
lid_list	XML tag for the collection of LID
lid	XML tag for LID
id	LID ID
dns_cache_status	DNS cache status, enabled(1) or disabled(0)
ttl	
weight	
conn_rate_limit	connection rate limit
conn_rate_limit_per	connection rate limit interval
over_limit_action	drop(0), forward(1), enable dns cache(2) or disable dns cache(3)
lockout	lockout
log_status	log status, enabled(1). disabled(0)
log_interval	log interval

Example Response

Request as:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
- **HTTP Action:**
GET

CLI configuration before aXAPI call:

```
!
class-list class_list_name
!
slb service-group srvgrp1 udp
!
slb template dns dns_template_name1
  malformed-query forward srvgrp1
  class-list name class_list_name
  class-list lid 2
    dns cache-enable
    conn-rate-limit 100000 per 100
    over-limit-action dns-cache-enable lockout 10 log 100
  class-list lid 3
    dns cache-enable
    conn-rate-limit 100000 per 100
    over-limit-action dns-cache-enable lockout 10 log 100
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxxxx&method=slb.template.dns.search&name=dns_template_name1

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
  <dns_template>
    <name>dns_template_name1</name>
    <malformed_query>1</malformed_query>
    <service_group_malformed_query>srvgrp1</service_group_malformed_query>
    <status>1</status>
    <def_policy>0</def_policy>
```

```

<log_period>10</log_period>
<max_cache_size>100</max_cache_size>
<class_list>
  <name>class_list_name</name>
  <lid_list>
    <lid>
      <id>2</id>
      <dns_cache_status>1</dns_cache_status>
      <ttl>300</ttl>
      <weight>1</weight>
      <conn_rate_limit>100000</conn_rate_limit>
      <conn_rate_limit_per>100</conn_rate_limit_per>
      <over_limit_action>2</over_limit_action>
      <lockout>10</lockout>
      <log_status>1</log_status>
      <log_interval>100</log_interval>
    </lid>
    <lid>
      <id>3</id>
      <dns_cache_status>1</dns_cache_status>
      <ttl>300</ttl>
      <weight>1</weight>
      <conn_rate_limit>100000</conn_rate_limit>
      <conn_rate_limit_per>100</conn_rate_limit_per>
      <over_limit_action>2</over_limit_action>
      <lockout>10</lockout>
      <log_status>1</log_status>
      <log_interval>100</log_interval>
    </lid>
  </lid_list>
</class_list>
</dns_template>
</response>

```

Menus Privilege

Config Mode >> Service >> Template >> Application >> DNS

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.24.3 “slb.template.dns.create” Method

This method is used to create a DNS template.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.template.dns.create	String		
name (*)	dns template name	String		
malformed_query	disabled(0), drop(1), forward to service group(2)	Int	0, 1 or 2	0
service_group_malformed_query	service group name	String		
status	disabled(0), enabled(1)	Int	0 or 1	1
def_policy	default policy, no cache(0), cache(1)	Int	0 or 1	0
log_period	log period	Int	0 - 10000	0
max_cache_size	max cache size	Int	0 - 2000000	0
class_list	class list name	String		
lid_list (*)	LID list to be added lid1^B ...lidN ^B: ASCII Code 0x02, URL-encode %02	String		
lid<n>	LID at element <n>	String		

id (*)	LID ID	Int	1 - 31	
dns_cache_status	enabled(1) or disabled(0)	Int	0 or 1	0
ttl	TTL	Int	1 - 65535	300
weight	weight	Int	1 - 7	1
conn_rate_limit	connection rate limit	Int	1-2147483647	
conn_rate_limit_per	connection rate limit per (100ms)	Int	1 – 65535	
over_limit_action	drop(0), forward(1), enable dns cache(2) or disable dns cache(3)	Int	0, 1, 2 or 3	0
lockout	lockout	Int	1 – 1023	0
log_status	disabled(0) or enabled(1)	Int	0 or 1	0
log_interval	log interval	Int	1 - 255	0

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
[https://\[AX_IP_ADDRESS\]/services/rest/V2/](https://[AX_IP_ADDRESS]/services/rest/V2/)
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
class-list class_list_name
!
slb service-group aaa udp
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.template.dns.create&name=dns_template_name1
&malformed_query=2&service_group_malformed_query=aaa&class_list=class_list_name&lid_li
st=lid1%02lid2&lid1=id%031%02conn_rate_limit%031%02conn_rate_limit_per%031%02lockout
%031&lid2=id%032%02conn_rate_limit%0310000%02conn_rate_limit_per%03100%02lockout%03
9
```

CLI configuration after aXAPI call:

```
!
class-list class_list_name
!
slb service-group aaa udp
!
slb template dns dns_template_name1
    malformed-query forward aaa
    class-list name class_list_name
    class-list lid 1
        dns weight 7
        conn-rate-limit 1 per 1
        over-limit-action lockout 1
    class-list lid 2
        dns weight 7
        conn-rate-limit 10000 per 100
        over-limit-action lockout 9
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> Template >> Application >> DNS

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.24.4 “slb.template.dns.update” Method

This method is used to update one or more parameters in a DNS template.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.template.dns.update	String		
name (*)	dns template name	String		
malformed_query	disabled(0), drop(1), forward to service group(2)	Int	0, 1 or 2	0
service_group_malformed_query	service group name	String		
status	disabled(0), enabled(1)	Int	0 or 1	1
def_policy	default policy, no cache(0), cache(1)	Int	0 or 1	0
log_period	log period	Int	0 - 10000	0
max_cache_size	max cache size	Int	0 - 2000000	0
class_list	class list name	String		
lid_list (*)	LID list to be added lid1^B ...lidN ^B: ASCII Code 0x02, URL-encode %02	String		
lid<n> id (*) dns_cache_status ttl weight conn_rate_limit conn_rate_limit_per over_limit_action lockout log_status log_interval	LID at element <n> LID ID enabled(1) or disabled(0) TTL weight connection rate limit connection rate limit per (100ms) drop(0), forward(1), enable dns cache(2) or disable dns cache(3) lockout disabled(0) or enabled(1) log interval	String Int Int Int Int Int Int Int Int Int Int Int Int Int	1 - 31 0 or 1 1 - 65535 1 - 7 1-2147483647 1 – 65535 0, 1, 2 or 3 1 – 1023 0 or 1 1 - 255	0 300 1 0 0 0 0 0 0 0 0 0 0 0

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.template.dns.update&name=dns_template_name1
&malformed_query=2&service_group_malformed_query=aaa&class_list=class_list_name&lid_li
st=lid1%02lid2&lid1=id%031%02conn_rate_limit%031%02conn_rate_limit_per%031%02lockout
%031&lid2=id%032%02conn_rate_limit%0310000%02conn_rate_limit_per%03100%02lockout%03
9
```

CLI configuration after aXAPI call:

```
!
class-list class_list_name
!
slb service-group aaa udp
!
slb template dns dns_template_name1
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> Template >> Application >> DNS

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.24.5 “slb.template.dns.delete” Method

This method is used to delete a DNS template.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.template.dns.delete	String		
name (*)	dns template name	String		

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
- **HTTP Action:**
POST

CLI configuration before aXAPI call:

```
!
slb template dns dns_template_name1
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxxxx&method=slb.template.dns.delete&name=dns_template_name1
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> Template >> Application >> DNS

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.25 Diameter Template

7.25.1 “slb.template.diameter.getAll” Method

This method is used to get configuration information for all Diameter templates configured on the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.template.diameter.getAll	String		

Note: (*) parameter is required in the API.

Response Fields

diameter_template_list	XML tag for the collection of diameter templates
diameter_template	XML tag for diameter template
name	diameter template name
multiple_origin_host	multiple origin host
origin_host	origin host
origin_realm	origin realm
product_name	product name
vendor_id	vendor id
idle_timeout	idle timeout
dwr_time_interval	dwr time interval
session_age	session age
customizing_cea_response	customizing cea response
duplicate_avp_code	duplicate avp code
duplicate_pattern	duplicate pattern
duplicate_service_name	duplicate service name
avps	XML tag for collection of avps
avp	XML tag for avp
code	code
mandatory	mandatory
type	INT32 (1), INT64(2), String(3)
value	value
message_codes	XML tag for collection of message codes
code	XML tag for message code
value	message code

Example Response

Request as:

- **URL:**
[https://\[AX IP ADDRESS\]/services/rest/v2/](https://[AX IP ADDRESS]/services/rest/v2/)
- **HTTP Action:**
GET

CLI configuration before aXAPI call:

```
!
slb service-group Test udp
!
slb template diameter diameter_template_name1
  origin-host 2.2.2.2
  origin-realm aa
  customize-cea
  multiple-origin-host
  product-name bbb
  vendor-id 12345
  idle-timeout 5
  session-age 10
  dwr-time 10
  duplicate 234 asdfqw Test
  message-code 123
  message-code 124
  avp 1 int32 0
  avp 23 mandatory int64 1234567
  avp 345 mandatory string asdf
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxxxx&method=slb.template.diameter.getAll

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
  <diameter_template_list>
    <diameter_template_list>
      <name>diameter_template_name1</name>
      <multiple_origin_host>1</multiple_origin_host>
      <origin_host>2.2.2.2</origin_host>
      <origin_realm>aa</origin_realm>
      <product_name>bbb</product_name>
      <vendor_id>12345</vendor_id>
      <idle_timeout>5</idle_timeout>
      <dwr_time_interval>10</dwr_time_interval>
      <session_age>10</session_age>
      <customizing_cea_response>1</customizing_cea_response>
      <duplicate_avp_code>234</duplicate_avp_code>
      <duplicate_pattern>asdfqw</duplicate_pattern>
      <duplicate_service_name>Test</duplicate_service_name>
      <avps>
        <avp>
          <code>1</code>
          <mandatory>0</mandatory>
          <type>1</type>
          <value>0</value>
        </avp>
        <avp>
          <code>23</code>
          <mandatory>1</mandatory>
          <type>2</type>
          <value>1234567</value>
        </avp>
        <avp>
          <code>345</code>
          <mandatory>1</mandatory>
          <type>3</type>
          <value>asdf</value>
        </avp>
      </avps>
    </diameter_template_list>
  </diameter_template_list>
  <message_codes>
```

```

<code>
  <value>123</value>
</code>
<code>
  <value>124</value>
</code>
</message_codes>
</diameter_template>
</diameter_template_list>
</response>

```

Menus Privilege

Config Mode >> Service >> Template >> Application >> Diameter

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.25.2 “slb.template.diameter.search” Method

This method is used to get Diameter template information by given template name.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.template.diameter.getAll	String		
name (*)	diameter template name	String		

Note: (*) parameter is required in the API.

Response Fields

diameter_template	XML tag for diameter template
name	diameter template name
multiple_origin_host	multiple origin host
origin_host	origin host
origin_realm	origin realm
product_name	product name
vendor_id	vendor id
idle_timeout	idle timeout
dwr_time_interval	dwr time interval
session_age	session age
customizing_cea_response	customizing cea response
duplicate_avp_code	duplicate avp code
duplicate_pattern	duplicate pattern
duplicate_service_name	duplicate service name
avps	XML tag for collection of avp
avp	XML tag for avp
code	code
mandatory	mandatory
type	INT32 (1), INT64(2), String(3)
value	value
message_codes	XML tag for collection of message code
code	XML tag for message code
value	message code

Example Response

Request as:

- URL:
[https://\[AX_IP_ADDRESS\]/services/rest/v2/](https://[AX_IP_ADDRESS]/services/rest/v2/)
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
! slb service-group Test udp
!
slb template diameter diameter_template_name1
  origin-host 2.2.2.2
  origin-realm aa
  customize-cea
  multiple-origin-host
  product-name bbb
  vendor-id 12345
  idle-timeout 5
  session-age 10
  dwr-time 10
  duplicate 234 asdfqw Test
  message-code 123
  message-code 124
  avp 1 int32 0
  avp 23 mandatory int64 1234567
  avp 345 mandatory string asdf
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.template.diameter.search&name=diameter_template_name1
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
  <diameter_template_list>
    <name>diameter_template_name1</name>
    <multiple_origin_host>1</multiple_origin_host>
    <origin_host>2.2.2.2</origin_host>
    <origin_realm>aa</origin_realm>
    <product_name>bbb</product_name>
    <vendor_id>12345</vendor_id>
    <idle_timeout>5</idle_timeout>
    <dwr_time_interval>10</dwr_time_interval>
    <session_age>10</session_age>
    <customizing_cea_response>1</customizing_cea_response>
    <duplicate_avp_code>234</duplicate_avp_code>
    <duplicate_pattern>asdfqw</duplicate_pattern>
    <duplicate_service_name>Test</duplicate_service_name>
    <avps>
      <avp>
        <code>1</code>
        <mandatory>0</mandatory>
        <type>1</type>
        <value>0</value>
      </avp>
      <avp>
        <code>23</code>
        <mandatory>1</mandatory>
        <type>2</type>
        <value>1234567</value>
      </avp>
      <avp>
        <code>345</code>
      </avp>
    </avps>
  </diameter_template_list>
</response>
```

```

<mandatory>1</mandatory>
<type>3</type>
<value>asdf</value>
</avp>
</avps>
<message_codes>
<code>
<value>123</value>
</code>
<code>
<value>124</value>
</code>
</message_codes>
</diameter_template>
</response>

```

Menus Privilege

Config Mode >> Service >> Template >> Application >> Diameter

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.25.3 “slb.template.diameter.create” Method

This method is used to create a Diameter template.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.template.diameter.create	String		
name (*)	diameter template name	String		
multiple_origin_host	multiple origin host	Int	0 or 1	0
origin_host	origin host	String		
origin_realm	origin realm	String		
product_name	product name	String		
vendor_id	vendor id	Int	0 – 2147483647	0
idle_timeout	idle timeout	Int	1 - 65535	5
dwr_time_interval	dwr time interval	Int	0 – 2147483647	10
session_age	session age	Int	1 - 65535	10
customizing_cea_response	customizing cea response	Int	0 or 1	0
duplicate_avp_code	duplicate avp code	Int	0 – 2147483647	
duplicate_pattern	duplicate pattern	String		
duplicate_service_name	duplicate service name	String		
avp_list				
avp<n>	header erase request at element <n>			
code	code	Int		
mandatory	mandatory	Int	0 or 1	
type	type	Int	1, 2, 3	
value	value	Int32, int64,string		
message_code_list				
code<n>	header erase request at element <n>			
value	value	Int	0 – 2147483647	

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
[https://\[AX IP ADDRESS\]/services/rest/v2/](https://[AX IP ADDRESS]/services/rest/v2/)
- HTTP Action:
POST

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxxx&method=slb.template.diameter.create&name=diameter_template_name1&multiple_origin_host=1&origin_host=3.3.3.2&origin_realm=abcdef&product_name=prod_name1&vendor_id=1234&idle_timeout=23&dwr_time_interval=34&session_age=21&customizing_cea_response=1&duplicate_avp_code=321&duplicate_pattern=dup_pattern1&duplicate_service_name=Test&avp_list=avp1%02avp2&avp1=code%03234%02mandatory%031%02type%031%02value%03111&avp2=code%03432%02mandatory%030%02type%032%02value%03112&message_code_list=code1%02code2&code1=code%03456&code2=code%03654
```

CLI configuration after aXAPI call:

```
!
slb service-group Test udp
!
slb template diameter diameter_template_name1
    origin-host 3.3.3.2
    origin-realm abcdef
    customize-cea
    multiple-origin-host
    product-name prod_name1
    vendor-id 1234
    idle-timeout 23
    session-age 21
    dwr-time 34
    duplicate 321 dup_pattern1 Test
    message-code 456
    message-code 654
    avp 234 mandatory int32 111
    avp 432 int64 112
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> Template >> Application >> Diameter

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.25.4 “slb.template.diameter.update” Method

This method is used to update one or more parameters in a Diameter template.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.template.diameter.update	String		
name (*)	template name	String		
multiple_origin_host	multiple origin host	Int	0 or 1	0
origin_host	origin host	String		

origin_realm	origin realm	String		
product_name	product name	String		
vendor_id	vendor id	Int	0 – 2147483647	0
idle_timeout	idle timeout	Int	1 - 65535	5
dwr_time_interval	dwr time interval	Int	0 – 2147483647	10
session_age	session age	Int	1 - 65535	10
customizing_cea_response	customizing cea response	Int	0 or 1	0
duplicate_avp_code	duplicate avp code	Int	0 – 2147483647	
duplicate_pattern	duplicate pattern	String		
duplicate_service_name	duplicate service name	String		
avp list				
avp<n> code mandatory type value	header erase request at element <n> code mandatory type value	Int Int Int Int32, int64,string	0 or 1 1, 2, 3	
message code list				
code<n> value	header erase request at element <n> value	Int	0 – 2147483647	

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
[https://\[AX IP ADDRESS\]/services/rest/V2/](https://[AX IP ADDRESS]/services/rest/V2/)
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
slb service-group Test udp
!
slb template diameter diameter_template_name1
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxxmethod=slb.template.diameter.update&name=diameter_template_name1&multiple_origin_host=1&origin_host=3.3.3.2&origin_realm=abcdef&product_name=prod_name1&vendor_id=1234&idle_timeout=23&dwr_time_interval=34&session_age=21&customizing_cea_response=1&duplicate_avp_code=321&duplicate_pattern=dup_pattern1&duplicate_service_name=Test&avp_list=avp1%02avp2&avp1=code%03234%02mandatory%031%02type%031%02value%03111&avp2=code%03432%02mandatory%030%02type%032%02value%03112&message_code_list=code1%02code2&code1=code%03456&code2=code%03654
```

CLI configuration after aXAPI call:

```
!
slb service-group Test udp
!
slb template diameter diameter_template_name1
    origin-host 3.3.3.2
    origin-realm abcdef
    customize-cea
    multiple-origin-host
    product-name prod_name1
    vendor-id 1234
    idle-timeout 23
    session-age 21
    dwr-time 34
    duplicate 321 dup_pattern1 Test
    message-code 456
    message-code 654
    avp 234 mandatory int32 111
```

```
! avp 432 int64 112
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> Template >> Application >> Diameter

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.25.5 “slb.template.diameter.delete” Method

This method is used to delete a Diameter template.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.template.diameter.delete	String		
name (*)	diameter template name	String		

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
[https://\[AX_IP_ADDRESS\]/services/rest/V2/](https://[AX_IP_ADDRESS]/services/rest/V2/)
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
slb template diameter diameter_template_name1
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.template.diameter.delete&name=diameter_template_name1
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> Template >> Application >> Diameter

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.26 Health Monitor

7.26.1 “slb.hm.getAll” Method

This method is used to get configuration information for all SLB health monitors configured on the AX.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.hm.getAll	String		

Note: (*) parameter is required in the API.

Response Fields

health_monitor_list	XML tag for the collection of health monitors
health_monitor	XML tag for health monitor
name	health monitor name
retry	health check retry
consec_pass_reqd	health check retry before declaring target up
interval	health check interval (seconds)
time_out	health check timeout (seconds)
strictly_retry	strictly retry
disable_after_down	disable the target if health check failed
override_ipv4	override IPv4
override_ipv6	override IPv6
override_port	override port
type	icmp, tcp, udp, http, https, ftp, smtp, pop3, snmp, dns, radius, ldap, rtsp, sip, ntp, imap, database, compound, external

The following XML tags are for health monitor ICMP method type:

mode	transparent mode
alias_addr	IPv4 or IPv6 alias address

The following XML tags are for health monitor TCP method type:

port	TCP port
half_open	false(0) or true(1)

The following XML tags are for health monitor UDP method type:

port	UDP port
------	----------

The following XML tags are for health monitor HTTP method type:

port	HTTP port
host	hostname
url	URL string, it must begin with GET /, HEAD / or POST /
post_file	HTTP POST file, only when URL begins with POST /
post_data	HTTP POST data, only when URL begins with POST /
user	user name

password	password
expect	what you expect from the response message
pattern	text expected
code	response code
maintenance_code	maintenance code

The following XML tags are for health monitor HTTPS method type:

port	HTTPs port
host	hostname
url	URL string, it must begin with GET /, HEAD / or POST /
post_file	HTTP POST file, only when URL begins with POST /
post_data	HTTP POST data, only when URL begins with POST /
user	user name
password	password
expect	what you expect from the response message
pattern	text expected
code	response code
maintenance_code	maintenance code

The following XML tags are for health monitor FTP method type:

port	FTP port
user	user name
password	password

The following XML tags are for health monitor SMTP method type:

port	SMTP port
domain	domain name

The following XML tags are for health monitor POP3 method type:

port	POP3 port
user	user name
password	password

The following XML tags are for health monitor SNMP method type:

port	SNMP port
operation	operation, get or get_next
oid	OID
community	community

The following XML tags are for health monitor DNS method type:

port	DNS port
domain	domain name
ip_addr.	IPv4 or IPv6 address
record_type	query type, it can be A, CNAME, SOA, PTR, MX, TXT, or AAAA, only when domain name is inputted
recursion	disabled(0) or enabled(1)
expect	what you expect from the response message

The following XML tags are for health monitor RADIUS method type:

port	RADIUS port
user	user name

password	password
secret	the shared secret of RADIUS server

The following XML tags are for health monitor LDAP method type:

port	LDAP port
ssl	disabled(0) or enabled(1)
distinguished_name	distinguished name
password	password

The following XML tags are for health monitor RTSP method type:

port	RTSP port
url	URL string, it must begin with /

The following XML tags are for health monitor SIP method type:

port	SIP port
register	disabled(0) or enabled(1)
tcp	disabled(0) or enabled(1)

The following XML tags are for health monitor NTP method type:

port	NTP port
------	----------

The following XML tags are for health monitor IMAP method type:

port	IMAP port
user	user name
password	password
auth_method	authentication method
plain_text	plain text, disabled(0) or enabled(1)
login	login, disabled(0) or enabled(1)
cram_md5	challenge-response authentication mechanism, disabled(0) or enabled(1)

The following XML tags are for health monitor DATABASE method type:

db_name	database name
user	user name
password	password
sql	SQL Query
res_string	respond string
row	
column	

The following XML tags are for health monitor Compound method type:

compound	Reverse Polish Notation
----------	-------------------------

The following XML tags are for health monitor external method type:

program	program name
arguments	program arguments
server_port	server port

Example Response

Request as:

- URL:

- **HTTP Action:**
GET
- **HTTP Body**
`session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.hm.getAll`

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
<health_monitor_list>
  <health_monitor>
    <name>ping</name>
    <retry>3</retry>
    <consec_pass_reqd>1</consec_pass_reqd>
    <interval>5</interval>
    <time_out>5</time_out>
    <strictly_retry>0</strictly_retry>
    <disable_after_down>0</disable_after_down>
    <override_ipv4/>
    <override_ipv6/>
    <override_port/>
    <type>icmp</type>
    <mode>reverse</mode>
  </health_monitor>
  <health_monitor>
    <name>hm1</name>
    <retry>4</retry>
    <consec_pass_reqd>4</consec_pass_reqd>
    <interval>4</interval>
    <time_out>4</time_out>
    <strictly_retry>0</strictly_retry>
    <disable_after_down>0</disable_after_down>
    <override_ipv4/>
    <override_ipv6/>
    <override_port/>
    <type>http</type>
    <port>88</port>
    <host>a.b.c</host>
    <url>GET /abc</url>
    <user/>
    <password/>
    <expect>
      <pattern>dddd</pattern>
      <maintenance_code>199</maintenance_code>
    </expect>
  </health_monitor>
</health_monitor_list>
</response>
```

Menus Privilege

Config Mode >> Service >> Health Monitor >> Health Monitor

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.26.2 “slb.hm.search” Method

This method is used to get health monitoring information for a given health monitor name.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.hm.search	String		

name (*)	health monitor name	String		
-----------------	---------------------	--------	--	--

Note: (*) parameter is required in the API.

Response Fields

health_monitor	XML tag for health monitor
name	health monitor name
retry	health check retry
consec_pass_reqd	health check retry before declaring target up
interval	health check interval (seconds)
time_out	health check timeout (seconds)
strictly_retry	strictly retry
disable_after_down	disable the target if health check failed
override_ip4	override IPv4
override_ip6	override IPv6
override_port	override port
type	icmp, tcp, udp, http, https, ftp, smtp, pop3, snmp, dns, radius, ldap, rtsp, sip, ntp, imap, database, compound, external

The following XML tags are for health monitor ICMP method:

mode	transparent mode
alias_addr	IPv4 or IPv6 alias address

The following XML tags are for health monitor TCP method:

port	TCP port
half_open	false(0) or true(1)

The following XML tags are for health monitor UDP method:

port	UDP port
------	----------

The following XML tags are for health monitor HTTP method:

port	HTTP port
host	hostname
url	URL string, must begin with GET /, HEAD / or POST /
post_file	HTTP POST file, only when URL begins with POST /
post_data	HTTP POST data, only when URL begins with POST /
user	user name
password	password
expect	what you expect from the response message
pattern	text expected
code	response code
maintenance_code	maintenance code

The following XML tags are for health monitor HTTPS method:

port	HTTPs port
host	hostname
url	URL string, must begin with GET /, HEAD / or POST /
post_file	HTTP POST file, only when URL begins with POST /
post_data	HTTP POST data, only when URL begins with POST /
user	user name
password	password

expect	what you expect from the response message
pattern	text expected
code	response code
maintenance_code	maintenance code

The following XML tags are for health monitor FTP method:

port	FTP port
user	user name
password	password

The following XML tags are for health monitor SMTP method:

port	SMTP port
domain	domain name

The following XML tags are for health monitor POP3 method:

port	POP3 port
user	user name
password	password

The following XML tags are for health monitor SNMP method:

port	SNMP port
operation	operation, get or get_next
oid	OID
community	community

The following XML tags are for health monitor DNS method:

port	DNS port
domain	domain name
ip_addr.	IPv4 or IPv6 address
record_type	query type, it can be A, CNAME, SOA, PTR, MX, TXT, or AAAA, only when domain name is inputted
recursion	disabled(0) or enabled(1)
expect	what you expect from the response message

The following XML tags are for health monitor RADIUS method:

port	RADIUS port
user	user name
password	password
secret	the shared secret of RADIUS server

The following XML tags are for health monitor LDAP method:

port	LDAP port
ssl	disabled(0) or enabled(1)
distinguished_name	distinguished name
password	password

The following XML tags are for health monitor RTSP method:

port	RTSP port
url	URL string, it must begin with /

The following XML tags are for health monitor SIP method type:

port	SIP port
register	disabled(0) or enabled(1)
tcp	disabled(0) or enabled(1)

The following XML tags are for health monitor IMAP method:

port	IMAP port
user	user name
password	password
auth_method	authentication method
plain_text	plain text, disabled(0) or enabled(1)
login	login, disabled(0) or enabled(1)
cram_md5	challenge-response authentication mechanism, disabled(0) or enabled(1)

The following XML tags are for health monitor DATABASE method type:

db_name	database name
user	user name
password	password
sql	SQL Query
res_string	respond string
row	
column	

The following XML tags are for health monitor external method:

Example Response

Example:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
 - **HTTP Action:**
GET
 - **HTTP Body**
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb_hm_search&name=hm1

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
<health_monitor_list>
    <health_monitor>
        <name>hml</name>
        <retry>4</retry>
        <consec_pass_reqd>4</consec_pass_reqd>
        <interval>4</interval>
        <time_out>4</time_out>
        <strictly_retry>0</strictly_retry>
        <disable_after_down>0</disable_after_down>
        <override_ipv4/>
        <override_ipv6/>
    </health_monitor>
</health_monitor_list>
</response>
```

```

<override_port/>
<type>http</type>
<port>88</port>
<host>a.b.c</host>
<url>GET /abc</url>
<user/>
<password/>
<expect>
    <pattern>dddd</pattern>
    <maintenance_code>199</maintenance_code>
</expect>
</health_monitor>
</health_monitor_list>
</response>

```

Menus Privilege

Config Mode >> Service >> Health Monitor >> Health Monitor

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.26.3 “slb.hm.create” Method

This method is used to create an SLB health monitor.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.hm.create	String		
name (*)	health monitor name	String		
retry	retry	Int	1 – 5	3
consec_pass_reqd	consec pass req'd	Int	1 – 10	1
interval	Interval	Int	1 – 180	5
time_out	timeout	Int	1 – 12	5
strictly_retry	strictly retry, disabled(0) or enabled(1)	Int	0 or 1	0
disable_after_down	disabled after down, disabled(0) or enabled(1)	Int	0 or 1	0
override_ip4	override IPv4	String		
override_ip6	override IPv6	String		
override_port	override port	Int	1 - 65534	
type	Icmp, tcp, udp, http, https, ftp, smtp, pop3,snmp, dns, radius, ldap, rtsp, sip, ntp, imap, database, compound, external	String		
Following parameters are for health monitor ICMP method				
mode	mode is transparent if there is alias address	String		
alias_addr	alias IPv4 or IPv6 address	String		
Following parameters are for health monitor TCP method				
port	TCP port	Int	1 – 65534	80
half_open	disabled(0) or enabled(1)	Int	0 or 1	0
Following parameters are for health monitor UDP method				
port	UDP port	Int	1 – 65534	61
Following parameters are for health monitor HTTP method				
port	HTTP port	Int	1 – 65534	80
host	Hostname	String		
url	URL strings	String		GET /
post_file	HTTP POST file	String		
post_data	post data	String		
user	user name	String		
password	password	String		
pattern	expect text	String		
code	expect code	String		

maintenance_code	maintenance code	String		
Following parameters are for health monitor HTTPS method				
port	HTTPs port	Int	1 – 65534	443
host	Hostname	String		
url	URL strings	String		GET /
post_file	HTTP POST file	String		
post_data	post data	String		
user	user name	String		
password	password	String		
pattern	expect text	String		
code	expect code	String		
maintenance_code	maintenance code	String		
Following parameters are for health monitor FTP method				
port	FTP port	Int	1 – 65534	21
user	user name	String		
password	password	String		
Following parameters are for health monitor SMTP method				
port	SMTP port	Int	1 – 65534	25
domain	domain name	String		A10
Following parameters are for health monitor POP3 method				
port	POP3 port	Int	1 – 65534	110
user	username	String		a10
password	password	String		
Following parameters are for health monitor SNMP method				
port	SNMP port	Int	1 – 65534	161
operation	operation	String		get
oid	OID	String		1.1.0
community	community	String		public
Following parameters are for health monitor DNS method				
port	DNS port	Int	1 – 65534	53
domain	domain name	String		www.a1 Onetwor ks.com
ip_addr	IPv4 or IPv6 address	String		
record_type	query type	String		A
recursion	recursion, disabled(0) or enabled(1)	Int	0 or 1	1
expect	expect	String		
Following parameters are for health monitor RADIUS method				
port	RADIUS port	Int	1 – 65534	1812
user	user name	String		a10
password	password	String		
secret	secret	String		a10
Following parameters are for health monitor LDAP method				
port	LDAP port	Int	1 – 65534	389
ssl	SSL, disabled(0) or enabled(1)	Int	0 or 1	0
distinguished_name	distinguished name	String		
password	password	String		
Following parameters are for health monitor RTSP method				
port	RTSP port	Int	1 – 65534	554
url	URL string	String		/sample. mpg
Following parameters are for health monitor SIP method				
port	SIP port	Int	1 – 65534	5060
register	register, disabled(0) or enabled(1)	Int	0 or 1	0
tcp	disabled(0) or enabled(1)	Int	0 or 1	0
Following parameters are for health monitor NTP method				
port	NTP port	Int	1 – 65534	123
Following parameters are for health monitor IMAP method				
port	IMAP port	Int	1 – 65534	143
user	user name	String		a10
password	password	String		
plain_text	plain text	Int	0 or 1	0
login	simple login	Int	0 or 1	0
cram_md5	Challenge-Response Authentication Mechanism	Int	0 or 1	0
Following parameters are for health monitor DATABASE method				
db_name	database name	String		

user	user name	String		
password	password	String		
sql	SQL query	String		
res_string	respond string	String		
row	row	Int	1 – 10	
column	column	Int	1 – 10	
Following parameters are for health monitor COMPOUND method				
compound	Reverse Polish Notation	String		
Following parameters are for health monitor EXTERNAL method				
program	program name	String		
arguments	program arguments	String		
server_port	server port	Int	1 – 65534	

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
- **HTTP Action:**
POST
- **HTTP Body**
session_id=xxxxxxxxxxxxxxxxxxxxxx&method=slb.hm.create&name=hml&retry=3&consec_pass_reqd=1&interval=5&time_out=5&strictly_retry=0&disable_after_down=0&override_ipv4=192.168.3.5&override_port=8008&type=http

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> Health Monitor >> Health Monitor

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.26.4 “slb.hm.update” Method

This method is used to update one or more parameters for an SLB health monitor.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.hm.update	String		
name (*)	health monitor name	String		
retry	retry	Int	1 – 5	3
consec_pass_reqd	consec pass req'd	Int	1 – 10	1
interval	Interval	Int	1 – 180	5
time_out	timeout	Int	1 – 12	5
strictly_retry	strictly retry, disabled(0) or enabled(1)	Int	0 or 1	0
disable_after_down	disabled after down, disabled(0) or enabled(1)	Int	0 or 1	0
override_ipv4	override IPv4	String		
override_ipv6	override IPv6	String		

override_port	override port	Int	1 - 65534	
type (*)	Icmp, tcp, udp, http, https, ftp, smtp, pop3,snmp, dns, radius, ldap, rtsp, sip, ntp, imap, database, compound, external	String		
Following parameters are for health monitor ICMP method				
mode	mode is transparent if there is alias address	String		
alias_addr	alias IPv4 or IPv6 address	String		
Following parameters are for health monitor TCP method				
port	TCP port	Int	1 – 65534	80
half_open	disabled(0) or enabled(1)	Int	0 or 1	0
Following parameters are for health monitor UDP method				
port	UDP port	Int	1 – 65534	61
Following parameters are for health monitor HTTP method				
port	HTTP port	Int	1 – 65534	80
host	Hostname	String		
url	URL strings	String		GET /
post_file	HTTP POST file	String		
post_data	post data	String		
user	user name	String		
password	password	String		
pattern	expect text	String		
code	expect code	String		
maintenance_code	maintenance code	String		
Following parameters are for health monitor HTTPS method				
port	HTTPS port	Int	1 – 65534	443
host	Hostname	String		
url	URL strings	String		GET /
post_file	HTTP POST file	String		
post_data	post data	String		
user	user name	String		
password	password	String		
pattern	expect text	String		
code	expect code	String		
maintenance_code	maintenance code	String		
Following parameters are for health monitor FTP method				
port	FTP port	Int	1 – 65534	21
user	user name	String		
password	password	String		
Following parameters are for health monitor SMTP method				
port	SMTP port	Int	1 – 65534	25
domain	domain name	String		A10
Following parameters are for health monitor POP3 method				
port	POP3 port	Int	1 – 65534	110
user	username	String		a10
password	password	String		
Following parameters are for health monitor SNMP method				
port	SNMP port	Int	1 – 65534	161
operation	operation	String		get
oid	OID	String		1.1.0
community	community	String		public
Following parameters are for health monitor DNS method				
port	DNS port	Int	1 – 65534	53
domain	domain name	String		www.a1 Onetwor ks.com
ip_addr	IPv4 or IPv6 address	String		
record_type	query type	String		A
recursion	recursion, disabled(0) or enabled(1)	Int	0 or 1	1
expect	expect	String		
Following parameters are for health monitor RADIUS method				
port	RADIUS port	Int	1 – 65534	1812
user	user name	String		a10
password	password	String		
secret	secret	String		a10
Following parameters are for health monitor LDAP method				
port	LDAP port	Int	1 – 65534	389

ssl	SSL, disabled(0) or enabled(1)	Int	0 or 1	0
distinguished_name	distinguished name	String		
password	password	String		
Following parameters are for health monitor RTSP method				
port	RTSP port	Int	1 – 65534	554
url	URL string	String		/sample.mpg
Following parameters are for health monitor SIP method				
port	SIP port	Int	1 – 65534	5060
register	register, disabled(0) or enabled(1)	Int	0 or 1	0
tcp	disabled(0) or enabled(1)	Int	0 or 1	0
Following parameters are for health monitor NTP method				
port	NTP port	Int	1 – 65534	123
Following parameters are for health monitor IMAP method				
port	IMAP port	Int	1 – 65534	143
user	user name	String		a10
password	password	String		
plain_text	plain text	Int	0 or 1	0
login	simple login	Int	0 or 1	0
cram_md5	Challenge-Response Authentication Mechanism	Int	0 or 1	0
Following parameters are for health monitor DATABASE method				
db_name	database name	String		
user	user name	String		
password	password	String		
sql	SQL query	String		
res_string	respond string	String		
row	row	Int	1 – 10	
column	column	Int	1 – 10	
Following parameters are for health monitor COMPOUND method				
compound	Reverse Polish Notation	String		
Following parameters are for health monitor EXTERNAL method				
program	program name	String		
arguments	program arguments	String		
server_port	server port	Int	1 – 65534	

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
- **HTTP Action:**
POST
- **HTTP Body:**
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.hm.update&name=hml&retry=3&consec_pass_reqd=1&interval=5&time_out=5&strictly_retry=0&disable_after_down=0&override_ipv4=192.168.3.5&override_port=8008&type=http

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> Health Monitor >> Health Monitor

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.26.5 “slb.hm.delete” Method

This method is used to delete an SLB health monitor.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.hm.delete	String		
name (*)	health monitor name	String		

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST
- HTTP Body
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.hm.delete&name=hml

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> Health Monitor >> Health Monitor

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.26.6 “slb.hm.external.getAll” Method

This method is used to get configuration information for all external health monitor scripts. For each script, this method only provides their name and description. To get the contents of the script, please use the slb.hm.external.search method.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.hm.external.getAll	String		

Note: (*) parameter is required in the API.

Response Fields

external_program_list: xml tag of all external scripts.

external_program: xml tag for external script

name: script name

description: script description

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
GET
- HTTP Body
session_id=xxxxxxxxxxxxxxxxxxxxxx&method=slb.hm.external.getAll

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
<external_program_list>
  <external_program>
    <name>external_name1</name>
    <description>external program for http</description>
  </external_program>
  <external_program>
    <name>external_name2</name>
    <description>external program for https</description>
  </external_program>
</external_program_list>
</response>
```

Menus Privilege

Config Mode >> Service >> Health Monitor >> External Program

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.26.7 “slb.hm.external.search” Method

Get the detail configuration information of a health monitor script.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.hm.external.search	String		
name(*)	health monitor external script name	String		

Note: (*) parameter is required in the API.

Response Fields

external_program: xml tag for external script
name: script name
description: script description

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
GET
- HTTP Body
session_id=xxxxxxxxxxxxxxxxxxxxxx&method=slb.hm.external.search&name=external_name1

Response as the HTTP body:

```

<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
  <external_program_list>
    <external_program>
      <name>external_name1</name>
      <description>
        TCL script
      </description>
      <definition>
        #!/bin/bash

        echo $1
        if [ "$1" = "del" ]
        then
          sed '/50.50.50.200/ d' /var/named/my.local.zone > my.txt
          cp my.txt /var/named/my.local.zone
        else
          if [ "$1" = "add" ]
          then
            sed '/50.50.50.200/ d' /var/named/my.local.zone > my.txt
            cp my.txt /var/named/my.local.zone
            sed '/40.40.40.200/ a\
www           A           50.50.50.200'
            /var/named/my.local.zone > you.txt
            cp you.txt /var/named/my.local.zone
          fi
        fi
      ]]>
    </definition>
  </external_program>
</external_program_list>
</response>

```

Menus Privilege

Config Mode >> Service >> Health Monitor >> External Program

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.26.8 “slb.hm.external.create” Method

This method is used to create an external health monitor script, including the script name, description, and definition of the contents of the script.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.hm.external.create	String		
name(*)	Health monitor external script name	String		
description	Description	String		
definition	Definition, the content of the script	String		

Note: (*) parameter is required in the API.

Response Fields

General XML response

Example Response

Request as:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
- **HTTP Action:**
POST

- **HTTP Body**
session_id=xxxxxxxxxxxxxxxxxxxxxx&method=slb.hm.external.create&name=external_name1&
description=this is a test&definition=#!/bin/bash
echo \$1
if ["\$1" = "del"]
then
sed '/50.50.50.200/ d' /var/named/my.local.zone > my.txt
cp my.txt /var/named/my.local.zone
else
if ["\$1" = "add"]
then
sed '/50.50.50.200/ d' /var/named/my.local.zone > my.txt
cp my.txt /var/named/my.local.zone
sed '/40.40.40.200/ a'
www A 50.50.50.200' /var/named/my.local.zone > you.txt
cp you.txt /var/named/my.local.zone
fi
fi

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> Health Monitor >> External Program

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.26.9 “slb.hm.external.update” Method

This method is used to update one or more of the parameters of an external health monitor script, such as the script name, script description, or script contents definition.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.hm.external.update	String		
name(*)	Health monitor external script name	String		
description	Description	String		
definition	Definition, the content of the script	String		

Note: (*) parameter is required in the API.

Response Fields

General XML response

Example Response

Request as:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
- **HTTP Action:**
POST
- **HTTP Body**
session_id=xxxxxxxxxxxxxxxxxxxxxx&method=slb.hm.external.update&name=external_name1&
description=this is a test&definition=#!/bin/bash
echo \$1

```

if [ "$1" = "del" ]
then
sed '/50.50.50.200/ d' /var/named/my.local.zone > my.txt
cp my.txt /var/named/my.local.zone
else
if [ "$1" = "add" ]
then
sed '/50.50.50.200/ d' /var/named/my.local.zone > my.txt
cp my.txt /var/named/my.local.zone
sed '/40.40.40.200/ a\'
www      A      50.50.50.200' /var/named/my.local.zone > you.txt
cp you.txt /var/named/my.local.zone
fi
fi

```

Response as the HTTP body:

```

<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>

```

Menus Privilege

Config Mode >> Service >> Health Monitor >> External Program

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.26.10 “slb.hm.external.delete” Method

This method is used to delete an external health monitor script.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.hm.external.delete	String		
name(*)	Health monitor external script name	String		

Note: (*) parameter is required in the API.

Response Fields

General XML response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST
- HTTP Body
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.hm.external.delete&name=external_name1

Response as the HTTP body:

```

<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>

```

Menus Privilege

Config Mode >> Service >> Health Monitor >> External Program

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.26.11 “slb.hm.http_post_file.getAll” Method

This method is used to get configuration information for all HTTP post files, without including the file contents.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.hm.http_post_file.getAll	String		

Note: (*) parameter is required in the API.

Response Fields

http_post_file_list	XML tag for the collection of health monitor HTTP POST files
http_post_file	XML tag for health monitor HTTP POST file
name	health monitor HTTP POST file name

Example Response

Request as:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
- **HTTP Action:**
POST
- **HTTP Body**
session_id=xxxxxxxxxxxxxxxxxxxxxx&method=slb.hm.http_post_file.getAll

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
<http_post_file_list>
    <http_post_file>
        <name>20100517008.xls</name>
    </http_post_file>
    <http_post_file>
        <name>NtpLog.txt</name>
    </http_post_file>
    <http_post_file>
        <name>aierrorlog.txt</name>
    </http_post_file>
    <http_post_file>
        <name>user01.key</name>
    </http_post_file>
</http_post_file_list>
</response>
```

Menus Privilege

Config Mode >> Service >> Health Monitor >> Health HTTP Post File

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.26.12 “slb.hm.http_post_file.delete” Method

This method is used to delete a health monitor HTTP post file.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.hm.http_post_file.delete	String		
name(*)	Health monitor HTTP POST file name	String		

Note: (*) parameter is required in the API.

Response Fields

General XML response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST
- HTTP Body
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.hm.http_post_file.delete&name=http_post1

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> Health Monitor >> Health HTTP Post File

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.26.13 “slb.hm.http_post_file.download” Method

This method is used to download an HTTP post file.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.hm.http_post_file.download	String		
name(*)	Health monitor HTTP POST file name	String		

Note: (*) parameter is required in the API.

Response Fields

HTTP CODE: 404: not found

HTTP CODE: 500: interface error

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
Multipart-GET

- **HTTP Body**
session_id=xxxxxxxxxxxxxxxxxxxxxx&method=slb.hm.http_post_file.download&name=http_post1

Response as the HTTP body:

Menus Privilege

Config Mode >> Service >> Health Monitor >> Health HTTP Post File
Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.26.14 “slb.hm.http_post_file.upload” Method

This method is used to upload an HTTP post file.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.hm.http_post_file.upload	String		

Note: (*) parameter is required in the API.

Response Fields

HTTP CODE: 404: not found

HTTP CODE: 500: interface error

Example Response

Request as:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
- **HTTP Action:**
Multipart-POST
- **HTTP Body**
session_id=xxxxxxxxxxxxxxxxxxxxxx&method=slb.hm.http_post_file.upload

Response as the HTTP body:

Menus Privilege

Config Mode >> Service >> Health Monitor >> Health HTTP Post File
Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.26.15 “slb.hm.global.get” Method

This method is used to get global health monitor information configured on the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.hm.global.get	String		

Note: (*) parameter is required in the API.

Response Fields

health_monitor_global	XML tag for the collection of health monitor global parameters
retry	health monitor retries
consec_pass_reqd	health monitor retries before declaring target up
interval	health monitor interval
timeout	health monitor timeout

Example Response

Request as:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
- **HTTP Action:**
GET
- **HTTP Body**
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.hm.global.get

Response as the HTTP body:

```
<response status="ok">
<health_monitor_global>
    <retry>3</retry>
    <consec_pass_reqd>1</consec_pass_reqd>
    <interval>5</interval>
    <timeout>5</timeout>
</health_monitor_global>
</response>
```

Menus Privilege

Config Mode >> Service >> Health Monitor >> Global

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.26.16 “slb.hm.global.set” Method

This method is used to set global health monitor information on the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.hm.global.set	String		
retry	Retry	Int	1 – 5	3
consec_pass_reqd	consec pass req'd	Int	1 – 10	1
interval	Interval	Int	1 – 180	5
timeout	Timeout	Int	1 – 12	5

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
- **HTTP Action:**
POST
- **HTTP Body**

```

session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.hm.global.set&retry=3&consecpass_req
d=1&interval=5&timeout=5
Response as the HTTP body:
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>

```

Menus Privilege

Config Mode >> Service >> Health Monitor >> Global

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.27 aFlex

7.27.1 “slb.aflex.getAll” Method

This method is used to get configuration information for all aFlex scripts configured on the AX device. You can get aFlex content (if absent) by using the slb.aflex.download method.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.aflex.getAll	String		

Note: (*) parameter is required in the API.

Response Fields

aFlex_list:	XML tag of aflex list
aFlex:	XML tag of aflex
name:	aflex name
syntax_check: 1:	syntax is correct, 0: syntax is incorrect
ref_count:	reference count

Example Response

Request as:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
- **HTTP Action:**
GET
- **HTTP Body**
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.aflex.get

Response as the HTTP body:

```

<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
  <aFlex_list>
    <aFlex>
      <name>aflex1</name>
      <syntax_check>1</syntax_check>
      <ref_count>0</ref_count>
    </aFlex>
    <aFlex>
      <name>aflex2</name>
      <syntax_check>1</syntax_check>
      <ref_count>1</ref_count>
    </aFlex>
  </aFlex_list>
</response>

```

```

<aFlex>
  <name>aFlex3</name>
  <syntax_check>1</syntax_check>
  <ref_count>3</ref_count>
</aFlex>
</aFlex_list>
</response>

```

Menus Privilege

Config Mode >> Service >> aFlex

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.27.2 “slb.aflex.search” Method

This method is used to get an aFlex script on the AX device by given name. If aFlex content is absent, you can get aFlex content using the slb.aflex.download method.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.aflex.search	String		
name(*)	aFlex name	String		

Note: (*) parameter is required in the API.

Response Fields

aFlex:	XML tag of aflex
name:	aflex name
syntax_check: 1:	syntax is correct, 0: syntax is incorrect
ref_count:	reference count

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
GET
HTTP Body
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.aflex.search&name=test

Response as the HTTP body:

```

<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
  <aFlex>
    <name>aFlex1</name>
    <syntax_check>1</syntax_check>
    <ref_count>0</ref_count>
  </aFlex>
</response>

```

Menus Privilege

Config Mode >> Service >> aFlex

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.27.3 “slb.aflex.upload” Method

This method is used to upload an aFlex file to the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.aflex.upload	String		

Note: (*) parameter is required in the API.

Response Fields

HTTP CODE: 404, not found

HTTP CODE: 500, internal err

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
Multipart-POST
- HTTP Body
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.aflex.upload

Response as the HTTP body:

Menus Privilege

Config Mode >> Service >> aFlex

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.27.4 “slb.aflex.download” Method

This method is used to download an aFlex file from the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.aflex.download	String		
name(*)	aFlex name	String		

Note: (*) parameter is required in the API.

Response Fields

HTTP CODE: 404, not found

HTTP CODE: 500, internal err

Example Response

Request as:

- URL:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
- **HTTP Action:**
Multipart-GET
- **HTTP Body:**
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.aflex.download&name=test1

Response as the HTTP body:

Menus Privilege

Config Mode >> Service >> aFlex

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.27.5 “slb.aflex.update” Method

This method is similar to the slb.aflex.upload method, except this “update” method can be used to override an existing aFlex script, while the slb.aflex.upload method cannot. Note that if no script exists to override, an error message will be generated.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.aflex.update	String		

Note: (*) parameter is required in the API.

Response Fields

HTTP CODE: 404, not found

HTTP CODE: 500, internal err

Example Response

Request as:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
- **HTTP Action:**
Multipart-POST
- **HTTP Body:**
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.aflex.update

Response as the HTTP body:

Menus Privilege

Config Mode >> Service >> aFlex

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.27.6 “slb.aflex.delete” Method

This method is used to delete an aFlex policy from the AX.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.aflex.delete	String		
name(*)	aFlex name	String		

Note: (*) parameter is required in the API.

Response Fields

General XML response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST
- HTTP Body
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.aflex.delete&name=aaa

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> aFlex

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.27.7 “slb.aflex.fetchAllStatistics” Method

This method is used to get statistics for all aFlex scrtips configured on the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.aflex.fetchAllStatistics	String		

Note: (*) parameter is required in the API.

Response Fields

General XML response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
GET
- HTTP Body
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.aflex.fetchAllStatistics

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
<aFlexStat list />
```

```
</response>
```

Menus Privilege

Monitor Mode >> Service >> aFlex

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.27.8 “slb.aflex.fetchStatistics” Method

This method is used to get statistics for an aFlex script from the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.aflex.fetchStatistics	String		
name(*)	aFlex name	String		

Note: (*) parameter is required in the API.

Response Fields

General XML response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
GET
- HTTP Body
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.aflex.fetchStatistics&name=aaa

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
  <aFlexStat_list />
</response>
```

Menus Privilege

Monitor Mode >> Service >> aFlex

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.27.9 “slb.aflex.clearAllStatistics” Method

This method is used to clear all statistics information for all aFlex scripts configured on the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.aflex.clearAllStatistics	String		

Note: (*) parameter is required in the API.

Response Fields

General XML response

Example Response

Request as:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
- **HTTP Action:**
GET
- **HTTP Body:**
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.aflex.clearAllStatistics

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
<aFlexStat_list />
</response>
```

Menus Privilege

Monitor Mode >> Service >> aFlex

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.27.10 “slb.aflex.clearStatistics” Method

This method is used to clear statistics information for a particular aFlex script from the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.aflex.clearStatistics	String		
name(*)	aFlex name	String		

Note: (*) parameter is required in the API.

Response Fields

General XML response

Example Response

Request as:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
- **HTTP Action:**
POST
- **HTTP Body:**
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.aflex.clearStatistics&name=aaa

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
<aFlexStat_list />
</response>
```

Menus Privilege

Monitor Mode >> Service >> aFlex

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.27.11 “slb.aflex.clearAllEvents” Method

This method is used to clear all events information for all aFlex scripts configured on the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.aflex.clearAllEvents	String		

Note: (*) parameter is required in the API.

Response Fields

General XML response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST
- HTTP Body
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.aflex.clearAllEvents

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
  <aFlexStat_list />
</response>
```

Menus Privilege

Monitor Mode >> Service >> aFlex

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.27.12 “slb.aflex.clearEvents” Method

This method is used to clear all aFlex events for a particular aFlex script.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.aflex.clearEvents	String		
name(*)	aFlex name	String		
event_name(*)	event name	String		

Note: (*) parameter is required in the API.

Response Fields

General XML response

Example Response

Request as:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
- **HTTP Action:**
POST
- **HTTP Body**
session_id=xxxxxxxxxxxxxxxxxxxxx&method=slb.aflex.clearEvents&name=aaa&event_name=C
LIENT_ACCEPTED

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
  <aFlexStat_list />
</response>
```

Menus Privilege

Monitor Mode >> Service >> aFlex

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.28 Class List

7.28.1 “slb.class_list.getAll” Method

This method is used to get configuration information for all class lists configured on the AX device.

Note: If the class list is created by importing a file, then the ‘entry_list’ and ‘dns_list’ will not appear in the output from this method. If you need to see the detailed output for an imported class list, please use the **slb.class_list.download** method.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.class_list.getAll	String		

Note: (*) parameter is required in the API.

Response Fields

class_list_list	XML tag for the collection of SLB class list
class_list	XML tag for SLB class list
name	class list name
location	location, 0: file, 1: config
entry_list	XML tag for collection of entry
entry	XML tag for entry
ip_address	A.B.C.D/nn for IPv4, A:B:C:D:E:F:G:H/nn for IPv6
flag	0: local; 1:global
lid_index	lid index
age	class-list entry aging time
dns_list	XML tag for collection of dns
dns	XML tag for dns
name	domain
flag	0: local; 1:global
lid_index	lid index
match_type	0: contains, 1: starts with, 2: ends with

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call

```
AX2100#show run | sec class-list
class-list cl1
2.3.3.0 /24 glid 1023
3.3.3.0 /24 lid 12
dns contains aaa lid 11
```

dns starts-with bbb glid 123

1. HTTP Request in URL Format:

session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.class_list.getAll&format=url

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
  <class_list_list>
    <class_list>
      <name>c11</name>
      <location>0</location>
      <entry_list>
        <entry>
          <ip_address>2.3.3.0/24</ip_address>
          <flag>1</flag>
          <lid_index>1023</lid_index>
          <age>20</age>
        </entry>
        <entry>
          <ip_address>3.3.3.0/24</ip_address>
          <flag>0</flag>
          <lid_index>12</lid_index>
          <age>0</age>
        </entry>
      </entry_list>
      <dns_list>
        <dns>
          <name>aaa</name>
          <flag>0</flag>
          <lid_index>11</lid_index>
          <match_type>0</match_type>
        </dns>
        <dns>
          <name>bbb</name>
          <flag>1</flag>
          <lid_index>123</lid_index>
          <match_type>1</match_type>
        </dns>
      </dns_list>
    </class_list>
  </class_list_list>
</response>
```

2. HTTP Request in JSON Format:

session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.class_list.getAll&format=json

Response as the HTTP body (JSON Format):

```
{
  "class_list_list": [
    {
      "name": "c11",
      "location": 1,
      "entry_list": [
        {
          "ip_address": "2.3.3.0/24",
          "flag": 1,
          "lid_index": 1023,
          "age": 20
        },
        {
          "ip_address": "3.3.3.0/24",
          "flag": 0,
          "lid_index": 12,
          "age": 0
        }
      ]
    }
  ]
}
```

```

        "ip_address": "3.3.3.0/24",
        "flag": 0,
        "lid_index": 12,
        "age": 0
    }
],
"dns_list": [
    {
        "name": "aaa",
        "flag": 0,
        "lid_index": 11,
        "match_type": 0
    },
    {
        "name": "bbb",
        "flag": 1,
        "lid_index": 123,
        "match_type": 1
    }
]
}
}

```

Menus Privilege

Config Mode >> Service >> SLB >> Class List

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.28.2 “slb.class_list.search” Method

This method is used to search all class list information configured on the AX device.

Note: If the class list is created by importing a file, then the ‘entry_list’ and ‘dns_list’ will not appear in the output from this method. If you need to see the detailed output for an imported class list, please use the **slb.class_list.download** method.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.class_list.getAll	String		
name (*)	class list name	String		

Note: (*) parameter is required in the API.

Response Fields

class_list	XML tag for SLB class list
name	class list name
location	location, 0: file, 1: config
entry_list	XML tag for collection of entry
entry	XML tag for entry

ip_address	A.B.C.D/nn for IPv4, A:B:C:D:E:F:G:H/nn for IPv6
flag	0: local; 1:global
lid_index	lid index
age	class-list entry aging time
dns_list	XML tag for collection of dns
dns	XML tag for DNS
name	domain
flag	0: local; 1:global
lid_index	lid index
match_type	0: contains, 1: starts with, 2: ends with

Example Response

Request as:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
- **HTTP Action:**
GET

CLI configuration before aXAPI call

```
AX2100#show run | sec class-list
class-list cl1
2.3.3.0 /24 glid 1023
3.3.3.0 /24 lid 12
dns contains aaa lid 11
dns starts-with bbb glid 123
```

1. HTTP Request in URL Format:

```
session_id=xxxxxxxxxxxxxxxxxxxxx&method=slb.class_list.search&name=cl1&format=url
```

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
    <class_list>
        <name>cl1</name>
        <location>0</location>
        <entry_list>
            <entry>
                <ip_address>2.3.3.0/24</ip_address>
                <flag>1</flag>
                <lid_index>1023</lid_index>
                <age>200</age>
            </entry>
            <entry>
                <ip_address>3.3.3.0/24</ip_address>
                <flag>0</flag>
                <lid_index>12</lid_index>
                <age>0</age>
            </entry>
        </entry_list>
        <dns_list>
            <dns>
                <name>aaa</name>
                <lid_index>11</lid_index>
                <match_type>0</match_type>
                <flag>0</flag>
            </dns>
        </dns_list>
    </class_list>
</response>
```

```

        <dns>
            <domain>bbb</domain>
            <lid_index>123</lid_index>
            <match_type>1</match_type>
            <flag>1</flag>
        </dns>
    </dns_list>
</class_list>
</response>

```

2. HTTP Request in JSON Format:

session_id=xxxxxxxxxxxxxxxxxxxxx&method=slb.class_list.search&name=c11&format=json

Response as the HTTP body (JSON Format):

```

{
    "class_list": {
        "name": "c11",
        "location": 1,
        "entry_list": [
            {
                "ip_address": "2.3.3.0/24",
                "flag": 1,
                "lid_index": 1023,
                "age": 20
            },
            {
                "ip_address": "3.3.3.0/24",
                "flag": 0,
                "lid_index": 12,
                "age": 0
            }
        ],
        "dns_list": [
            {
                "name": "aaa",
                "flag": 0,
                "lid_index": 11,
                "match_type": 0
            },
            {
                "name": "bbb",
                "flag": 1,
                "lid_index": 123,
                "match_type": 1
            }
        ]
    }
}

```

Menus Privilege

Config Mode >> Service >> SLB >> Class List

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.28.3 “slb.class_list.upload” Method

Create a class list by uploading a class-list file.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.class_list.upload	String		

Note: (*) parameter is required in the API.

Response Fields

No

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
Multi-POST (HTTP upload)

1. HTTP Request in URL Format:

session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.class_list.upload&format=url

Response as the HTTP body (URL Format):

NONE

2. HTTP Request in JSON Format:

session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.class_list.upload&format=json

Response as the HTTP body (JSON Format):

NONE

Menus Privilege

Config Mode >> Service >> SLB >> Class List

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

- | | |
|---|---|
| 3602: Internal I/O err. | (HTTP response code: 500 Internal server err) |
| 3603: File with that name already exist. | (HTTP response code: 500 Internal server err) |
| 3604: HTTP Multi-POST, file name not set. | (HTTP response code: 500 Internal server err) |

7.28.4 “slb.class_list.download” Method

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.class_list.getAll	String		
file_name (*)	class list name	String		

Note: (*) parameter is required in the API.

Response Fields

No

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
Multi-GET (HTTP download)

1. HTTP Request in URL Format:

```
session_id=xxxxxxxxxxxxxxxxxxxxxx&method=slb.class_list.download&file_name=clistFile01&format=url
```

Response as the HTTP body (URL Format):

NONE

2. HTTP Request in JSON Format:

```
session_id=xxxxxxxxxxxxxxxxxxxxxx&method=slb.class_list.download&file_name=clistFile01&format=json
```

Response as the HTTP body (JSON Format):

NONE

Menus Privilege

Config Mode >> Service >> SLB >> Class List

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

3601:	No such file to download.	(HTTP response code: 404 not found)
3602:	Internal I/O err.	(HTTP response code: 500 Internal server err)
3603:	File with that name already exist.	(HTTP response code: 500 Internal server err)
3604:	HTTP Multi-POST, file name not set.	(HTTP response code: 500 Internal server err)

7.28.5 “slb.class_list.create” Method

This method is used to create a class list.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.class_list.create	String		
name (*)	class list name	String		
location	location, 0: file, 1: config	Int	0 or 1	1
entry_list	class list entry list			
entry<n> ip_address flag	header erase request at element <n> A.B.C.D/n for IPv4, A:B:C:D:E:F:G:H/n for IPv6 flag, 0 local, 1 global	String Int	0 or 1	

lid_index	lid index	Int	1 – 31 (local) 1 – 1023 (global) 0 – 2000	0 0 0
age	class-list entry aging time in minutes	Int		
dns_list				
dns<n>	header erase request at element <n>	String		
name (*)	domain	int	0, 1 or 2	0
match_type	0 contains, 1 starts with, 2 ends with	int	0 or 1	0
flag	flag, 0 local, 1 global	Int	1 – 31 (local)	0
lid_index	lid index		1 – 1023 (global)	0

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
- **HTTP Action:**
POST

CLI configuration before aXAPI call

```
AX2100#show run | sec class-list
```

1. HTTP Request in URL Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.class_list.create&class_list=name%03clist_na
me1%02location%031%02entry_list%02dns_list&entry_list=entry1&entry1=ip_address%033.3.3.
6/24%02flag%031%02lid_index%0315&dns_list=dns1%02dns2&dns1=name%03abcd%02match_type%031
%02flag%030%02lid_index%0312&dns2=name%03ffff%02match_type%032%02flag%031%02lid_index%0
3131&format=url
```

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in JSON Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.class_list.create&format=json
```

Response as the HTTP body (JSON Format):

```
{
  "class_list": {
    "name": "cl2",
    "location": 1,
    "entry_list": [
      {
        "ip_address": "2.3.3.0/24",
        "flag": 0,
        "lid_index": 1,
        "age": 0
      },
      {
        "ip_address": "3.3.3.0/24",
        "flag": 0,
        "lid_index": 12,
        "age": 0
      }
    ]
  }
}
```

```

        "age": 0
    },
],
"dns_list": [
{
    "name": "aaa",
    "flag": 0,
    "lid_index": 11,
    "match_type": 0
},
{
    "name": "bbb",
    "flag": 0,
    "match_type": 0
}
]
}
}

```

CLI configuration after aXAPI call

```

AX2100#show run | sec class-list
class-list cl2
2.3.3.0 /24 lid 1
3.3.3.0 /24 lid 12
dns contains aaa lid 11
dns contains bbb lid 1

```

Menus Privilege

Config Mode >> Service >> SLB >> Class List

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.28.6 “slb.class_list.update” Method

This method is used to update one or more parameters of a class list.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.class_list.update	String		
name (*)	class list name	String		
location	location, 0: file, 1: config	Int	0 or 1	1
entry_list	class list entry list			
entry<n>	header erase request at element <n>			
ip_address	A.B.C.D/nn for IPv4, A:B:C:D:E:F:G:H/nn for IPv6	String		
flag	flag, 0 local, 1 global	Int	0 or 1	0
lid_index	lid index	Int	1 – 31 (local) 1 – 1023 (global)	0
age	class-list entry aging time in minutes	Int	0 – 2000	0
dns_list				
dns<n>	header erase request at element <n>			
name (*)	domain	String		
match_type	0 contains, 1 starts with, 2 ends with	int	0, 1 or 2	0
flag	flag, 0 local, 1 global	int	0 or 1	0

lid_index	lid index	Int	1 – 31 (local) 1 – 1023 (global)	0
------------------	-----------	-----	-------------------------------------	---

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

1. HTTP Request in URL Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.class_list.update&class_list=name%03cli
st_name1%02location%031%02entry_list%02dns_list&entry_list=entry1&entry1=ip_addres
s%033.3.3.6/24%02flag%031%02lid_index%0315&dns_list=dns1%02dns2&dns1=name%03abcd%0
2match_type%031%02flag%030%02lid_index%0312&dns2=name%03ffff%02match_type%032%02fl
ag%031%02lid_index%03131&format=url
```

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in JSON Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.class_list.update&format=json
```

Response as the HTTP body (JSON Format):

```
{
  "class_list": {
    "name": "cl1",
    "location": 1,
    "entry_list": [
      {
        "ip_address": "2.3.3.0/24",
        "flag": 0,
        "lid_index": 1,
        "age": 0
      },
      {
        "ip_address": "3.3.3.0/24",
        "flag": 0,
        "lid_index": 12,
        "age": 0
      }
    ],
    "dns_list": [
      {
        "name": "aaa",
        "flag": 0,
        "lid_index": 11,
        "match_type": 0
      },
      {
        "name": "bbb",
        "flag": 0,
        "match_type": 0
      }
    ]
  }
}
```

```

        }
    ]
}

```

Menus Privilege

Config Mode >> Service >> SLB >> Class List

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.28.7 “slb.class_list.delete” Method

This method is used to delete a class list for IP limiting or DNS caching.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.class_list.delete	String		
name (*)	class list name	String		

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call

```

AX2100#show run | sec class-list
class-list cl2
2.3.3.0 /24 lid 1
3.3.3.0 /24 lid 12
dns contains aaa lid 11
dns contains bbb lid 1

```

1. HTTP Request in URL Format:

```

1      session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.class_list.delete&name=clist1&format=ur

```

Response as the HTTP body (URL Format):

```

<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>

```

2. HTTP Request in JSON Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.class_list.delete&format=json
```

Response as the HTTP body (JSON Format):

```
{"response": {"status": "OK"}}
```

CLI configuration after aXAPI call

```
AX2100#show run | sec class-list
```

Menus Privilege

Config Mode >> Service >> SLB >> Class List

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.28.8 “slb.class_list.entry.create” Method

This method is used to add an entry to a class list.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.class_list.entry.create	String		
name (*)	class list name	String		
entry<n> ip_address flag lid_index age	Header erase request at element <n> A.B.C.D/n for IPv4, A:B:C:D:E:F:G:H/n for IPv6 flag, 0 local, 1 global lid index class-list entry aging time in minutes	String Int Int Int	0 or 1 1 – 31 (local) 1 – 1023 (global) 0 - 2000	0 0

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
AX2100#show run | sec class-list
class-list cl1
2.3.3.0 /24 glid 1023
3.3.3.0 /24 lid 12
dns contains aaa lid 11
dns starts-with bbb glid 123
```

1. HTTP Request in URL Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.class_list.entry.create&name=clist_name
1&entry_list=entry1&entry1=ip_address%033.3.3.6/24%02flag%031%02lid_index%0315&for
mat=url
```

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in JSON Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.class_list.entry.create&format=json
```

Response as the HTTP body (JSON Format):

```
{
  "name": "cl1",
  "entry_list": [
    {
      "ip_address": "2.3.9.0/24",
      "flag": 1,
      "lid_index": 15
    }
  ]
}
```

CLI configuration after aXAPI call

```
AX2100#show run | sec class-list
class-list cl1
2.3.3.0 /24 glid 15
2.3.9.0 /24 glid 15
3.3.3.0 /24 lid 12
dns contains aaa lid 11
dns starts-with bbb glid 123
```

Menus Privilege

Config Mode >> Service >> SLB >> Class List

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.28.9 “slb.class_list.entry.update” Method

This method is used to update one or more of parameters in a class list entry.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.class_list.entry.update	String		
name (*)	class list name	String		
entry<n>	Header erase request at element <n>			

ip_adress	A.B.C.D/nn for IPv4, A:B:C:D:E:F:G:H/nn for IPv6	String		
flag	flag, 0 local, 1 global	Int	0 or 1 1 – 31 (local) 1 – 1023 (global)	0
lid_index	lid index	Int		
age	class-list entry aging time in minutes	Int	0 – 2000	0

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- **URL:**
[https://\[AX_IP_ADDRESS\]/services/rest/V2/](https://[AX_IP_ADDRESS]/services/rest/V2/)
- **HTTP Action:**
POST

CLI configuration before aXAPI call:

```
AX2100#show run | sec class-list
class-list cl1
2.3.3.0 /24 glid 1023
3.3.3.0 /24 lid 12
dns contains aaa lid 11
dns starts-with bbb glid 123
```

1. HTTP Request in URL Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.class_list.entry.update&name=clist_name
1&entry_list=entry1&entry1=ip_address%033.3.3.6/24%02flag%031%02lid_index%0315&for
mat=url
```

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in JSON Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.class_list.entry.update&format=json
```

Response as the HTTP body (JSON Format):

```
{
  "name": "cl1",
  "entry_list": [
    {
      "ip_address": "3.3.3.6/24",
      "flag": 0,
      "lid_index": 3,
      "age": 0
    }
  ]
}
```

CLI configuration after aXAPI call (JSON Format):

```

AX2100#show run | sec class-list
class-list cl1
2.3.3.0 /24 glid 10
3.3.3.0 /24 glid 15
dns contains aaa lid 11
dns starts-with bbb glid 123

```

Menus Privilege

Config Mode >> Service >> SLB >> Class List

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.28.10 “slb.class_list.entry.delete” Method

This method is used to delete a class list entry.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.class_list.entry.delete	String		
name (*)	class list name	String		
entry<n> ip_address	Header erase request at element <n> A.B.C.D/n for IPv4, A:B:C:D:E:F:G:H/n for IPv6	String		

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
[https://\[AX_IP_ADDRESS\]/services/rest/V2/](https://[AX_IP_ADDRESS]/services/rest/V2/)
- HTTP Action:
POST

CLI configuration before aXAPI call:

```

AX2100#show run | sec class-list
class-list cl1
2.3.3.0 /24 glid 10
3.3.3.0 /24 glid 10
3.3.7.0 /24 glid 7
3.3.8.0 /24 glid 8
dns contains aaa lid 11
dns starts-with bbb glid 123

```

1. HTTP Request in URL Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.class_list.entry.delete&name=clist_name1&entry_list=entry1&entry1=ip_address%033.3.3.6/24%02flag%031%02lid_index%0315&format=url
```

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in JSON Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.class_list.entry.delete&format=json
```

Response as the HTTP body (JSON Format):

```
{
  "name": "cl1",
  "entry_list": [
    {
      "ip_address": "3.3.8.0/24"
    },
    {
      "ip_address": "3.3.7.0/24"
    }
  ]
}
```

CLI after aXAPI call (JSON Format):

```
AX2100#show run | sec class-list
class-list cl1
2.3.3.0 /24 glid 10
3.3.3.0 /24 glid 10
dns contains aaa lid 11
dns starts-with bbb glid 123
```

Menus Privilege

Config Mode >> Service >> SLB >> Class List

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.28.11 “slb.class_list.dns.create” Method

This method is used to create a DNS class list.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.class_list.dns.create	String		
name (*)	class list name	String		
dns<n> name	Header erase request at element <n> domain	String		

match_type	0 contains, 1 starts with, 2 ends with	Int	0, 1 or 2	0
flag	flag, 0 local, 1 global	Int	0 or 1	0
lid_index	lid index	int	1 – 31 (local) 1 – 1023 (global)	0

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
- **HTTP Action:**
POST

CLI configuration before aXAPI call:

```
AX2100#show run | sec class-list
class-list cl1
2.3.3.0 /24 glid 10
3.3.3.0 /24 glid 10
dns contains aaa lid 11
dns starts-with bbb glid 123
```

1. HTTP Request in URL Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.class_list.dns.create&name=clist_name1&dns_
list=dns1%02dns2&dns1=name%03abcd%02match_type%031%02flag%030%02lid_index%0312&dns2=na
me%03ffff%02match_type%032%02flag%031%02lid_index%03131&format=url
```

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in JSON Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.class_list.dns.create&format=json
```

Response as the HTTP body (JSON Format):

```
{
  "name": "cl1",
  "dns_list": [
    {
      "name": "abcd",
      "flag": 0,
      "lid_index": 12,
      "match_type": 1
    },
    {
      "name": "ffff",
      "flag": 1,
      "lid_index": 131,
      "match_type": 2
    }
  ]
}
```

CLI configuration after aXAPI call:

```
AX2100#show run | sec class-list
class-list cl1
2.3.3.0 /24 glid 10
3.3.3.0 /24 glid 10
dns ends-with ffff glid 131
dns starts-with abcd lid 12
dns contains aaa lid 11
dns starts-with bbb glid 123
```

Menus Privilege

Config Mode >> Service >> SLB >> Class List

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.28.12 “slb.class_list.dns.update” Method

This method is used to update one or more parameters of a DNS class list.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.class_list.dns.update	String		
name (*)	class list name	String		
dns<n>	Header erase request at element <n>	String		
name	domain	Int	0, 1 or 2	0
match_type	0 contains, 1 starts with, 2 ends with	Int	0 or 1	0
flag	flag, 0 local, 1 global	int	1 – 31 (local) 1 – 1023 (global)	0
lid_index	lid index			

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call

```
AX2100#show run | sec class-list
class-list cl1
2.3.3.0 /24 glid 10
3.3.3.0 /24 glid 10
dns ends-with ffff glid 131
dns starts-with abcd lid 12
dns contains aaa lid 11
dns starts-with bbb glid 123
```

1. HTTP Request in URL Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.class_list.dns.update&name=c1&dns_
list=dns1%02dns2&dns1=name%03abcd%02match_type%031%02flag%030%02lid_index%0312&dns2=na
me%03ffff%02match_type%032%02flag%031%02lid_index%03131&format=url
```

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in JSON Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.class_list.dns.update&format=json
```

Response as the HTTP body (JSON Format):

```
{
  "name": "c1",
  "dns_list": [
    {
      "name": "abcd",
      "flag": 1,
      "lid_index": 10,
      "match_type": 1
    },
    {
      "name": "gggg",
      "flag": 1,
      "lid_index": 131,
      "match_type": 2
    }
  ]
}
```

CLI configuration after aXAPI call (JSON Format):

```
AX2100#show run | sec class-list
class-list c1
2.3.3.0 /24 glid 10
3.3.3.0 /24 glid 10
dns ends-with gggg glid 131
dns ends-with ffff glid 131
dns starts-with abcd glid 10
dns contains aaa lid 11
dns starts-with bbb glid 123
```

Menus Privilege

Config Mode >> Service >> SLB >> Class List

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.28.13 “slb.class_list.dns.delete” Method

This method is used to delete a DNS class list.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.class_list.dns.delete	String		
name (*)	class list name	String		
dns<n> name match_type flag lid_index	Header erase request at element <n> domain 0 contains, 1 starts with, 2 ends with flag, 0 local, 1 global lid index	String Int Int int	0, 1 or 2 0 or 1 1 – 31 (local) 1 – 1023 (global)	0 0 0

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
AX2100#show run | sec class-list
class-list cl1
2.3.3.0 /24 glid 10
3.3.3.0 /24 glid 10
dns ends-with gggg glid 131
dns ends-with ffff glid 131
dns starts-with abcd glid 10
dns contains aaa lid 11
dns starts-with bbb glid 123
```

1. HTTP Request in URL Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.class_list.dns.delete&name=clist_name1&dns_
list=dns1%02dns2&dns1=name%03abcd%02match_type%031%02flag%030%02lid_index%0312&dns2=na
me%03ffff%02match_type%032%02flag%031%02lid_index%03131&format=url
```

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in JSON Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.class_list.dns.delete&format=json
```

Response as the HTTP body (JSON Format):

```
{
  "name": "cl1",
```

```

    "dns_list": [
      {
        "name": "ffff"
      },
      {
        "name": "gggg"
      }
    ]
}

```

CLI configuration after aXAPI call (JSON Format):

```

AX2100#show run | sec class-list
class-list cl1
2.3.3.0 /24 glid 10
3.3.3.0 /24 glid 10
dns starts-with abcd glid 10
dns contains aaa lid 11
dns starts-with bbb glid 123

```

Menus Privilege

Config Mode >> Service >> SLB >> Class List

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.29 GLID List

7.29.1 “slb.glid.getAll” Method

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.glid.getAll	String		

Note: (*) parameter is required in the API.

Response Fields

glid_list	XML tag for the collection of SLB GLID list
glid	XML tag for SLB GLID
id	GLID id
conn_limit	connection limit
conn_rate_limit	connection rate limit
conn_rate_limit_per	connection rate limit per (100 ms)
req_limit	request limit
req_rate_limit	request rate limit
req_rate_limit_per	request rate limit per (100 ms)
over_limit	0: drop; 1: reset; 2: forward; 3: enable dns cache; 4: disable dns cache;
over_limit_lockout	over limit lockout
over_limit_log	0 menas not set

<code>dns_cache</code>	0: disabled; 1: enabled
<code>ttl</code>	
<code>weight</code>	
<code>source_nat_pool</code>	ipv4 pool or pool group

Example Response

Request as:

- URL:
`https://[AX_IP_ADDRESS]/services/rest/V2/`
- HTTP Action:
GET
- HTTP Body
`session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.glid.getAll`

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
    <glid_list>
        <glid>
            <id>1</id>
            <conn_limit>100</conn_limit>
            <conn_rate_limit>200</conn_rate_limit>
            <conn_rate_limit_per>300</conn_rate_limit_per>
            <req_limit>400</req_limit>
            <req_rate_limit>500</req_rate_limit>
            <req_rate_limit_per>600</req_rate_limit_per>
            <over_limit>0</over_limit>
            <over_limit_lockout>0</over_limit_lockout>
            <over_limit_log>0</over_limit_log>
            <dns_cache>0</dns_cache>
            <ttl>300</ttl>
            <weight>1</weight>
            <source_nat_pool>pool_group_1</source_nat_pool>
        </glid>
        <glid>
            <id>2</id>
            <conn_limit>100</conn_limit>
            <conn_rate_limit>200</conn_rate_limit>
            <conn_rate_limit_per>300</conn_rate_limit_per>
            <req_limit>400</req_limit>
            <req_rate_limit>500</req_rate_limit>
            <req_rate_limit_per>600</req_rate_limit_per>
            <over_limit>0</over_limit>
            <over_limit_lockout>0</over_limit_lockout>
            <over_limit_log>0</over_limit_log>
            <dns_cache>0</dns_cache>
            <ttl>300</ttl>
            <weight>1</weight>
            <source_nat_pool>pool_group_1</source_nat_pool>
        </glid>
    </glid_list>
</response>
```

Menus Privilege

Config Mode >> Service >> SLB >> GLID

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.29.2 “slb.class_list.search” Method

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.glid.getAll	String		
id (*)	GLID id	Int		

Note: (*) parameter is required in the API.

Response Fields

glid	XML tag for SLB GLID
id	GLID id
conn_limit	connection limit
conn_rate_limit	connection rate limit
conn_rate_limit_per	connection rate limit per (100 ms)
req_limit	request limit
req_rate_limit	request rate limit
req_rate_limit_per	request rate limit per (100 ms)
over_limit	0: drop; 1: reset; 2: forward; 3: enable dns cache; 4: disable dns cache;
over_limit_lockout	over limit lockout
over_limit_log	0 means not set
dns_cache	0: disabled; 1: enabled
ttl	
weight	
source_nat_pool	ipv4 pool or pool group

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
GET
- HTTP Body
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.glid.search&id=1

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
    <glid>
        <id>1</id>
        <conn_limit>100</conn_limit>
        <conn_rate_limit>200</conn_rate_limit>
        <conn_rate_limit_per>300</conn_rate_limit_per>
        <req_limit>400</req_limit>
        <req_rate_limit>500</req_rate_limit>
        <req_rate_limit_per>600</req_rate_limit_per>
        <over_limit>0</over_limit>
        <over_limit_lockout>0</over_limit_lockout>
        <over_limit_log>0</over_limit_log>
        <dns_cache>0</dns_cache>
        <ttl>300</ttl>
        <weight>1</weight>
        <source_nat_pool>pool_group_1</source_nat_pool>
    </glid>
</response>
```

Menus Privilege

Config Mode >> Service >> SLB >> GLID

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.29.3 “slb.glid.create” Method

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.glid.create	String		
id (*)	GLID id	Int		
conn_limit	connection limit	Int	0 - 1048575	0
conn_rate_limit	connection rate limit	Int	0 - 2147483647	0
conn_rate_limit_per	connection rate limit per	Int	0 - 65535	0
req_limit	request limit	Int	0 - 1048575	0
req_rate_limit	request rate limit	Int	0 - 4294967295	0
req_rate_limit_per	request rate limit per	Int	0 - 65535	0
over_limit	0: drop; 1: reset; 2: forward; 3: enable dns cache; 4: disable dns cache	Int	0 - 4	0
over_limit_lockout	over limit lockout	Int	0 - 1023	0
over_limit_log	over limit log	Int	0 - 255	0
dns_cache	0: disabled; 1: enabled	Int	0, 1	0
ttl	ttl	Int	0 - 65535	300
weight	weight	Int	0 - 7	1
source_nat_pool	Ipv4 pool or pool group	String		

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
`https://[AX_IP_ADDRESS]/services/rest/V2/`
- HTTP Action:
`POST`
- HTTP Body
`session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.glid.create&glid=id%031%02conn_limit%031%02conn_rate_limit%03200%02req_limit%03300%02over_limit%031%02dns_cache%031%02ttl%03500%02weight%031%02source_nat_pool%03pool_group_1`

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> SLB >> GLID

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.29.4 “slb.glid.update” Method

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.glid.update	String		
id (*)	GLID id	Int		
conn_limit	connection limit	Int	0 - 1048575	0
conn_rate_limit	connection rate limit	Int	0 - 2147483647	0
conn_rate_limit_per	connection rate limit per	Int	0 - 65535	0
req_limit	request limit	Int	0 - 1048575	0
req_rate_limit	request rate limit	Int	0 - 4294967295	0
req_rate_limit_per	request rate limit per	Int	0 - 65535	0
over_limit	0: drop; 1: reset; 2: forward; 3: enable dns cache; 4: disable dns cache	Int	0 - 5	0
over_limit_lockout	over limit lockout	Int	0 – 1023	0
over_limit_log	over limit log	Int	0 - 255	0
dns_cache	0: disabled; 1: enabled	Int	0, 1	0
ttl	ttl	Int	0 - 65535	300
weight	weight	Int	0 - 7	1
source_nat_pool	Ipv4 pool or pool group	String		

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST
- HTTP Body
● session_id=xxxxxxxxxxxxxxxxxxxxx&method=slb.glid.update&glid=id%031%02conn_limit%03100%02conn_rate_limit%03200%02req_limit%03300%02over_limit%031%02dns_cache%031%02ttl%03500%02weight%031%02source_nat_pool%03pool_group_1

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> SLB >> GLID

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.29.5 “slb.glid.delete” Method

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.glid.delete	String		
id (*)	GLID id	Int		

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST
- HTTP Body
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.delete&id=1

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> SLB >> GLID

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.30 Global

7.30.1 “`slb.global.settings.get`” Method

This method is used to get global SLB settings configured on the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	<code>slb.global.settings.get</code>	String		

Note: (*) parameter is required in the API.

Response Fields

slb_global_settings	XML tag for the collection of the slb global settings.
dsr_health_check	DSR health check, enabled(1) or disabled (0).
graceful_shutdown	
status	Status of graceful shutdown, enabled(1) or disabled(0).
time	graceful shutdown time, only when graceful shutdown is enabled.
after_disable	Whether graceful shutdown the server after the server is disabled.
apply	server only(1) , virtual server only (2) or both (0).
syn_cookie	
status	syn cookie
syn_cookie_on	status of syn cookie, enabled(1) or disabled(0)
syn_cookie_off	syn cookie on
max_session_life	syn cookie off
stats_data	max session life.
l7_request_accounting	enabled statistics data(1) or disable statistics data(0).
	Layer 7 request accounting, enabled(1) or disabled(0)

fast_path_process	Fast path processing, enabled(1) or disabled(0)
compression_block_size	Compression block size.
aFlex_max_file_size	Max aFlex file size.
dns_cache	DNS cache
status	status of DNS cache
dns_cache_option	disabled(0), Round Robin(1) or Single answer (2)
ttl	TTL of DNS cache
dns_cache_age	DNS cache age, only when ‘DNS cache’ is enabled.
snat_against_vip	source NAT against virtual IP. Enabled (1) or disabled (0)
extended_stats	extended status, enabled(1) or disabled(0)
system_glid	system GLID

Example Response

Request as:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
- **HTTP Action:**
GET
- **HTTP Body**
session_id=xxxxxxxxxxxxxxxxxxxxxx&method=slb.global.settings.get

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
<slb_global_setting>
    <dsr_health_check>1</dsr_health_check>
    <graceful_shutdown>
        <status>1</status>
        <time>100</time>
        <after_disable>0</after_disable>
        <apply>1</apply>
    </graceful_shutdown>
    <max_session_life>40</max_session_life>
    <stats_data>1</stats_data>
    <l7_request_accounting>0</l7_request_accounting>
    <fast_path_process>1</fast_path_process>
    <compression_block_size>6000</compression_block_size>
    <aFlex_max_file_size>128</aFlex_max_file_size>
    <dns_cache>
        <status>1</status>
        <dns_cache_option>2</dns_cache_option>
        <ttl>100</ttl>
    </dns_cache>
    <dns_cache_age>1000</dns_cache_age>
    <snat_against_vip>1</snat_against_vip>
    <extended_stats>1</extended_stats>
    <system_glid>0</system_glid>
</slb_global_setting>
</response>
```

Menus Privilege

Config Mode >> Service >> SLB >> Global >> Settings

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.30.2 “slb.global.settings.set” Method

This method is used to set global settings, such as DSR health checks, graceful shutdown, and maximum session life configured on the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.global.settings.set	String		
slb_global_settings	the object name	Object		
dsr_health_check	DSR health check, enabled(1) or disabled (0).	Bool		0
graceful_shutdown	graceful shutdown object.	Object		
status	status of graceful shutdown, enabled(1) or disabled(0).	Bool		0
time	graceful shutdown time, only when graceful shutdown is enabled.	Int	1 - 65535	0
after_disable	whether graceful shutdown the server after the server is disabled.	Bool		0
apply	what the graceful shutdown applies to, server only(1) , virtual server only (2) or both (0).	Int	0 - 2	0
syn_cookie	syn cookie object	Object		
status	status of syn cookie, enabled(1) or disabled(0)	Bool		0
syn_cookie_on	syn cookie off	Int	0-2147483647	0
syn_cookie_off	syn cookie off	Int	0-2147483647	0
max_session_life	max session life	Int	1 - 40	2
stats_data	enabled statistics data(1) or disable statistics data(0).	Bool		1
l7_request_accounting	layer 7 request accounting, enabled(1) or disabled(0)	Bool		0
fast_path_process	fast path processing, enabled(1) or disabled(0)	Bool		1
compression_block_size	compression block size, 0 means disabled	Int	6000 – 32000 or 0	0
aflex_max_file_size	max aFlex file size (KB)	Int	16 - 128	32
dns_cache	DNS cache	Object		
status	status of DNS cache	Bool		
dns_cache_option	DNS cache option, disabled(0), Round Robin(1) or Single answer (2)	Int	0 - 2	0
ttl	TTL of DNS cache	Int	1 - 1000000	300
dns_cache_age	DNS cache age, only when ‘DNS cache’ is enabled.	Int	1 - 1000000	0
snat_against_vip	source NAT against virtual IP. Enabled (1) or disabled (0)	Bool		0
extended_stats	extended status, enabled(1) or disabled(0)	Bool		0
system_glid	system GLID	Int	1 – 1023	0

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST
- HTTP Body

```
session_id=123456789&method=slb.global.settings.set&slb_global_setting=dsr_health_check%031%02max_session_life%0320%02stats_data%031%02l7_request_accounting%030%02fast_path_process%031%02compression_block_size%0332000%02aflex_max_file_size%03127%02dns_cache%02dns_cache_age%0310000%02snat_against_vip%031%02extended_stats%031&gr
```

```
aceful_shutdown=status%031%02time%032000%02apply%032&dns_cache=status%031%02ttl%03  
100
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>  
<response status="ok">  
</response>
```

Menus Privilege

Config Mode >> Service >> SLB >> Global >> Settings

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.30.3 “slb.global.ddos_protection.get” Method

This method is used to get global DDoS protection settings, such as out of sequence packets, or dropping certain types of packets.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.global.get	String		

Note: (*) parameter is required in the API.

Response Fields

ddos_protection	XML tag of DDOS protection
ddos_protection	XML tag of DDOS protection
drop_all	drop all attacking packets
ip_option	drop ip option packets
land_attack	drop land attack packets
ping_of_death	drop ping of death attack packets
frag	drop all fragmented packets
tcp_no_flags	drop TCP no flag attack packets
tcp_syn_fin	drop syn-fin attack packets
tcp_syn_frag	drop fragmented syn attack packets
out_of_sequence	Out of Sequence
zero_window	Zero Window
bad_content	bad content

Example Response

Request as:

- **URL:**
`https://[AX_IP_ADDRESS]/services/rest/V2/`
- **HTTP Action:**
`GET`
- **HTTP Body**
`session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.global.ddos_protection.get`

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>  
<response status="ok">  
<ddos_protection>  
<ddos_protection>
```

```

<drop_all>0</drop_all>
<ip_option>0</ip_option>
<land_attack>0</land_attack>
<ping_of_death>0</ping_of_death>
<frag>0</frag>
<tcp_no_flags>0</tcp_no_flags>
<tcp_syn_fin>0</tcp_syn_fin>
<tcp_syn_frag>0</tcp_syn_frag>
</ddos_protection>
<out_of_sequence>1</out_of_sequence>
<zero_window>1</zero_window>
<bad_content>1</bad_content>
</ddos_protection>
</response>

```

Menus Privilege

Config Mode >> Service >> SLB >> Global >> DDos Protection

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.30.4 “slb.global.ddos_protection.set” Method

This method is used to set global DDoS protection settings, such as out of sequence packets, or dropping certain types of packets.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	User authenticated session id	String		
method (*)	slb.global.ddos_protection.set	String		
ddos_protection	Ddos protection object	Object		
ddos_protection	Ddos protection options object	Object		
drop_all	drop all attacking packets	Bool		
ip_option	drop ip option packets	Bool		
land_attack	drop land attack packets	Bool		
ping_of_death	drop ping of death attack packets	Bool		
frag	drop all fragmented packets	Bool		
tcp_no_flags	drop TCP no flag attack packets	Bool		
tcp_syn_fin	drop syn-fin attack packets	Bool		
tcp_syn_frag	drop fragmented syn attack packets	Bool		
out_of_sequence	Out of Sequence	Int	1 - 127	1
zero_window	Zero Window	Int	1 - 127	1
bad_content	bad content	Int	1 - 127	1

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST
- HTTP Body
session_id=123456789&method=slb.global.ddos_protection.set&ddos_protection=ddos_protection%02out_of_sequence%03127%02zero_window%03127%02bad_content%03127&ddos_protection=ip_option%031%02ping_of_death%031

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> SLB >> Global >> DDos Protection

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.30.5 “slb.global.log_rate_limiting.get” Method

This method is used to get global log rate limiting settings, such as max limits on local or remote logging.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.global.log_rate_limiting.get	String		

Note: (*) parameter is required in the API.

Response Fields

log_rate_limiting

max_local_logging	max local logging
max_remote_logging	max remote logging
excluding	exclude none(0), local(1) or remote(2)

Example Response

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
GET
- HTTP Body
session_id=123456789&method=slb.global.log_rate_limiting.get

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
<log_rate_limiting>
  <max_local_logging>100</max_local_logging>
  <max_remote_logging>10000</max_remote_logging>
  <excluding>0</excluding>
</log_rate_limiting>
</response>
```

Menus Privilege

Config Mode >> Service >> SLB >> Global >> Log Rate Limiting

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.30.6 “slb.global.log_rate_limiting.set” Method

This method is used to set global log rate limiting settings, such as max limits on local or remote logging.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.global.log_rate_limiting.set	String		
log_rate_limiting	log rate limiting object	Object		
max_local_logging	max local logging	Int	1 – 100	32
max_remote_logging	max remote logging	Int	1 – 100000	15000
excluding	exclude none(0), local(1) or remote(2)	Int	0 - 2	0

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
POST
- HTTP Body
session_id=123456789&method=slb.global.log_rate_limiting.set&log_rate_limiting=max_local_logging%03100%02max_remote_logging%03100000%02excluding%032

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> Service >> SLB >> Global >> Log Rate Limiting

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.31 SSL Management

7.31.1 “slb.ssl.getAll” Method

This method is used to get the entire x509 file stored on the AX device. The x509 file may be a certificate, certificate/key, key or CRL file. The necessary message is returned by an XML-based document. To get the source file, please use the slb.ssl.download method.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.ssl.getAll	String		

Note: (*) parameter is required in the API.

Response Fields

x509_files	xml tag for all x509 files
subnodes	
x509_file	xml tag of x509 file
subnodes	
file_name	the x509 file name
type	the x509 file type. possible value are: if the x509 file include both certificate and private key if the x509 file includes only a certificate if the x509 file includes only a key if the x509 file contents crl
certificate-key	
certificate	
key	
crl	
issuer	certificate issuer
common_name	certificate common name
division	certificate division
organization	certificate organization
locality	certificate locality
state_province	certificate state or province
country	certificate country
email	certificate email address
expiration	certificate expiration
key-size	the size of the key

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
GET
- HTTP Body
session_id=xxxxxxxxxxxxxxxxxxxxxx&method=slb.ssl.getAll

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
  <x509_files>
    <x509_file>
      <file_name>user2</file_name>
      <type>certificate-key</type>
      <issuer>Self</issuer>
      <common_name>user2</common_name>
      <division></division>
      <organization></organization>
      <locality></locality>
      <state_province></state_province>
      <country>US</country>
      <email></email>
      <expiration>Jan 14 06:43:56 2013 GMT</expiration>
      <key_size>2048</key_size>
    </x509_file>
    <x509_file>
      <file_name>test</file_name>
      <type>certificate</type>
      <issuer>CA</issuer>
      <common_name>User02</common_name>
      <division></division>
      <organization></organization>
      <locality></locality>
```

```

<state_province></state_province>
<country></country>
<email></email>
<expiration>May 30 02:22:34 2020 GMT</expiration>
</x509_file>
<x509_file>
<file_name>key</file_name>
<type>key</type>
<key_size>1024</key_size>
</x509_file>
<x509_file>
<file_name>AX.crl</file_name>
<type>crl</type>
<issuer>/C=CN/ST=BJ/L=HD/O=A10/OU=AX/CN=AX/emailAddress=xli@a10networksc.com.cn
</issuer>
</x509_file>
</x509_files>
</response>

```

Menus Privilege

Config Mode >> Service >> SSL Management >> Certificate

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.31.2 “slb.ssl.search” Method

Get information for a specific x509 file stored on the AX device. The x509 file may be a certificate, certificate/key or key. The necessary message is returned by an XML-based document. To get the source file, please use the slb.ssl.download method.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.ssl.search	String		
name(*)	Certificate/key name	String		

Note: (*) parameter is required in the API.

Response Fields

x509_file	xml tag of x509 file
subnodes	
file_name	the x509 file name
type	the x509 file type. possible value are: <i>certificate-key</i> <i>certificate</i> <i>key</i> <i>crl</i>
certificate-key	if the x509 file include both certificate and private key
certificate	if the x509 file includes only a certificate
key	if the x509 file includes only a key
crl	if the x509 file contents crl
issuer	certificate issuer
common_name	certificate common name
division	certificate division
organization	certificate organization
locality	certificate locality
state_province	certificate state or province
country	certificate country
email	certificate email address
expiration	certificate expiration

key-size the size of the key

Example Response

Request as:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
- **HTTP Action:**
GET
- **HTTP Body**
session_id=xxxxxxxxxxxxxxxxxxxxxx&method=slb.ssl.search&name=test

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
  <x509_file>
    <file_name>user2</file_name>
    <type>certificate-key</type>
    <issuer>Self</issuer>
    <common_name>user2</common_name>
    <division></division>
    <organization></organization>
    <locality></locality>
    <state_province></state_province>
    <country>US</country>
    <email></email>
    <expiration>Jan 14 06:43:56 2013 GMT</expiration>
    <key_size>2048</key_size>
  </x509_file>
</response>
```

Menus Privilege

Config Mode >> Service >> SSL Management >> Certificate

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.31.3 “slb.ssl.create” Method

This method is used to create x509 certificate and key file (if self-issued), or to create a key and CSR file with an issuing certificate authority.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.ssl.create	String		
file_name(*)	Certificate file name	String		
Issuer	Certificate issuer	String	'self','ca'	'self'
common_name(*)	Certificate common name	String		
division	Certificate division	String	"	
organization	Certificate organization	String	"	
locality	Certificate locality	String	"	
state-province	Certificate state or province	String	"	
country	Certificate country	String		'US'
email	Certificate email	String		"
password	Certificate password, if this option is present, this method will create a certificate signing request to third-part CA. if this option is not present, this method will create a self signed certificate.	String		"
expiration	certificate expiration (days)	Int	30 - 3650	730

key_size	Key size, 512, 1024 or 2048	Int	512, 1024, 2048	1024
-----------------	-----------------------------	-----	-----------------	------

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response1

Create with self issuer.

Request as:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
- **HTTP Action:**
POST
- **HTTP Body**
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.ssl.create&file_name=certca&common_name=abc&issuer=self&key_size=2048&division=AX&organization=A10&locality=SJ&state-province=CA&country=US&email=xli@a10networks.com.cn&expiration=365

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Example Response2

Create a certificate with issuer “ca”.

Request as:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
- **HTTP Action:**
POST
- **HTTP Body**
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.ssl.create&common_name=Network_management&file_name=catest&issuer=ca&key_size=2048&division=AX&organization=A10&locality=SJ&state-province=CA&country=US&email=xli@a10networks.com.cn&expiration=365

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
<csr_file>
<text>-----BEGIN CERTIFICATE REQUEST-----
MIICvDCCAAQCAQAwdzELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAKNBMQswCQYDVQQH
EwJTSjEMMAoGA1UEChMDQTEwMQswCQYDVQQLEwJBWDEMMAoGA1UEAxMDYWJjMSUw
...
4bRlgcwI9c30GZ81wjHMJiJ81dmmyvUxvuqshwHbmXVPMtMcGcEtGuePdEWYo09m
ptFM7pODFA+1B3dLx19Dmnwl/bI2ep7o38Yn/4chgwy= -----END CERTIFICATE REQUEST-----</text>
</csr_file>
</response>
```

Menus Privilege

Config Mode >> Service >> SSL Management >> Certificate

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

7.31.4 “slb.ssl.upload” Method

This method is used to upload a certificate, key, or CRL file to the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.ssl.upload	String		
type(*)	File type. 'certificate', 'key', 'crl'.	String	'certificate', 'key', 'crl'	

Note: (*) parameter is required in the API.

Response Fields

HTTP Code 404: not found

HTTP code 501: internal err

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
Multipart-POST
- HTTP Body
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.ssl.upload&type=crl

Response as the HTTP body:

Menus Privilege

Config Mode >> Service >> SSL Management >> Certificate

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.31.5 “slb.ssl.download” Method

This method is used to download a certificate, key, or CRL file to the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.ssl.download	String		
type(*)	File type. 'certificate', 'key', 'crl'	String	'certificate', 'key', 'crl'	
file_name(*)	The file name	String		

Note: (*) parameter is required in the API.

Response Fields

HTTP Code 404: not found

HTTP code 501: internal err

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
Multipart-GET
- HTTP Body
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.ssl.download&type=certificate&file_name=cert01

Response as the HTTP body:

Menus Privilege

Config Mode >> Service >> SSL Management >> Certificate

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.32 RAM Cache

7.32.1 “slb.cache.entry.clear” Method

This method is used to clear all slb cache entries (when a URI is not given), or this method can be used to clear specific cache entries (when a URI is given) under a specific virtual port.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.cache.entry.clear	String		
name (*)	tcp proxy template name	String		
vip_name(*)	the name of virtual server	String	Len: 1 – 31	
virtual_port(*)	Virtual port	Int	0 – 65534	
uri	The URI prefix of cache entries, when this option is absent, remove all cache entries. When this option is present, remove all the entries matches this prefix.	String	Len: 1- 31	

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
- **HTTP Action:**
POST
- **HTTP Body**
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.cache.entry.clear&vip_name=vs01&virtual_port=80

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Monitor Mode >> Service >> Application >> RAM Caching >> Details

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.32.2 “slb.cache.stats.clear” Method

This method is used to clear all slb cache statistics under a specific virtual port.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.cache.stats.clear	String		
name (*)	tcp proxy template name	String		
vip_name(*)	the name of virtual server	String	Len: 1 – 31	
virtual_port(*)	Virtual port	Int	0 – 65534	

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST
- HTTP Body
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.cache.stats.clear&vip_name=vs01&virtual_port=80

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Monitor Mode >> Service >> Application >> RAM Caching >> Objects

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.33 PBSLB

7.33.1 “slb.pbslb.getAll” Method

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.pbslb.getAll	String		

Note: (*) parameter is required in the API.

Response Fields

pbslb_list	pbslb list
pbslb	pbslb structure
name	name of the pbslb
local	pbslb is located in local host or remote host, local(1), remote(0)
interval	interval, available only when local=0
use_management_port	use management port, available only when local=0
host	host name or IP address, available only when local=0
location	location of the file (file name), available only when local=0

Example Response

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
GET

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxxxx&method=slb.pbslb.getAll

Response as the HTTP body:

```
<response status="ok">
<pbslb_list>
<pbslb>
<name>111</name>
<local>0</local>
<interval>300</interval>
<use_management_port>1</use_management_port>
<host>1.1.1.1</host>
<location>policy.txt</location>
</pbslb>
<pbslb>
<name>222</name>
<local>0</local>
<interval>86400</interval>
<use_management_port>0</use_management_port>
<host>192.168.3.214</host>
<location>policy.txt</location>
</pbslb>
<pbslb>
<name>333</name>
<local>0</local>
<interval>300</interval>
<use_management_port>1</use_management_port>
<host>1.1.1.2</host>
<location>policy.txt</location>
</pbslb>
</pbslb_list>
</response>
```

Menus Privilege

Config Mode >> Service >> SLB >> PBSLB

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.33.2 “slb.pbslb.search” Method

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.pbslb.search	String		
name(*)	Pbslb name	String		

Note: (*) parameter is required in the API.

Response Fields

pbslb_list	pbslb list
pbslb	pbslb structure
name	name of the pbslb
local	pbslb is located in local host or remote host, local(1), remote(0)
interval	interval, available only when local=0
use_management_port	use management port, available only when local=0
host	host name or IP address, available only when local=0
location	location of the file (file name), available only when local=0

Example Response

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
POST

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.pbslb.search&name=pbslb01

Response as the HTTP body:

```
<response status="ok">
<pbslb>
  <name>pbslb01</name>
  <local>0</local>
  <interval>300</interval>
  <use_management_port>1</use_management_port>
  <host>1.1.1.1</host>
  <location>policy.txt</location>
</pbslb>
</response>
```

Menus Privilege

Config Mode >> Service >> SLB >> PBSLB

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.33.3 “slb.pbslb.create” Method

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.pbslb.create	String		
Pbslb_list	Pbslb list	List		
pbslb	Pbslb structure	structure		
name(*)	Pbslb name	String		
local	Is pbslb located at local host or remote host	Bool	0 - 1	1
interval	interval, available only when local=0	int	60 - 86400	300
use_management-port	Use management port either 0 or 1, available only when local=0	bool	0 - 1	0
host	Host name or IP address, available only when local=0	String		
location	The name of the policy file in the remote host, available only when local=0	string		

Note: (*) parameter is required in the API.

Response Fields

Example Response

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
POST

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.pbslb.create&pbslb=name%03policy01%02interval%03300%02use_management_port%031%02host%03AAAA%02location%03policy01.txt
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?><response status="ok"></response>
```

Menus Privilege

Config Mode >> Service >> SLB >> PBSLB

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.33.4 “slb.pbslb.update” Method

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.pbslb.update	String		
Pbslb_list	Pbslb list	List		
pbslb	Pbslb structure	structure		
name(*)	Pbslb name	String		
local	Is pbslb located at local host or remote host	Bool	0 - 1	1
interval	interval, available only when local=0	int	60 - 86400	300

use_management-port	Use management port either 0 or 1, available only when local=0	bool	0 - 1	0
host	Host name or IP address, available only when local=0	String		
location	The name of the policy file in the remote host, available only when local=0	string		

Note: (*) parameter is required in the API.

Response Fields

Example Response

Request as:

- **URL:**
https://_AX_IP_Address_/services/rest/V2/
- **HTTP Action:**
POST

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.pbslb.update&pbslb=name%03policy01%02interval%03300%02use_management_port%03%02host%03AAAA%02location%03policy01.txt
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?><response status="ok"></response>
```

Menus Privilege

Config Mode >> Service >> SLB >> PBSLB

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.33.5 “slb.pbslb.delete” Method

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.pbslb.delete	String		
name(*)	Pbslb name	String		

Note: (*) parameter is required in the API.

Response Fields

Example Response

Request as:

- **URL:**
https://_AX_IP_Address_/services/rest/V2/
- **HTTP Action:**
POST

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.pbslb.delete&name=pbslb01
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?><response status="ok"></response>
```

Menus Privilege

Config Mode >> Service >> SLB >> PBSLB

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

7.33.6 “slb.pbslb.import” Method

Upload a policy file to AX through http multi-POST protocol (http upload).

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	slb.pbslb.import	String		

Note: (*) parameter is required in the API.

Response Fields

Generic HTTP multi-POST response

Example Response

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
Multi-POST (HTTP Upload)

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxx&method=slb.pbslb.import

Response as the HTTP body:

Menus Privilege

Config Mode >> Service >> SLB >> PBSLB

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

8 HA Management APIs

8.1 Global

8.1.1 “ha.global.get” Method

This method is used to get the global HA configuration information.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	ha.global.get	String		

Note: (*) parameter is required in the API.

Response Fields

ha_global	XML tag of HA global configuration
id	HA ID, when this option is set, ha is enabled, if this option is not set or the value is 0, ha is not enabled on this device.
set_id	HA set ID.
ha_preempt_status	Preempt status, 1: enabled, 0: disabled
interval	HA heartbeat interval
mirror_ip	HA mirror IP, the IP address on the other side
retry	When heartbeat failed, how many times to retries.
arp_retry	ARP retry times

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
ha id 1 set-id 1
ha conn-mirror ip 124.254.6.6
ha arp-retry 100
ha time-interval 3
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxx&method=ha.global.get

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
<ha_global>
<id>1</id>
<set_id>1</set_id>
<ha_preempt_status>0</ha_preempt_status>
<interval>3</interval>
<mirror_ip>124.254.6.6</mirror_ip>
<retry>5</retry>
<arp_retry>100</arp_retry>
</ha_global>
```

```
</response>
```

Menus Privilege

Config Mode >> HA >> Setting >> HA Global

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

8.1.2 “ha.global.set” Method

This method is used to set the global HA configuration information.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	ha.global.set	String		
id	ha id, when this option is set, ha is enabled, if this option is not set or the value is 0, ha is not enabled on this device.	Int	0 or 1	0
set_id	ha set id.	Int	1 - 7	0
ha_preempt_status	preempt status, 1: enabled, 0: disabled	Int	0 or 1	0
interval	ha heartbeat interval, (unit: ms)	Int	1 - 255	2
mirror_ip	ha mirror ip, the ip address on the other side	String		“”
retry	when heartbeat failed, how many times to retries	Int	2 – 255	5
arp_retry	arp retry times	Int	1 - 255	4

Note: (*) parameter is required in the API.

Response Fields

General XML response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!  
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=ha.global.set&id=2
```

CLI configuration after aXAPI call:

```
!  
ha id 2  
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>  
<response status="ok">  
</response>
```

Menus Privilege

Config Mode >> HA >> Setting >> HA Global

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

8.1.3 “ha.sync_config” Method

This method is used to synchronize the Layer 4-7 configuration information with the other AX device in the HA pair.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	ha.sync_config	String		
user (*)	User name	String	1-32	
password(*)	password	String	1-64	
sync_all_partitions	Sync all partition? 0:no / 1:yes	Int	0 or 1	0
operation	operation type, 0: All; 1: Startup-config 2:Running-config 3: Data Files	Int	0,1,2,3	0
peer_destination	Peer options, 0:to startup config, 1:to running config	Int	0 or 1	1
peer_reload	Peer reload option. 0:no / 1:yes	Int	0 or 1	0

Note: (*) parameter is required in the API.

Response Fields

General XML response

Example Response

Request as:

- URL:
[https://\[AX_IP_ADDRESS\]/services/rest/V2/](https://[AX_IP_ADDRESS]/services/rest/V2/)
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!  
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=ha.sync_config&user=admin&password=password&sync_all_partitions=1&operation=0&peer_destination=1&peer_reload=0
```

Note: After running this method, both AX devices (in HA configuration) will have exactly the same configuration.

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>  
<response status="ok">  
</response>
```

Menus Privilege

Config Mode >> HA >> Setting >> HA Global

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

8.1.4 “ha.force_self_standby” Method

This method is used to force HA groups into a standby state on a given AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	ha.force_self_standby	String		
ha_group (*)	Ha group ID, 0 means All	Int	0-32	
flag(*)	1: Force Self Standby 0: Cancel Force Self Standby	Int	0 or 1	

Note: (*) parameter is required in the API.

Response Fields

General XML response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST
- CLI configuration before aXAPI call:

```
!
ha id 1 set-id 1
ha group 1 priority 1
!
```

- HTTP Body
`session_id=xxxxxxxxxxxxxxxxxxxx&method=ha.force_self_standby&ha_group=1&flag=1`
- CLI configuration after aXAPI call:
N/A

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> HA >> Setting >> HA Global

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

8.1.5 “ha.status.get” Method

This method is used to get the entire HA status for this AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		

method (*)	ha.status.get	String		
-------------------	----------------------	--------	--	--

Note: (*) parameter is required in the API.

Response Fields

ha_status	xml tag of ha status
ha_group_status	xml tag of status of one ha group
config_sync_status	Config Sync Status. 1: Syncronized; 0: Not Syncronized
last_sync_time	Last Syncronized Time
connectivity_server_ports	Connectivity Server Ports
connectivity_router_ports	Connectivity Router Ports
ha_packages_sent	HA Packets Sent
ha_packages_received	HA Packets Received
conn_sync_sent	HA Conn Sync Sent
conn_sync_received	HA Conn Sync Received
errors_in_duplicated_ha_id	HA Errors: In Duplicated HA ID
errors_in_invalid_group	HA Errors: In Invalid Group
errors_version_mismatch	HA Errors: Version Mismatch
errors_set_id_mismatch	HA Errors: HA Set ID Mismatch
errors_miss_heartbeat	HA Errors: Missed Heartbeat
errors_inaccurate_timer	HA Errors: Inaccurate Timer
ha_port_list	xml tag of ha port list
ha_port	xml tag of ha port
port_num	port num
sent	Packets sent
received	Packets received
miss_heartbeat	Missed Heartbeat

Example Response

Request as:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
- **HTTP Action:**
GET
- **CLI configuration before axAPI call:**
N/A
- **HTTP Body**
session_id=xxxxxxxxxxxxxxxxxxxx&method=ha.status.get

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
    <ha_status>
        <config_sync_status>0</config_sync_status>
        <last_sync_time />
        <connectivity_server_ports>0</connectivity_server_ports>
        <connectivity_router_ports>0</connectivity_router_ports>
        <ha_packages_sent>0</ha_packages_sent>
        <ha_packages_received>0</ha_packages_received>
        <conn_sync_sent>0</conn_sync_sent>
        <conn_sync_received>0</conn_sync_received>
        <errors_in_duplicated_ha_id>0</errors_in_duplicated_ha_id>
        <errors_in_invalid_group>0</errors_in_invalid_group>
        <errors_version_mismatch>0</errors_version_mismatch>
        <errors_set_id_mismatch>0</errors_set_id_mismatch>
        <errors_miss_heartbeat>0</errors_miss_heartbeat>
        <errors_inaccurate_timer>0</errors_inaccurate_timer>
    </ha_status>
</response>
```

```
</response>
```

Menus Privilege

Config Mode >> HA >> Status

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

8.2 Group

8.2.1 “ha.group.getAll” Method

This method is used to get configuration information for all HA groups configured on the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	ha.group.getAll	String		

Note: (*) parameter is required in the API.

Response Fields

group_list	xml tag of ha group list
group	xml tag of ha group
name	ha group name
priority	ha priority of this group

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
ha group 1 priority 100
ha group 2 priority 50
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxx&method=ha.group.getAll

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
<group_list>
<group>
<name>1</name>
<priority>100</priority>
</group>
<group>
<name>2</name>
<priority>50</priority>
```

```

</group>
</group_list>
</response>

```

Menus Privilege

Config Mode >> HA >> Setting >> HA Global

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

8.2.2 “ha.group.fetchStatistics” Method

This method is used to get all statistics for a given HA group.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	ha.group.fetchStatistics	String		

Note: (*) parameter is required in the API.

Response Fields

ha_group_status_list	xml tag of ha group list
ha_group_status	xml tag of status of one ha group
id	ha group id
local_status	ha group local status. 1:active; 0: standby
local_priority	ha group local priority
peer_status	peer status. 1:active; 0: standby
peer_priority	peer priority
force_self_standby	force self standby. 1:yes; 0:no

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
ha group 1 priority 1
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxx&method=ha.group.fetchStatistics

Response as the HTTP body:

```

<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
    <ha_group_status_list>
        <ha_group_status>
            <id>1</id>
            <local_status>1</local_status>
            <local_priority>1</local_priority>
            <peer_status>255</peer_status>
            <peer_priority>0</peer_priority>
            <force_self_standby>0</force_self_standby>
        </ha_group_status>
    </ha_group_status_list>
</response>

```

```

</ha_group_status>
</ha_group_status_list>
</response>

```

Menus Privilege

Monitor Mode >> HA >> Group

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

8.2.3 “ha.group.create” Method

This method is used to add an HA group configuration to the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	ha.group.create	String		
name(*)	HA group name	String	1 - 31	
priority(*)	HA group priority	String	1 - 255	

Note: (*) parameter is required in the API.

Response Fields

General XML response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
ha group 1 priority 1
ha group 2 priority 50
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxx&method=ha.group.create&name=22&priority=2

CLI configuration after aXAPI call:

```
!
ha group 1 priority 1
ha group 2 priority 50
ha group 22 priority 2
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> HA >> Setting >> HA Global

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

8.2.4 “ha.group.delete” Method

This method is used to delete an HA group from the AX.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	ha.group.delete	String		
name(*)	HA group name	String	1 - 31	
priority(*)	HA group priority	String	1 - 255	

Note: (*) parameter is required in the API.

Response Fields

General XML response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
ha group 1 priority 1
ha group 2 priority 50
ha group 22 priority 2
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxx&method=ha.group.delete&name=22

CLI configuration after aXAPI call:

```
!
ha group 1 priority 1
ha group 2 priority 50
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> HA >> Setting >> HA Global

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

8.3 Status Check

8.3.1 “ha.status_check.gw.getAll” Method

This method is used to get configuration information for all HA status check gateways configured on the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	ha.status_check.gw.getAll	String		

Note: (*) parameter is required in the API.

Response Fields

status_check_gw_list	xml tag of gateway list of ha status check
status_check_gw	xml tag of gateway
gw_addr	the address of the gateway

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
ha check gateway 124.254.6.2
ha check gateway 1.2.3.6
ha check gateway 255.255.255.1
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxx&method=ha.status_check.gw.getAll

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
<status_check_gw_list>
    <status_check_gw>
        <gw_addr>124.254.6.2</gw_addr>
    </status_check_gw>
    <status_check_gw>
        <gw_addr>1.2.3.6</gw_addr>
    </status_check_gw>
    <status_check_gw>
        <gw_addr>255.255.255.1</gw_addr>
    </status_check_gw>
</status_check_gw_list>
</response>
```

Menus Privilege

Config Mode >> HA >> Setting >> HA Global

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

8.3.2 “ha.status_check.gw.create” Method

This method is used to add a gateway to an HA status check.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	ha.status_check.gw.create	String		
gw_addr (*)	HA status check gateway IP address	String		

Note: (*) parameter is required in the API.

Response Fields

General XML response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
ha check gateway 124.254.6.2
ha check gateway 1.2.3.6
ha check gateway 255.255.255.1
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxx&method=ha.status_check.gw.create&gw_addr=192.168.1.1

CLI configuration after aXAPI call:

```
!
ha check gateway 124.254.6.2
ha check gateway 1.2.3.6
ha check gateway 255.255.255.1
ha check gateway 192.168.1.1
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> HA >> Setting >> HA Global

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

8.3.3 “ha.status_check.gw.delete” Method

The method is used to delete a gateway from an HA status check.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	ha.status_check.gw.delete	String		
gw_addr (*)	HA status check gateway IP address	String		

Note: (*) parameter is required in the API.

Response Fields

General XML response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
ha check gateway 124.254.6.2
ha check gateway 1.2.3.6
ha check gateway 255.255.255.1
ha check gateway 192.168.1.1
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxx&method=ha.group.delete&gw_addr=192.168.1.1

CLI configuration after aXAPI call:

```
!
ha check gateway 124.254.6.2
ha check gateway 1.2.3.6
ha check gateway 255.255.255.1
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> HA >> Setting >> HA Global

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

8.3.4 “ha.status_check.vlan.getAll” Method

This method is used to get configuration information for all HA status checks for VLANs configured on the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	ha.status_check.vlan.getAll	String		

Note: (*) parameter is required in the API.

Response Fields

status_check_vlan_list	xml tag of vlan list of ha status check
status_check_vlan	xml tag of vlan
vlan_id	vlan id
vlan_timeout	vlan timeout

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
vlan 2
tagged ethernet 8
router-interface ve 2
!
vlan 123
tagged ethernet 7
router-interface ve 123
!
interface ve 2
ip address 2.2.2.2 255.255.255.0
!
interface ve 123
ip address 123.123.123.123 255.255.255.0
!
ha check vlan 2 timeout 2
ha check vlan 123 timeout 600
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxx&method=ha.status_check.vlan.getAll

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
<status_check_vlan_list>
<status_check_vlan>
<vlan_id>2</vlan_id>
<vlan_timeout>2</vlan_timeout>
</status_check_vlan>
<status_check_vlan>
<vlan_id>123</vlan_id>
<vlan_timeout>600</vlan_timeout>
</status_check_vlan>
</status_check_vlan_list>
</response>
```

Menus Privilege

Config Mode >> HA >> Setting >> HA Global

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

8.3.5 “ha.status_check.vlan.create” Method

This method is used to add a VLAN to an HA status check.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	ha.status_check.vlan.create	String		
vlan_id (*)	Vlan ID	Num	1-4094	
vlan_timeout(*)	Vlan timeout	Num	2-600	

Note: (*) parameter is required in the API.

Response Fields

General XML response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
vlan 2
tagged ethernet 8
router-interface ve 2
!
interface ve 2
ip address 2.2.2.2 255.255.255.0
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxx&method=ha.status_check.vlan.create&vlan_id=1&vlan_timeout=30

CLI configuration after aXAPI call:

```
!
vlan 2
tagged ethernet 8
router-interface ve 2
!
interface ve 2
ip address 2.2.2.2 255.255.255.0
!
ha check vlan 2 timeout 30
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> HA >> Setting >> HA Global

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

8.3.6 “ha.status_check.vlan.delete” Method

This method is used to delete a VLAN from an HA status check.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	ha.status_check.vlan.delete	String		
vlan_id (*)	Vlan ID	Num	1-4094	

Note: (*) parameter is required in the API.

Response Fields

General XML response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
ha check vlan 2 timeout 30
ha check vlan 123 timeout 600
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxx&method=ha.status_check.vlan.delete&vlan_id=1

CLI configuration after aXAPI call:

```
!
ha check vlan 123 timeout 600
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> HA >> Setting >> HA Global

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

8.4 Floating IP

8.4.1 “ha.floating_ip.getAll” Method

This method is used to get configuration information for all floating IPs (or the IP addresses that downstream devices use as their default gateway) configured for all HA pairs configured on the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	ha.floating_ip.getAll	String		

Note: (*) parameter is required in the API.

Response Fields

floating_ip_list	xml tag of floating ip list
floating_ip	xml tag of floating ip
name	the ha group name(id)
floating_ip	
vlan_timeout	

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
floating-ip 124.254.6.6 ha-group 1
floating-ip 3ff5::6789 ha-group 2
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxx&method=ha.floating_ip.getAll

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
    <floating_ip_list>
        <floating_ip>
            <name>1</name>
            <floating_ip>124.254.6.6</floating_ip>
        </floating_ip>
        <floating_ip>
            <name>2</name>
            <floating_ip>3ff5::6789</floating_ip>
        </floating_ip>
    </floating_ip_list>
</response>
```

Menus Privilege

Config Mode >> HA >> Setting >> HA Global

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

8.4.2 “ha.floating_ip.create” Method

This method is used to create a floating IP (or the IP address that downstream devices use as their default gateway) in an HA pair.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	ha.floating_ip.create	String		
name(*)	Group Name	Num	1-31	
floating_ip(*)	Floating ip address(IPv4 or IPv6)	String	1-64	

Note: (*) parameter is required in the API.

Response Fields

General XML response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
floating-ip 124.254.6.6 ha-group 1
floating-ip 3ff5::6789 ha-group 2
!
```

HTTP Body
 session_id=xxxxxxxxxxxxxxxxxxxx&method=ha.floating_ip.create&name=1&floating_ip=192.168.4.22

CLI configuration after aXAPI call:

```
!
floating-ip 124.254.6.6 ha-group 1
floating-ip 192.168.4.22 ha-group 1
floating-ip 3ff5::6789 ha-group 2
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> HA >> Setting >> HA Global

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

8.4.3 “ha.floating_ip.delete” Method

This method is used to delete an HA floating IP address (IPv4 or IPv6).

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	ha.floating_ip.delete	String		
name(*)	Group Name	Num	1-31	
floating_ip(*)	Floating ip address(IPv4 or IPv6)	String	1-64	

Note: (*) parameter is required in the API.

Response Fields

General XML response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
floating-ip 124.254.6.6 ha-group 1
floating-ip 192.168.4.22 ha-group 1
floating-ip 3ff5::6789 ha-group 2
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxx&method=ha.floating_ip.delete&name=1&floating_ip=192.168.4.2
2

CLI configuration after aXAPI call:

```
!
floating-ip 124.254.6.6 ha-group 1
floating-ip 3ff5::6789 ha-group 2
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> HA >> Setting >> HA Global

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

8.5 Inline Mode

8.5.1 “ha.inline_mode.get” Method

This method is used to get inline mode configuration information, such as status, preferred port, or restart time, for an HA pair.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	ha.inline_mode.get	String		

Note: (*) parameter is required in the API.

Response Fields

ha_inline_mode	XML tag of ha inline_mode configuration
status	Inline Mode Status, 1: enabled, 0: disabled
preferred_port	Preferred port
restart_time	Restart Time(100 ms)
restart_port_list	XML tag of Restart Port List
restart_port	Restart Port Number
l3_inline_mode	L3 Inline Mode: 1: enabled, 0: disabled
link_event_delay	Link Event Delay

Example Response

Request as:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
- **HTTP Action:**
GET

CLI configuration before aXAPI call:

```
!
ha id 1 set-id 1
ha inline-mode
ha restart-port-list ethernet 1 to 2
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxxxx&method=ha.inline_mode.get

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
  <ha_inline_mode>
    <status>1</status>
    <preferred_port>0</preferred_port>
    <restart_time>20</restart_time>
    <restart_port_list>
      <restart_port>1</restart_port>
      <restart_port>2</restart_port>
    </restart_port_list>
    <l3_inline_mode>0</l3_inline_mode>
    <link_event_delay>30</link_event_delay>
  </ha_inline_mode>
</response>
```

Menus Privilege

Config Mode >> HA >> Setting >> HA Inline Mode

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

8.5.2 “ha.inline_mode.set” Method

This method is used to set inline mode configuration information, such as status, preferred port, or restart time, for an HA pair.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	ha.inline_mode.set	String		
status	Inline Mode Status, 1: enabled, 0: disabled	Int	0 or 1	0
preferred_port	Preferred Port	Int	1-100	
restart_time	Restart Time	Int	1-100	20
l3_inline_mode	L3 Inline Mode, 0:disable / 1:enable	Int	0 or 1	0
link_event_delay	Link Event Delay	Int	1 - 100	30
restart_port_list	List of restart port list	List		
eth<n> restart_port	header erase request at element <n> restart_port	Int	0-22	

Note: (*) parameter is required in the API.

Response Fields

General XML response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
ha id 1 set-id 1
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=ha.inline_mode.set&status=1&preferred_port=80&restart_time=30&l3_inline_mode=0&link_event_delay=50&restart_port_list=eth1%02eth2&eth1=restart_port%031&eth2=restart_port%032
```

CLI configuration after aXAPI call:

```
!
ha id 1 set-id 1
ha inline-mode preferred-port 8
ha restart-port-list ethernet 1 to 2
ha restart-time 30
ha link-event-delay 50
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> HA >> Setting >> HA Inline Mode

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

8.6 Interface

8.6.1 “ha.interface.getAll” Method

This method is used to get configuration information for all interfaces configured for HA pairs on the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	ha.interface.getAll	String		

Note: (*) parameter is required in the API.

Response Fields

ha_interface_list	xml tag of floating ip list
ha_interface	xml tag of floating ip
if_num:	port number
type	interface type(Ethernet/ve/loop)
status	0 means Disable / 1 means Enable
ha_status	0 means Disable / 1 means Enable
ha_type	0: none, 1: server, 2: route, 3: both
heartbeat	0 means Disable / 1 means Enable
vlan_id	Vlan ID

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
ha interface ethernet 1 no-heartbeat
ha interface ethernet 2 both vlan 1
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxxxx&method=ha.interface.getAll

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
    <ha_interface_list>
        <ha_interface>
            <if_num>1</if_num>
            <type>0</type>
            <status>0</status>
            <ha_status>1</ha_status>
            <ha_type>0</ha_type>
            <heartbeat>0</heartbeat>
```

```

<vlan_id>0</vlan_id>
</ha_interface>
<ha_interface>
  <if_num>2</if_num>
  <type>0</type>
  <status>0</status>
  <ha_status>0</ha_status>
  <ha_type>0</ha_type>
  <heartbeat>1</heartbeat>
<vlan_id>0</vlan_id>
</ha_interface>
</ha_interface_list>
</response>

```

Menus Privilege

Config Mode >> HA >> Setting >> HA Interface

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

8.6.2 “ha.interface.get” Method

This method is used to get configuration information for a given interfaces in an HA pair that is configured on the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	ha.interface.get	String		
if_num(*)	Interface number	Int	1-100	

Note: (*) parameter is required in the API.

Response Fields

ha_interface	xml tag of floating ip
if_num	port number
type	interface type(Ethernet/ve/loop)
status	0 means Disable / 1 means Enable
ha_status	0 means Disable / 1 means Enable
ha_type	0: none, 1: server, 2: route, 3: both
heartbeat	0 means Disable / 1 means Enable
vlan_id	Vlan ID

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
ha interface ethernet 2 both vlan 1
!
```

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxx&method=ha.interface.get&if_num=2

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
    <ha_interface>
        <if_num>2</if_num>
        <type>0</type>
        <status>0</status>
        <ha_status>0</ha_status>
        <ha_type>3</ha_type>
        <heartbeat>1</heartbeat>
        <vlan_id>1</vlan_id>
    </ha_interface>
</response>
```

Menus Privileq

Config Mode >> HA >> Setting >> HA Interface

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

8.6.3 “ha.interface.set” Method

This method is used to set parameters for a specific interface in an HA pair configured on the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	ha.interface.set	String		
if_num(*)	Interface number	Int	1-100	
status	0:disable / 1:enable	Int	0 or 1	0
ha_status	ha status, 0:disable / 1:enable	Int	0 or 1	0
ha_type	ha type, 0 , 1, 2, 3 0:none, 1:server, 2:route, 3:both	Int	0,1,2 or 3	0
heartbeat	Heartbeat, 0:disable / 1:enable	Int	0 or 1	0
vlan_id	Vlan ID	Int	1 - 4094	

Note: (*) parameter is required in the API.

0: none, 1: server, 2: route, 3: both

Response Fields

General XML response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
!
```

HTTP Body

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=ha.interface.set&if_num=2&status=1&ha_
type=3&heartbeat=1&vlan_id=1
```

CLI configuration after aXAPI call:

```
!
ha interface ethernet 2 both vlan 1
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> HA >> Setting >> HA Interface

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9 GSLB Management APIs

9.1 DNS Proxy

9.1.1 “gslb.dns_proxy.getAll” Method

This method is used to get configuration information for all GSLB DNS proxies configured on the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	gslb.dns_proxy.getAll	String		

Note: (*) parameter is required in the API.

Response Fields

gslb_vserver_list	XML tag for the collection of gslb virtual server list
gslb_vserver	XML tag for gslb virtual server
name	GSLB DNS proxy name
ip_address	IP address of the dns proxy
status	status, enabled(1) or disabled(0)
ha_group	HA group ID (1 - 31)
ha_group_dynamic_weight	Dynamic weight of the DNS proxy in the HA group
vport_list	Virtual port list of the GSLB DNS proxy
vport	Virtual port of the GSLB DNS proxy
port_number	Virtual port number (1 - 65535)
service_group	service group(name) of this virtual port
status	enabled(1) or disabled(0)
connection_limit	connection limit (1 - 8000000)
over_connection_limit_action	over connection limit action, drop(0), reset (1)
source_nat_pool	source NAT pool name
aflex	aFlex(name) of this virtual port
udp_template	UDP template (name) of this virtual port

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
ha group 1 priority 1
!
ip nat pool nat_v4-o1 13.13.13.11 13.13.13.20 netmask /24 ha-group-id 1
!
slb server _s_9.9.9.1 9.9.9.1
    conn-limit 8000000 no-logging
    port 6000 udp
        conn-limit 8000000 no-logging
    port 10000 udp
        conn-limit 8000000 no-logging
!
```

```

slb service-group http_srv_grp1 udp
    member _s_9.9.9.1:6000
!
slb service-group http udp
    member _s_9.9.9.1:10000
!
slb template udp udpt
    aging short
!
slb virtual-server dns_proxy_01 2.33.6.31
    ha-group 1
    ha-dynamic 1
    port 6400 udp
        name _2.33.6.31_UDP_6400
        gslb-enable
        service-group http_srv_grp1
        conn-limit 8000000 no-logging
    port 13568 udp
        name _2.33.6.31_UDP_13568
        gslb-enable
        source-nat pool nat_v4-01
        service-group http
        template udp udpt
        conn-limit 1000000 reset no-logging
!
slb virtual-server dns_proxy-02 253.66.32.129
    port 13568 udp
        name _253.66.32.129_UDP_13568
        gslb-enable
        conn-limit 8000000 no-logging
!

```

1. HTTP Request in URL Format:

session_id=xxxxxxxxxxxxxxxxxxxxxx&method=gslb.dns_proxy.getAll&format=url

Response as the HTTP body (URL Format):

```

<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
    <gslb_vserver_list>
        <gslb_vserver>
            <name>dns_proxy_01</name>
            <ip_address>2.33.6.31</ip_address>
            <status>1</status>
            <ha_group>1</ha_group>
            <ha_group_dynamic_weight>1</ha_group_dynamic_weight>
            <vport_list>
                <vport>
                    <port_number>6400</port_number>
                    <service_group>http_srv_grp1</service_group>
                    <status>1</status>
                    <ha_connection_mirror>0</ha_connection_mirror>
                    <connection_limit>8000000</connection_limit>
                    <over_connection_limit_action>0</over_connection_limit_action>
                    <source_nat_pool/>
                    <aflex/>
                    <udp_template/>
                </vport>
                <vport>
                    <port_number>13568</port_number>
                    <service_group>http</service_group>
                    <status>0</status>
                    <ha_connection_mirror>0</ha_connection_mirror>
                    <connection_limit>1000000</connection_limit>
                    <over_connection_limit_action>1</over_connection_limit_action>
                    <source_nat_pool>nat_v4-01</source_nat_pool>
                    <aflex/>
                </vport>
            </vport_list>
        </gslb_vserver>
    </gslb_vserver_list>
</response>

```

```

                <udp_template>udpt</udp_template>
            </vport>
        </vport_list>
    </gslb_vserver>
    <gslb_vserver>
        <name>dns_proxy-02</name>
        <ip_address>253.66.32.129</ip_address>
        <status>1</status>
        <ha_group>0</ha_group>
        <ha_group_dynamic_weight>0</ha_group_dynamic_weight>
        <vport_list>
            <vport>
                <port_number>13568</port_number>
                <service_group/>
                <status>1</status>
                <ha_connection_mirror>0</ha_connection_mirror>
                <connection_limit>8000000</connection_limit>
                <over_connection_limit_action>0</over_connection_limit_action>
                <source_nat_pool/>
                <aflex/>
                <udp_template/>
            </vport>
        </vport_list>
    </gslb_vserver>
</gslb_vserver_list>
</response>

```

2. HTTP Request in JSON Format:

session_id=xxxxxxxxxxxxxxxxxxxxxx&method=gslb.dns_proxy.getAll&format=json

Response as the HTTP body (JSON Format):

```
{
    "gslb_vserver_list": [
        {
            "name": "dns_proxy_01",
            "ip_address": "2.33.6.31",
            "status": 1,
            "ha_group": 1,
            "ha_group_dynamic_weight": 1,
            "vport_list": [
                {
                    "port_number": 13568,
                    "service_group": "http",
                    "status": 1,
                    "ha_connection_mirror": 0,
                    "connection_limit": 1000000,
                    "over_connection_limit_action": 1,
                    "source_nat_pool": "nat_v4-01",
                    "aflex": "",
                    "udp_template": "udpt"
                },
                {
                    "port_number": 6400,
                    "service_group": "http_srv_grp1",
                    "status": 1,
                    "ha_connection_mirror": 0,
                    "connection_limit": 8000000,
                    "over_connection_limit_action": 0,
                    "source_nat_pool": "",
                    "aflex": "",
                    "udp_template": ""
                }
            ]
        }
    ]
}
```

```

        }
    ],
{
    "name": "dns_proxy-02",
    "ip_address": "253.66.32.129",
    "status": 1,
    "ha_group": 0,
    "ha_group_dynamic_weight": 0,
    "vport_list": [
        {
            "port_number": 13568,
            "service_group": "",
            "status": 1,
            "ha_connection_mirror": 0,
            "connection_limit": 8000000,
            "over_connection_limit_action": 0,
            "source_nat_pool": "",
            "aflex": "",
            "udp_template": ""
        }
    ]
}
}

```

Menus Privilege

Config Mode >> Service >> GSLB >> DNS Proxy

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.1.2 “gslb.dns_proxy.search” Method

This method is used to search all information for a GSLB DNS proxy configured on the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	gslb.dns_proxy.search	String		
name(*)	the name of the gslb dns proxy	String		

Note: (*) parameter is required in the API.

Response Fields

gslb_vserver	XML tag for gslb virtual server
name	GSLB DNS proxy name
ip_address	IP address of the dns proxy
status	status, enabled (1) or disabled(0)
ha_group	HA group ID (1 - 31)
ha_group_dynamic_weight	dynamic weight of the DNS proxy in the HA group
vport_list	virtual port list of the GSLB DNS proxy
vport	virtual port of the GSLB DNS proxy
port_number	virtual port number (1 - 65535)
service_group	service group(name) of this virtual port
status	enabled(1) or disabled(0)

connection_limit	connection limit (1 - 8000000)
over_connection_limit_action	over connection limit action, drop(0), reset (1)
source_nat_pool	source nat pool name
aflex	aFlex(name) of this virtual port
udp_template	UDP template (name) of this virtual port

Example Response

Request as:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
- **HTTP Action:**
GET

CLI configuration before aXAPI call:

```
!
ha group 1 priority 1
!
ip nat pool nat_v4-o1 13.13.13.11 13.13.13.20 netmask /24 ha-group-id 1
!
slb server _s_9.9.9.1 9.9.9.1
    conn-limit 8000000 no-logging
    port 6000 udp
        conn-limit 8000000 no-logging
    port 10000 udp
        conn-limit 8000000 no-logging
!
slb service-group http_srv_grp1 udp
    member _s_9.9.9.1:6000
!
slb template udp udpt
    aging short
!
slb virtual-server dns_proxy_01 2.33.6.31
    ha-group 1
    ha-dynamic 1
    port 6400 udp
        name _2.33.6.31_UDP_6400
        gslb-enable
        service-group http_srv_grp1
        conn-limit 8000000 no-logging
    port 13568 udp
        name _2.33.6.31_UDP_13568
        gslb-enable
        source-nat pool nat_v4-o1
        service-group http
        template udp udpt
        conn-limit 1000000 reset no-logging
!
```

1. HTTP Request in URL Format:

session_id=xxxxxxxxxxxxxxxxxxxxxx&method=gslb.dns_proxy.search&format=url

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
    <gslb_vserver>
        <name>dns_proxy_01</name>
        <ip_address>2.33.6.31</ip_address>
        <status>1</status>
        <ha_group>1</ha_group>
        <ha_group_dynamic_weight>1</ha_group_dynamic_weight>
        <vport_list>
            <vport>
                <port_number>6400</port_number>
                <service_group>http_srv_grp1</service_group>
            
```

```

<status>1</status>
<ha_connection_mirror>0</ha_connection_mirror>
<connection_limit>8000000</connection_limit>
<over_connection_limit_action>0</over_connection_limit_action>
<source_nat_pool/>
<aflex/>
<udp_template/>
</vport>
<vport>
<port_number>13568</port_number>
<service_group>http</service_group>
<status>0</status>
<ha_connection_mirror>0</ha_connection_mirror>
<connection_limit>1000000</connection_limit>
<over_connection_limit_action>1</over_connection_limit_action>
<source_nat_pool>nat_v4-01</source_nat_pool>
<aflex/>
<udp_template>udpt</udp_template>
</vport>
</vport_list>
</gslb_vserver>
</response>

```

2. HTTP Request in JSON Format:

session_id=xxxxxxxxxxxxxxxxxxxxxx&method=gslb.dns_proxy.search&format=json

Request POST data (JSON Format):

```
{
  "name": "dns_proxy_01"
}
```

Response as the HTTP body (JSON Format):

```
{
  "gslb_vserver": [
    {
      "name": "dns_proxy_01",
      "ip_address": "2.33.6.31",
      "status": 1,
      "ha_group": 1,
      "ha_group_dynamic_weight": 1,
      "vport_list": [
        {
          "port_number": 13568,
          "service_group": "http",
          "status": 1,
          "ha_connection_mirror": 0,
          "connection_limit": 1000000,
          "over_connection_limit_action": 1,
          "source_nat_pool": "nat_v4-01",
          "aflex": "",
          "udp_template": "udpt"
        },
        {
          "port_number": 6400,
          "service_group": "http_srv_grp1",
          "status": 1,
          "ha_connection_mirror": 0,
          "connection_limit": 8000000,
          "over_connection_limit_action": 0
        }
      ]
    }
  ]
}
```

```

        "source_nat_pool":"",
        "aflex":"",
        "udp_template":""
    }
]
}
}

```

Menus Privilege

Config Mode >> Service >> GSLB >> DNS Proxy

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.1.3 “gslb.dns_proxy.create” Method

This method is used to create a GSLB DNS proxy on the AX device.

Parameters

Parameter name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	gslb.dns_proxy.create	String		
name (*)	virtual server template name	String	1 - 31	
Ip address	ip address of the dns proxy	IPv4		
status	status, enabled(1) or disabled(0)	INT	0 - 1	
ha_group	ha group ID	INT	1 - 31	
ha_group_dynamic_weight	dynamic weight of the dns proxy in the ha group	INT	1 - 255	
vport_list	Virtual port list statement	LIST		
vport	Vport items. Has the following parameters	OBJ		
port_number	virtual port number	INT	1 - 65535	
service_group	service group of this virtual port	String	1 - 31	
status	enabled(1) or disabled(0)	INT	0 - 1	
connection_limit	connection limit, 0 means no limit	INT	0 - 8000000	
over_connection_limit_action	over connection limit action, drop(0), reset(1)	INT	0 - 1	
source_nat_pool	source nat pool name	String	1 - 31	
aflex	aFlex(name) of this virtual port	String	1 - 31	
udp_template	udp template of this virtual port	String	1 - 31	

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
- **HTTP Action:**
POST

CLI configuration before aXAPI call:

```
!
ha group 31 priority 255
!
```

1. HTTP Request in URL Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=gslb.dns_proxy.create&gslb_vserver=name%03YYFYY%02ip_address%038.2.3.18%02status%030%02ha_group%0331%02ha_group_dynamic_weight%0310%02vport_list&vport_list=P1%02P2&P1=port_number%0312345%02status%030&P2=port_number%0312346%02status%030&format=url
```

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in JSON Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=gslb.dns_proxy.create&format=json
```

Request POST data (JSON Format):

```
{
  "gslb_vserver": {
    {
      "name": "YYFYY",
      "ip_address": "8.2.3.18",
      "status": 0,
      "ha_group": 31,
      "ha_group_dynamic_weight": 10,
      "vport_list": [
        {
          {
            "port_number": 12346,
            "service_group": "",
            "status": 0,
            "ha_connection_mirror": 0,
            "connection_limit": 8000000,
            "over_connection_limit_action": 0,
            "source_nat_pool": "",
            "aflex": "",
            "udp_template": ""
          },
          {
            "port_number": 12345,
            "service_group": "",
            "status": 0,
            "ha_connection_mirror": 0,
            "connection_limit": 8000000,
            "over_connection_limit_action": 0,
            "source_nat_pool": "",
            "aflex": "",
            "udp_template": ""
          }
        ]
      }
    }
  }
}
```

CLI configuration after aXAPI call:

```
!
ha group 31 priority 255
!
slb virtual-server YYFYY 8.2.3.18
```

```

 disable
 ha-group 31
 ha-dynamic 10
 port 12345 udp
   name _8.2.3.18_UDP_12345
   disable
   gslb-enable
 port 12346 udp
   name _8.2.3.18_UDP_12346
   disable
   gslb-enable
!
```

Response as the HTTP body:

```
{
  "response": {
    "status" : "OK"
  }
}
```

Menus Privilege

Config Mode >> Service >> GSLB >> DNS Proxy

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.1.4 “gslb.dns_proxy.update” Method

This method is used to update one or more parameters for a GSLB DNS proxy configured on the AX device.

Parameters

Parameter name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	gslb.dns_proxy.update	String		
name (*)	virtual server template name	String	1 - 31	
ip_address	ip address of the dns proxy	IPv4		
status	status, enabled(1) or disabled(0)	INT	0 - 1	
ha_group	ha group ID	INT	1 – 31	
ha_group_dynamic_weight	dynamic weight of the dns proxy in the ha group	INT	1 - 255	
vport_list	Virtual port list statement	LIST		
vport	Vport items. Has the following parameters	OBJ		
port_number	virtual port number	INT	1 – 65535	
service_group	service group of this virtual port	String	1 - 31	
status	enabled(1) or disabled(0)	INT	0 - 1	
connection_limit	connection limit, 0 means no limit	INT	0 - 8000000	
over_connection_limit_action	over connection limit action, drop(0), reset(1)	INT	0 - 1	
source_nat_pool	source nat pool name	String	1 - 31	
aflex	aFlex(name) of this virtual port	String	1 - 31	
udp_template	udp template of this virtual port	String	1 - 31	

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
- **HTTP Action:**
POST

CLI configuration before aXAPI call:

```
!
ha group 31 priority 255
!
slb virtual-server AAA 8.2.3.19
    ha-group 31
    ha-dynamic 1
    port 12345 udp
        name _8.2.3.19_UDP_12345
        gslb-enable
    port 12346 udp
        name _8.2.3.19_UDP_12346
        gslb-enable
    port 12347 udp
        name _8.2.3.19_UDP_12347
        gslb-enable
!
```

1. HTTP Request in URL Format:

```
session_id=xxxxxxxxxxxxxxxxxxxxxx&method=gslb.dns_proxy.update&gslb_vserver=name%03A
AA%02ip_address%038.2.3.19%02status%030%02ha_group%0331%02ha_group_dynamic_weight%030%02v
port_list&vport_list=P1%02P2%02P3&P1=port_number%0312345%02status%030&P2=port_number%0312
346%02status%030&P3=port_number%0312347%02status%030&format=url
```

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in JSON Format:

```
session_id=xxxxxxxxxxxxxxxxxxxxxx&method=gslb.dns_proxy.update&format=json
```

Request POST data (JSON Format):

```
{
  "gslb_vserver": {
    "name": "AAA",
    "ip_address": "8.2.3.19",
    "status": 0,
    "ha_group": 31,
    "ha_group_dynamic_weight": 1,
    "vport_list": [
      {
        "port_number": 12347,
        "service_group": "",
        "status": 0,
        "ha_connection_mirror": 0,
        "connection_limit": 8000000,
        "over_connection_limit_action": 0,
        "source_nat_pool": "",
        "aflex": "",
        "udp_template": ""
      }
    ]
  }
}
```

```

    "port_number":12346,
    "service_group":"",
    "status":0,
    "ha_connection_mirror":0,
    "connection_limit":8000000,
    "over_connection_limit_action":0,
    "source_nat_pool":"",
    "aflex":"",
    "udp_template":""
},
{
    "port_number":12345,
    "service_group":"",
    "status":0,
    "ha_connection_mirror":0,
    "connection_limit":8000000,
    "over_connection_limit_action":0,
    "source_nat_pool":"",
    "aflex":"",
    "udp_template":""
}
]
}

```

CLI configuration after aXAPI call:

```

!
ha group 31 priority 255
!
slb virtual-server AAA 8.2.3.19
    disable
    ha-group 31
    port 12345  udp
        name _8.2.3.19_UDP_12345
        disable
        gslb-enable
    port 12346  udp
        name _8.2.3.19_UDP_12346
        disable
        gslb-enable
    port 12347  udp
        name _8.2.3.19_UDP_12347
        disable
        gslb-enable
!

```

Response as the HTTP body:

```

{
    "response":
    {
        "status" : "OK"
    }
}

```

Menus Privilege

Config Mode >> Service >> GSLB >> DNS Proxy

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.1.5 “gslb.dns_proxy.delete” Method

This method is used to delete a GSLB DNS proxy from the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	gslb.dns_proxy.delete	String		
name (*)	gslb dns proxy name	String		

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
- **HTTP Action:**
POST

CLI configuration before aXAPI call:

```
!
slb virtual-server test01 4.4.4.4
  port 888 udp
    name _4.4.4.4_UDP_888
  gslb-enable
!
```

1. **HTTP Request in URL Format:**

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=gslb.dns_proxy.delete&name=test01&format=ur
```

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. **HTTP Request in JSON Format:**

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=gslb.dns_proxy.delete&format=json
```

Request POST data (JSON Format):

```
{
  "name": "test01"
}
```

CLI configuration after aXAPI call:

```
!
!
```

Response as the HTTP body:

```
{
  "response":
```

```

    {
        "status" : "OK"
    }
}

```

Menus Privilege

Config Mode >> Service >> GSLB >> DNS Proxy

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.1.6 “gslb.dns_proxy.vport.create” Method

This method is used to add a virtual port to the virtual port list of the specified GSLB DNS proxy.

Parameters

Parameter name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	gslb.dns_proxy.vport.create	String		
name (*)	virtual server template name	String	1 - 31	
vport(*)	Vport items. Has the following parameters	OBJ		
port_number(*)	virtual port number	INT	1 – 65535	
service_group	service group of this virtual port	String	1 - 31	
status	enabled(1) or disabled(0)	INT	0 - 1	
connection_limit	connection limit, 0 means no limit	INT	0 - 8000000	
over_connection_limit_action	over connection limit action, drop(0), reset(1)	INT	0 - 1	
source_nat_pool	source nat pool name	String	1 - 31	
aFlex	aFlex(name) of this virtual port	String	1 - 31	
udp_template	udp template of this virtual port	String	1 - 31	

Notes: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```

!
ha group 31 priority 255
!
slb virtual-server AAA 8.2.3.19
    disable
    ha-group 31
    port 12345 udp
        name _8.2.3.19_UDP_12345
        disable
        gslb-enable
    port 12346 udp
        name _8.2.3.19_UDP_12346
        disable
        gslb-enable
    port 12347 udp

```

```

! name _8.2.3.19_UDP_12347
  disable
  gslb-enable
!
```

1. HTTP Request in URL Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=gslb.dns_proxy.vport.create&name=AAA&vport=
port_number%0365534%02connection_limit%03100&format=url
```

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in JSON Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=gslb.dns_proxy.vport.create&format=json
```

Request POST data (JSON Format):

```
{
  "name": "AAA",
  "vport": {
    {
      "port_number": 65534,
      "service_group": "",
      "status": 1,
      "ha_connection_mirror": 0,
      "connection_limit": 100,
      "over_connection_limit_action": 0,
      "source_nat_pool": "",
      "aflex": "",
      "udp_template": ""
    }
  }
}
```

CLI configuration after aXAPI call:

```
!
ha group 31 priority 255
!
slb virtual-server AAA 8.2.3.19
  disable
  ha-group 31
  port 12345  udp
    name _8.2.3.19_UDP_12345
    disable
    gslb-enable
  port 12346  udp
    name _8.2.3.19_UDP_12346
    disable
    gslb-enable
  port 12347  udp
    name _8.2.3.19_UDP_12347
    disable
    gslb-enable
  port 65534  udp
    name _8.2.3.19_UDP_65534
    disable
    conn-limit 100 no-logging
!
```

Response as the HTTP body:

```
{
    "response":
    {
        "status" : "OK"
    }
}
```

Privileges

Config Mode >> Service >> GSLB >> DNS Proxy

Error Codes

TBD

9.1.7 “gslb.dns_proxy.vport.update” Method

This method is used to update one or more of the parameters for a virtual port to the virtual port list of the specified GSLB DNS proxy.

Parameters

Parameter name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	gslb.dns_proxy.vport.update	String		
name (*)	virtual server template name	String	1 - 31	
vport(*)	Vport items. Has the following parameters	OBJ		
port_number(*)	virtual port number	INT	1 - 65535	
service_group	service group of this virtual port	String	1 - 31	
status	enabled(1) or disabled(0)	INT	0 - 1	
connection_limit	connection limit, 0 means no limit	INT	0 - 8000000	
over_connection_limit_action	over connection limit action, drop(0), reset(1)	INT	0 - 1	
source_nat_pool	source nat pool name	String	1 - 31	
aflex	aFlex(name) of this virtual port	String	1 - 31	
udp_template	udp template of this virtual port	String	1 - 31	

Notes: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
ha group 31 priority 255
!
slb virtual-server AAA 8.2.3.19
  disable
  ha-group 31
  port 12345  udp
    name _8.2.3.19_UDP_12345
    disable
    gslb-enable
  port 12346  udp
    name _8.2.3.19_UDP_12346
    disable
    gslb-enable
```

```

port 12347  udp
    name _8.2.3.19_UDP_12347
    disable
    gslb-enable
port 65534  udp
    name _8.2.3.19_UDP_65534
    gslb-enable
    conn-limit 100 no-logging
!
```

1. HTTP Request in URL Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=gslb.dns_proxy.vport.update&name=AAA&vport=
port_number%0365534%02connection_limit%0380&format=url
```

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in JSON Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=gslb.dns_proxy.vport.update&format=json
```

Request POST data (JSON Format):

```
{
  "name": "AAA",
  "vport": {
    {
      "port_number": 65534,
      "service_group": "",
      "status": 1,
      "ha_connection_mirror": 0,
      "connection_limit": 80,
      "over_connection_limit_action": 0,
      "source_nat_pool": "",
      "aflex": "",
      "udp_template": ""
    }
  }
}
```

CLI configuration after aXAPI call:

```
!
ha group 31 priority 255
!
slb virtual-server AAA 8.2.3.19
  disable
  ha-group 31
  port 12345  udp
    name _8.2.3.19_UDP_12345
    disable
    gslb-enable
  port 12346  udp
    name _8.2.3.19_UDP_12346
    disable
    gslb-enable
  port 12347  udp
    name _8.2.3.19_UDP_12347
    disable
    gslb-enable
port 65534  udp
  name _8.2.3.19_UDP_65534
  gslb-enable
```

```

conn-limit 80 no-logging
!

```

Response as the HTTP body:

```
{
    "response": [
        {
            "status" : "OK"
        }
    ]
}
```

Privileges

Config Mode >> Service >> GSLB >> DNS Proxy

Error Codes

TBD

9.1.8 “gslb.dns_proxy.vport.delete” Method

This method is used to delete a virtual port from the virtual port list of a user-specified GSLB DNS proxy.

Parameters

Parameter name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	gslb.dns_proxy.vport.delete	String		
name (*)	virtual server template name	String	1 - 31	
vport(*)	Vport items. Has the following parameters	OBJ		
port_number(*)	virtual port number	INT	1 – 65535	

Notes: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```

!
slb virtual-server AAA 8.2.3.19
    disable
    ha-group 31
    port 12345 udp
        name _8.2.3.19_UDP_12345
        disable
        gslb-enable
    port 12346 udp
        name _8.2.3.19_UDP_12346
        disable
        gslb-enable
    port 12347 udp
        name _8.2.3.19_UDP_12347
        disable
        gslb-enable
    port 65534 udp

```

```

name _8.2.3.19_UDP_65534
gslb-enable
conn-limit 80 no-logging
port 8080 udp
  name _8.2.3.19_UDP_8080
  gslb-enable
!

```

1. HTTP Request in URL Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=gslb.dns_proxy.vport.delete&name=AAA&vport=
port_number%038080&format=url
```

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in JSON Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=gslb.dns_proxy.vport.delete&format=json
```

Request POST data (JSON Format):

```
{
  "name": "AAA",
  "vport": {
    "port_number": 8080,
  }
}
```

CLI configuration after aXAPI call:

```
!
slb virtual-server AAA 8.2.3.19
  disable
  ha-group 31
  port 12345 udp
    name _8.2.3.19_UDP_12345
    disable
    gslb-enable
  port 12346 udp
    name _8.2.3.19_UDP_12346
    disable
    gslb-enable
  port 12347 udp
    name _8.2.3.19_UDP_12347
    disable
    gslb-enable
  port 65534 udp
    name _8.2.3.19_UDP_65534
    gslb-enable
    conn-limit 80 no-logging
!
```

Response as the HTTP body:

```
{
  "response": {
  }
```

```

        "status" : "OK"
    }
}
```

Privileges

Config Mode >> Service >> GSLB >> DNS Proxy

Error Codes

TBD

9.2 Policy

9.2.1 “gslb.policy.getAll” Method

This method is used to get configuration information for all GSLB policy objects configured on the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	gslb.policy.getAll	String		

Note: (*) parameter is required in the API.

Response Fields

policy_list	XML tag for the gslb policy list		
policy	XML tag for the gslb policy object		
name	GSLB policy name		
metric	XML tag for the metric list		
session_capacity	XML tag for session capacity		
enabled	the status of session capacity		
threshold	threshold of session capacity		
active_rdt	XML tag for active RDT		
enabled	the status of active RDT		
samples	the number of samples		
difference	difference		
tolerance	tolerance		
time_out	time out		
skip	skip		
single_shot	single shot		
connection_load	XML tag for connection load		
enabled	the status of connection load		
limit	connection limit		
samples_num	sample number		
samples_interval	sample interval		
num_session	number session		
enabled	status of number session		
tolerance	tolerance		
geo_graphic	XML tag of geographic		
enabled	status of geographic option		
admin_ip	XML tag if administrative IP option		

	enabled	status of ordered IP option
weighted_site	enabled	XML tag of weighted site option
	enabled	status of weighted site option
active_servers	enabled	XML tag of active servers
	enabled	status of active servers
weighted_ip	enabled	XML tag of weighted IP option
	enabled	status of weighted IP option
bandwidth_cost	enabled	XML tag of bandwidth option
	enabled	status of bandwidth
health_check	enabled	XML tag of health check option
	enabled	status of health check option
admin_perference	enabled	XML tag of administrator preference
	enabled	status of administrator preference option
least_response	enabled	XML tag of least connection option
	enabled	status of least response option
round_robin	enabled	XML tag of round robin option
	enabled	status of round robin
dns_options		XML tag of DNS options
	action	action
	active_only	action only
	best_only	best only
	dns_cache	XML tag of DNS cache
	enabled	status of DNS cache
	dns_cache_aging_time	DNS cache aging time
cname_detect		CName detected
external_ip		external IP
ip_replace		IP replace
geo_location_alias		geographic location alias
geo_location_action		geographic location action
geo_location_policy		geographic location policy
mx_additional		MX additional
server_mode		XML tag of server mode
	enabled	status of server mode
	authoritative_mode	authoritative mode
	full_server_list	full server list
	server_mx	server MX record
	server_mx_additional	server MX additional record
	server_ns	server NS record
	server_auto_ns	server auto NS
	server_ptr	server PTR
	server_auto_ptr	server auto PTR
sticky		XML tag of sticky
	enabled	status of sticky
	sticky_dns_client_ip_mask_len	sticky DNS client IP mask length
ttl		XML tag of TTL
	enabled	status of TTL
	ttl_time_live	TTL time live
geo_location		XML tag of geographic location
	geo_location_match_first	geographic location match first
	geo_location_overlap	geographic location overlap

Example Response

Request as:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
- **HTTP Action:**
POST

CLI configuration before aXAPI call:

```
!
gslb policy default
gslb policy policy1
    metric-order geographic health-check weighted-ip weighted-site active-servers
        capacity active-rdt connection-load num-session admin-preference bw-cost
            least-response admin-ip
!
```

1. HTTP Request in URL Format:

```
session_id=xxxxxxxxxxxxxxxxxxxxx&method=gslb.policy.getAll&format=url
```

Response as the HTTP body (URL Format):

```
<response status="ok">
    <policy_list>
        <policy>
            <name>default</name>
            <metric>
                <session_capacity>
                    <enabled>0</enabled>
                    <threshold>90</threshold>
                </session_capacity>
                <active_rdt>
                    <enabled>0</enabled>
                    <samples>5</samples>
                    <difference>0</difference>
                    <tolerance>10</tolerance>
                    <timeout>3</timeout>
                    <skip>3</skip>
                    <single_shot>0</single_shot>
                </active_rdt>
                <connection_load>
                    <enabled>0</enabled>
                    <limit>0</limit>
                    <samples_num>5</samples_num>
                    <samples_interval>5</samples_interval>
                </connection_load>
                <num_session>
                    <enabled>0</enabled>
                    <tolerance>10</tolerance>
                </num_session>
                <geo_graphic>
                    <enabled>1</enabled>
                </geo_graphic>
                <admin_ip>
                    <enabled>0</enabled>
                </admin_ip>
                <weighted_site>
                    <enabled>0</enabled>
                </weighted_site>
                <active_servers>
                    <enabled>0</enabled>
                </active_servers>
                <weighted_ip>
                    <enabled>0</enabled>
                </weighted_ip>
                <bandwidth_cost>
                    <enabled>0</enabled>
                </bandwidth_cost>
                <health_check>
                    <enabled>1</enabled>
                </health_check>
            </metric>
        </policy>
    </policy_list>
</response>
```

```

        </health_check>
        <admin_preference>
            <enabled>0</enabled>
        </admin_preference>
        <least_response>
            <enabled>0</enabled>
        </least_response>
        <round_robin>
            <enabled>1</enabled>
        </round_robin>
    </metric>
    <dns_options>
        <action>0</action>
        <active_only>0</active_only>
        <best_only>0</best_only>
        <dns_cache>
            <enabled>0</enabled>
            <dns_cache_aging_time>0</dns_cache_aging_time>
        </dns_cache>
        <cname_detect>1</cname_detect>
        <external_ip>1</external_ip>
        <ip_replace>0</ip_replace>
        <geo_location_alias>0</geo_location_alias>
        <geo_location_action>0</geo_location_action>
        <geo_location_policy>0</geo_location_policy>
        <mx_additional>0</mx_additional>
        <server_mode>
            <enabled>0</enabled>
            <authoritative_mode>0</authoritative_mode>
            <full_server_list>0</full_server_list>
            <server_mx>0</server_mx>
            <server_mx_additional>0</server_mx_additional>
            <server_ns>0</server_ns>
            <server_auto_ns>0</server_auto_ns>
            <server_ptr>0</server_ptr>
            <server_auto_ptr>0</server_auto_ptr>
        </server_mode>
        <sticky>
            <enabled>0</enabled>
            <sticky_dns_client_ip_mask_len>32</sticky_dns_client_ip_mask_len>
        </sticky>
        <ttl>
            <enabled>1</enabled>
            <ttl_time_live>10</ttl_time_live>
        </ttl>
    </dns_options>
    <geo_location>
        <geo_location_match_first>0</geo_location_match_first>
        <geo_location_overlap>0</geo_location_overlap>
    </geo_location>
</policy>
<policy>
    <name>policy1</name>
    <metric>
        <session_capacity>
            <enabled>0</enabled>
            <threshold>90</threshold>
        </session_capacity>
        <active_rdt>
            <enabled>0</enabled>
            <samples>5</samples>
            <difference>0</difference>
            <tolerance>10</tolerance>
            <timeout>3</timeout>
            <skip>3</skip>
            <single_shot>0</single_shot>
        </active_rdt>
        <connection_load>
            <enabled>0</enabled>
            <limit>0</limit>
        </connection_load>
    </metric>

```

```

        <samples_num>5</samples_num>
        <samples_interval>5</samples_interval>
    </connection_load>
    <num_session>
        <enabled>0</enabled>
        <tolerance>10</tolerance>
    </num_session>
    <geo_graphic>
        <enabled>1</enabled>
    </geo_graphic>
    <admin_ip>
        <enabled>0</enabled>
    </admin_ip>
    <weighted_site>
        <enabled>0</enabled>
    </weighted_site>
    <active_servers>
        <enabled>0</enabled>
    </active_servers>
    <weighted_ip>
        <enabled>0</enabled>
    </weighted_ip>
    <bandwidth_cost>
        <enabled>0</enabled>
    </bandwidth_cost>
    <health_check>
        <enabled>1</enabled>
    </health_check>
    <admin_perference>
        <enabled>0</enabled>
    </admin_perference>
    <least_response>
        <enabled>0</enabled>
    </least_response>
    <round_robin>
        <enabled>1</enabled>
    </round_robin>
</metric>
<dns_options>
    <action>0</action>
    <active_only>0</active_only>
    <best_only>0</best_only>
    <dns_cache>
        <enabled>0</enabled>
        <dns_cache_aging_time>0</dns_cache_aging_time>
    </dns_cache>
    <cname_detect>1</cname_detect>
    <external_ip>1</external_ip>
    <ip_replace>0</ip_replace>
    <geo_location_alias>0</geo_location_alias>
    <geo_location_action>0</geo_location_action>
    <geo_location_policy>0</geo_location_policy>
    <mx_additional>0</mx_additional>
    <server_mode>
        <enabled>0</enabled>
        <authoritative_mode>0</authoritative_mode>
        <full_server_list>0</full_server_list>
        <server_mx>0</server_mx>
        <server_mx_additional>0</server_mx_additional>
        <server_ns>0</server_ns>
        <server_auto_ns>0</server_auto_ns>
        <server_ptr>0</server_ptr>
        <server_auto_ptr>0</server_auto_ptr>
    </server_mode>
    <sticky>
        <enabled>0</enabled>
        <sticky_dns_client_ip_mask_len>32</sticky_dns_client_ip_mask_len>
    </sticky>
    <ttl>
        <enabled>1</enabled>

```

```

                <ttl_time_live>10</ttl_time_live>
            </ttl>
        </dns_options>
        <geo_location>
            <geo_location_match_first>0</geo_location_match_first>
            <geo_location_overlap>0</geo_location_overlap>
        </geo_location>
    </policy>
</policy_list>
</response>

```

2. HTTP Request in JSON Format:

session_id=xxxxxxxxxxxxxxxxxxxx&method=gslb.policy.getAll&format=json

Response as the HTTP body (JSON Format):

```
{
    "policy_list": [
        {
            "name": "default",
            "metric": {
                "session_capacity": {
                    "sequence_num": 3,
                    "enabled": 0,
                    "threshold": 90
                },
                "active_rdt": {
                    "sequence_num": 6,
                    "enabled": 0,
                    "samples": 5,
                    "difference": 0,
                    "tolerance": 10,
                    "timeout": 3,
                    "skip": 3,
                    "single_shot": 0
                },
                "connection_load": {
                    "sequence_num": 8,
                    "enabled": 0,
                    "limit": 0,
                    "samples_num": 5,
                    "samples_interval": 5
                },
                "num_session": {
                    "sequence_num": 9,
                    "enabled": 0,
                    "tolerance": 10
                },
                "geo_graphic": {
                    "sequence_num": 7,
                    "enabled": 1
                },
                "admin_ip": {
                    "sequence_num": 16,
                    "enabled": 0
                },
                "weighted_site": {
                    "sequence_num": 2,
                    "enabled": 0
                }
            }
        }
    ]
}
```

```

"active_servers":
{
    "sequence_num":4,
    "enabled":0
},
"weighted_ip":
{
    "sequence_num":1,
    "enabled":0
},
"bandwidth_cost":
{
    "sequence_num":13,
    "enabled":0
},
"health_check":
{
    "sequence_num":0,
    "enabled":1
},
"admin_preference":
{
    "sequence_num":11,
    "enabled":0
},
"least_response":
{
    "sequence_num":15,
    "enabled":0
},
"round_robin":
{
    "sequence_num":17,
    "enabled":1
}
},
"dns_options":
{
    "action":0,
    "active_only":0,
    "best_only":0,
    "dns_cache":
    {
        "enabled":0,
        "dns_cache_aging_time":0
    },
    "cname_detect":1,
    "external_ip":1,
    "ip_replace":0,
    "geo_location_alias":0,
    "geo_location_action":0,
    "geo_location_policy":0,
    "mx_additional":0,
    "server_mode":
    {
        "enabled":0,
        "authoritative_mode":0,
        "full_server_list":0,
        "server_mx":0,
        "server_mx_additional":0,
        "server_ns":0,
        "server_auto_ns":0,
        "server_ptr":0,
        "server_auto_ptr":0
    },
    "sticky":
    {
        "enabled":0,
        "sticky_dns_client_ip_mask_len":32,
        "sticky_aging_time":5
    }
},

```

```

        "ttl":
        {
            "enabled":1,
            "ttl_time_live":10
        }
    },
    "geo_location":
    {
        "geo_location_match_first":0,
        "geo_location_overlap":0
    }
},
{
    "name":"policy1",
    "metric":
    {
        "session_capacity":
        {
            "sequence_num":5,
            "enabled":0,
            "threshold":90
        },
        "active_rdt":
        {
            "sequence_num":6,
            "enabled":0,
            "samples":5,
            "difference":0,
            "tolerance":10,
            "timeout":3,
            "skip":3,
            "single_shot":0
        },
        "connection_load":
        {
            "sequence_num":7,
            "enabled":0,
            "limit":0,
            "samples_num":5,
            "samples_interval":5
        },
        "num_session":
        {
            "sequence_num":8,
            "enabled":0,
            "tolerance":10
        },
        "geo_graphic":
        {
            "sequence_num":0,
            "enabled":1
        },
        "admin_ip":
        {
            "sequence_num":12,
            "enabled":0
        },
        "weighted_site":
        {
            "sequence_num":3,
            "enabled":0
        },
        "active_servers":
        {
            "sequence_num":4,
            "enabled":0
        },
        "weighted_ip":
        {
            "sequence_num":2,
            "enabled":0
        }
    }
}

```

```

},
"bandwidth_cost":
{
    "sequence_num":10,
    "enabled":0
},
"health_check":
{
    "sequence_num":1,
    "enabled":1
},
"admin_preference":
{
    "sequence_num":9,
    "enabled":0
},
"least_response":
{
    "sequence_num":11,
    "enabled":0
},
"round_robin":
{
    "sequence_num":17,
    "enabled":1
}
},
"dns_options":
{
    "action":0,
    "active_only":0,
    "best_only":0,
    "dns_cache":
    {
        "enabled":0,
        "dns_cache_aging_time":0
    },
    "cname_detect":1,
    "external_ip":1,
    "ip_replace":0,
    "geo_location_alias":0,
    "geo_location_action":0,
    "geo_location_policy":0,
    "mx_additional":0,
    "server_mode":
    {
        "enabled":0,
        "authoritative_mode":0,
        "full_server_list":0,
        "server_mx":0,
        "server_mx_additional":0,
        "server_ns":0,
        "server_auto_ns":0,
        "server_ptr":0,
        "server_auto_ptr":0
    },
    "sticky":
    {
        "enabled":0,
        "sticky_dns_client_ip_mask_len":32,
        "sticky_aging_time":5
    }
},
"ttl":
{
    "enabled":1,
    "ttl_time_live":10
}
},
"geo_location":
{
    "geo location match first":0,

```

```

        "geo_location_overlap":0
    }
]
}

```

Menus Privilege

Config Mode >> Service >> GSLB >> Policy

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.2.2 “gslb.policy.search” Method

This method is used to search GSLB policy objects configured on the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	gslb.policy.search	String		
name(*)	the name of the gslb policy	String		

Note: (*) parameter is required in the API.

Response Fields

policy_list	XML tag for the gslb policy list		
policy	XML tag for the gslb policy object		
name	GSLB policy name		
metric	XML tag for the metric list		
session_capacity	XML tag for session capacity		
enabled	the status of session capacity		
threshold	threshold of session capacity		
active_rdt			
enabled	the status of active RDT		
samples	the number of samples		
difference	difference		
tolerance	tolerance		
time_out	time out		
skip	skip		
single_shot	single shot		
connection_load			
enabled	the status of connection load		
limit	connection limit		
samples_num	sample number		
samples_interval	sample interval		
num_session	number session		
enabled	status of number session		
tolerance	tolerance		
geo_graphic	XML tag of geographic		
enabled	status of geographic option		
admin_ip	XML tag if administrative IP option		

	enabled	status of ordered IP option
weighted_site	enabled	XML tag of weighted site option
	enabled	status of weighted site option
active_servers	enabled	XML tag of active servers
	enabled	status of active servers
weighted_ip	enabled	XML tag of weighted IP option
	enabled	status of weighted IP option
bandwidth_cost	enabled	XML tag of bandwidth option
	enabled	status of bandwidth
health_check	enabled	XML tag of health check option
	enabled	status of health check option
admin_perference	enabled	XML tag of administrator preference
	enabled	status of administrator preference option
least_response	enabled	XML tag of least connection option
	enabled	status of least response option
round_robin	enabled	XML tag of round robin option
	enabled	status of round robin
dns_options		XML tag of DNS options
	action	action
	active_only	action only
	best_only	best only
	dns_cache	XML tag of DNS cache
	enabled	status of DNS cache
	dns_cache_aging_time	DNS cache aging time
cname_detect		CName detected
external_ip		external IP
ip_replace		IP replace
geo_location_alias		geographic location alias
geo_location_action		geographic location action
geo_location_policy		geographic location policy
mx_additional		MX additional
server_mode		XML tag of server mode
	enabled	status of server mode
	authoritative_mode	authoritative mode
	full_server_list	full server list
	server_mx	server MX record
	server_mx_additional	server MX additional record
	server_ns	server NS record
	server_auto_ns	server auto NS
	server_ptr	server PTR
	server_auto_ptr	server auto PTR
sticky		XML tag of sticky
	enabled	status of sticky
	sticky_dns_client_ip_mask_len	sticky DNS client IP mask length
ttl		XML tag of TTL
	enabled	status of TTL
	ttl_time_live	TTL time live
geo_location		XML tag of geographic location
	geo_location_match_first	geographic location match first
	geo_location_overlap	geographic location overlap

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
gslb policy policy1
    metric-order geographic health-check weighted-ip weighted-site active-servers
        capacity active-rdt connection-load num-session admin-preference bw-cost
            least-response admin-ip
!
!
```

1. HTTP Request in URL Format:

```
session_id=xxxxxxxxxxxxxxxxxxxxxx&method=gslb.policy.search&name=policy1&format=url
```

Response as the HTTP body (URL Format):

```
<response status="ok">
<policy>
    <name>policy1</name>
    <metric>
        <session_capacity>
            <enabled>0</enabled>
            <threshold>90</threshold>
        </session_capacity>
        <active_rdt>
            <enabled>0</enabled>
            <samples>5</samples>
            <difference>0</difference>
            <tolerance>10</tolerance>
            <timeout>3</timeout>
            <skip>3</skip>
            <single_shot>0</single_shot>
        </active_rdt>
        <connection_load>
            <enabled>0</enabled>
            <limit>0</limit>
            <samples_num>5</samples_num>
            <samples_interval>5</samples_interval>
        </connection_load>
        <num_session>
            <enabled>0</enabled>
            <tolerance>10</tolerance>
        </num_session>
        <geo_graphic>
            <enabled>1</enabled>
        </geo_graphic>
        <admin_ip>
            <enabled>0</enabled>
        </admin_ip>
        <weighted_site>
            <enabled>0</enabled>
        </weighted_site>
        <active_servers>
            <enabled>0</enabled>
        </active_servers>
        <weighted_ip>
            <enabled>0</enabled>
        </weighted_ip>
        <bandwidth_cost>
            <enabled>0</enabled>
        </bandwidth_cost>
    <health_check>
```

```

<enabled>1</enabled>
</health_check>
<admin_preference>
<enabled>0</enabled>
</admin_preference>
<least_response>
<enabled>0</enabled>
</least_response>
<round_robin>
<enabled>1</enabled>
</round_robin>
</metric>
<dns_options>
<action>0</action>
<active_only>0</active_only>
<best_only>0</best_only>
<dns_cache>
<enabled>0</enabled>
<dns_cache_aging_time>0</dns_cache_aging_time>
</dns_cache>
<cname_detect>1</cname_detect>
<external_ip>1</external_ip>
<ip_replace>0</ip_replace>
<geo_location_alias>0</geo_location_alias>
<geo_location_action>0</geo_location_action>
<geo_location_policy>0</geo_location_policy>
<mx_additional>0</mx_additional>
<server_mode>
<enabled>0</enabled>
<authoritative_mode>0</authoritative_mode>
<full_server_list>0</full_server_list>
<server_mx>0</server_mx>
<server_mx_additional>0</server_mx_additional>
<server_ns>0</server_ns>
<server_auto_ns>0</server_auto_ns>
<server_ptr>0</server_ptr>
<server_auto_ptr>0</server_auto_ptr>
</server_mode>
<sticky>
<enabled>0</enabled>
<sticky_dns_client_ip_mask_len>32</sticky_dns_client_ip_mask_len>
</sticky>
<ttl>
<enabled>1</enabled>
<ttl_time_live>10</ttl_time_live>
</ttl>
</dns_options>
<geo_location>
<geo_location_match_first>0</geo_location_match_first>
<geo_location_overlap>0</geo_location_overlap>
</geo_location>
</policy>
</response>

```

2. HTTP Request in JSON Format:

session_id=xxxxxxxxxxxxxxxxxxxxxx&method=gslb.policy.search&format=json

Request POST data (JSON Format):

```
{
  "name": "policy1"
}
```

Response as the HTTP body:

```
{
```

```

"policy":
{
    "name":"policy1",
    "metric":
    {
        "session_capacity":
        {
            "sequence_num":5,
            "enabled":0,
            "threshold":90
        },
        "active_rdt":
        {
            "sequence_num":6,
            "enabled":0,
            "samples":5,
            "difference":0,
            "tolerance":10,
            "timeout":3,
            "skip":3,
            "single_shot":0
        },
        "connection_load":
        {
            "sequence_num":7,
            "enabled":0,
            "limit":0,
            "samples_num":5,
            "samples_interval":5
        },
        "num_session":
        {
            "sequence_num":8,
            "enabled":0,
            "tolerance":10
        },
        "geo_graphic":
        {
            "sequence_num":0,
            "enabled":1
        },
        "admin_ip":
        {
            "sequence_num":12,
            "enabled":0
        },
        "weighted_site":
        {
            "sequence_num":3,
            "enabled":0
        },
        "active_servers":
        {
            "sequence_num":4,
            "enabled":0
        },
        "weighted_ip":
        {
            "sequence_num":2,
            "enabled":0
        },
        "bandwidth_cost":
        {
            "sequence_num":10,
            "enabled":0
        },
        "health_check":
        {
            "sequence_num":1,
            "enabled":1
        }
    }
}

```

```

"admin_preference":
{
    "sequence_num":9,
    "enabled":0
},
"least_response":
{
    "sequence_num":11,
    "enabled":0
},
"round_robin":
{
    "sequence_num":17,
    "enabled":1
}
},
"dns_options":
{
    "action":0,
    "active_only":0,
    "best_only":0,
    "dns_cache":
    {
        "enabled":0,
        "dns_cache_aging_time":0
    },
    "cname_detect":1,
    "external_ip":1,
    "ip_replace":0,
    "geo_location_alias":0,
    "geo_location_action":0,
    "geo_location_policy":0,
    "mx_additional":0,
    "server_mode":
    {
        "enabled":0,
        "authoritative_mode":0,
        "full_server_list":0,
        "server_mx":0,
        "server_mx_additional":0,
        "server_ns":0,
        "server_auto_ns":0,
        "server_ptr":0,
        "server_auto_ptr":0
    },
    "sticky":
    {
        "enabled":0,
        "sticky_dns_client_ip_mask_len":32,
        "sticky_aging_time":5
    }
},
"ttl":
{
    "enabled":1,
    "ttl_time_live":10
}
},
"geo_location":
{
    "geo_location_match_first":0,
    "geo_location_overlap":0
}
}
}

```

Menus Privilege

Config Mode >> Service >> GSLB >> Policy

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.2.3 “gslb.policy.create” Method

This method is used to create a GSLB policy.

Parameters

Parameter name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	gslb.policy.create	String		
name(*)	gslb policy name	String	1 - 31	
metric	Metric object	Object		
 session_capacity	Session capacity object of metric object	Object		
 enabled	Status of session capacity, enabled(1) or disabled (0)	Integer	0 - 1	0
 threshold	Threshold of session capacity (%)	Integer	0 - 100	90
 active_rdt	Active RDT object of metric object	Object		
 enabled	Status of active RDT	Integer	0 - 1	0
 samples	The number of samples of active RDT	Integer	1 - 8	5
 difference	Difference of active RDT	Integer	0 - 16383	0
 tolerance	Tolerance of active RDT	Integer	0 - 100	0
 timeout	Timeout of active RDT	Integer	1 - 255	3
 skip	Skip of active RDT	Integer	1 - 31	3
 single_shot	Single-shot of active RDT, enabled (1) or disabled (0)	Integer	0 - 1	0
 connection_load	Connection load object of metric object			
 enabled	Status of connection load object, enabled (1) or disabled (0)	Integer	0 - 1	0
 limit	Connection limit of connection load object	Integer	0 - 999999999	3
 samples_num	The number of samples of connection load	Integer	1 - 8	5
 samples_interval	Interval of samples of connection load	Integer	1 - 60	5
 num_session	Number session object of metric object			
 enabled	Status of number session, enabled (1) or disabled (0)	Integer	0 - 1	0
 tolerance	Tolerance of number session	Integer	0 - 100	10
 geo_graphic	Geographic object of metric object	Object		
 enabled	Status of geographic, enabled (1) or disable (0)	Integer	0 - 1	1
 admin_ip	Administrative IP of metric object	Object		
 enabled	Status of ordered IP, enabled (1) or disabled (0)	Integer	0 - 1	0
 weighted_site	Weighted site object of metric object	Object		
 enabled	Status of weighted site object, enabled (1) or disabled (0)	Integer	0 - 1	0
 active_servers	Active server object of metric object	Object		
 enabled	Status of active server object, enabled (1) or disabled(0)	Integer	0 - 1	0
 weighted_ip	Weighted IP object of metric object	Object		
 enabled	Status of weight IP, enabled (1) or disabled (0)	Integer	0 - 1	0
 bandwidth_cost	Bandwidth cost object of metric object	Object		
 enabled	Status of bandwidth cost object, enabled (1) or disabled (0)	Integer	0 - 1	0
 health_check	Health check object of metric object	Object		
 enabled	Status of health check object, enabled (1) or disabled (0)	integer	0 - 1	1
 admin_preference	Admin preference object of metric object	Object		
 enabled	Status of administrator preference	Integer	0 - 1	0
 least_response	Least response object of metric object	Object		
 enabled	Status of least response object, enabled (1) or disabled (0)	Integer	0 - 1	0
 round_robin	Round robin object of metric object	Object		
 enabled	Status of round robin object	Integer	0 - 1	1
 dns_options	DNS option object			
 action	Action of DNS option object, enabled (1) or disabled (0)	Integer	0 - 1	0
 active_only	Active only option of DNS options object, enabled (1)	Integer	0 - 1	0

	or disabled (0)			
best_only	Best only object of DNS option object	Integer	0 – 1	0
dns_cache	DNS cache status object of DNS option	object		
enabled	Status of DNS cache object, enabled (1) or disabled (0)	Integer	0 – 1	0
dns_cache_aging_time	DNS cache aging time of DNS cache object, only when dns_cache is enabled. Value 0 means 'by TTL'.	Integer	0 - 1000000000 0	0
cname_detect	CName detect option of DNS option object, enabled (1) or disabled (0)	Integer	0 - 1	1
external_ip	External IP option of DNS option object, enabled (1) or disabled (0)	Integer	0 - 1	1
ip_replace	IP replace option of DNS option object, enabled (1) or disabled (0)	Integer	0 – 1	0
geo_location_alias	Geographic location alias option of DNS option object, enabled (1) or disabled (0)	Integer	0 - 1	0
geo_location_action	Geographic location action option of DNS option object, enabled (1) or disabled (0)	Integer	0 - 1	0
geo_location_policy	Geographic location policy option of DNS option object , enabled (1) or disabled (0)	Integer	0 – 1	0
mx_additional	MX record additional option of DNS option object, enabled (1) or disabled (0)	Integer	0 – 1	0
server_mode	Server mode object of DNS option	Object		
enabled	Status of server mode , enabled (1) or disabled (0)	Integer	0 - 1	0
authoritative_mode	Authoritative mode of server mode object, enabled (1) or disabled (0)	Integer	0 - 1	0
full_server_list	Full server list of server mode object, only when authoritative mode is enabled , enabled (1) or disabled (0)	Integer	0 - 1	0
server_mx	Server MX record of server mode object, enabled (1) or disabled (0)	Integer	0 – 1	0
server_mx_additional	MX record additional of server mode object, enabled (1) or disabled (0)	Integer	0 – 1	0
server_ns	Server NS record of server mode object, enabled (1) or disabled (0)	Integer	0 – 1	0
server_auto_ns	Server auto NS of server mode object, enabled (1) or disabled (0)	Integer	0 – 1	0
server_ptr	Server PRT of server mode, enabled (1) or disabled (0)	Integer	0 – 1	0
server_auto_ptr	Server auto PRT of server mode, enabled (1) or disabled (0)	Integer	0 – 1	0
sticky	sticky object of DNS option object	Object		
sticky_aging_time	Sticky aging time of sticky object	Integer	0 – 32	32
sticky_dns_client_ip_mask_len	Sticky dns client IP mask length of Sticky object	Integer	1 - 65535	5
ttl	TTL object of DNS option object	Object		
status	Status of TTL object, enabled (1) or disabled (0)	Integer	0 – 1	1
ttl_time_live	TTL time live of TTL object	Integer	0 - 100000000 0	10
geo_location	Geographic location object	Object		
geo_location_match_first	Match first option of geographic location object, global (0) or policy (1)	Integer	0 - 1	0
geo_location_overlap	Overlap of geographic object, enabled (1) or disabled (0)	Integer	0 – 1	0

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
gslb policy default
!
```

1. HTTP Request in URL Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=gslb.policy.create&policy=name%03policy01%02metric&metric=session_capacity%02active_rdt&session_capacity=enabled%031%02threshold%0380&active_rdt=enabled%031%02samples%038%02difference%035%02tolerance%0320%02timeout%032%02skip%035%02single_shot%031&dns_options=action%031%02action_only%031&format=url
```

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in JSON Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=gslb.policy.create&format=json
```

Request POST data (JSON Format):

```
{
  "policy": {
    "name": "policy01",
    "metric": {
      "session_capacity": {
        "sequence_num": 3,
        "enabled": 0,
        "threshold": 80
      },
      "active_rdt": {
        "sequence_num": 6,
        "enabled": 0,
        "samples": 8,
        "difference": 5,
        "tolerance": 20,
        "timeout": 2,
        "skip": 5,
        "single_shot": 1
      },
      "connection_load": {
        "sequence_num": 8,
        "enabled": 0,
        "limit": 0,
        "samples_num": 5,
        "samples_interval": 5
      },
      "num_session": {
        "sequence_num": 9,
        "enabled": 0,
        "tolerance": 0
      },
      "geo_graphic": {
        "sequence_num": 7,
        "enabled": 1
      }
    }
  }
}
```

```

},
"admin_ip":
{
  "sequence_num":16,
  "enabled":0
},
"weighted_site":
{
  "sequence_num":2,
  "enabled":0
},
"active_servers":
{
  "sequence_num":4,
  "enabled":0
},
"weighted_ip":
{
  "sequence_num":1,
  "enabled":0
},
"bandwidth_cost":
{
  "sequence_num":13,
  "enabled":0
},
"health_check":
{
  "sequence_num":0,
  "enabled":1
},
"admin_preference":
{
  "sequence_num":11,
  "enabled":0
},
"least_response":
{
  "sequence_num":15,
  "enabled":0
},
"round_robin":
{
  "sequence_num":17,
  "enabled":1
},
},
"dns_options":
{
  "action":1,
  "active_only":0,
  "best_only":0,
  "dns_cache":
  {
    "enabled":0,
    "dns_cache_aging_time":0
  },
  "cname_detect":1,
  "external_ip":1,
  "ip_replace":0,
  "geo_location_alias":0,
  "geo_location_action":0,
  "geo_location_policy":0,
  "mx_additional":0,
  "server_mode":
  {
    "enabled":0,
    "authoritative_mode":0,
    "full_server_list":0,
    "server_mx":0,
  }
}

```

```

"server_mx_additional":0,
"server_ns":0,
"server_auto_ns":0,
"server_ptr":0,
"server_auto_ptr":0
},
"sticky":
{
    "enabled":0,
    "sticky_dns_client_ip_mask_len":32,
    "sticky_aging_time":5
},
"ttl":
{
    "enabled":1,
    "ttl_time_live":0
},
"geo_location":
{
    "geo_location_match_first":0,
    "geo_location_overlap":0
}
}
}

```

CLI configuration after aXAPI call:

```

!
gslb policy default
!
gslb policy policy01
    dns action
    dns ttl 0
    capacity threshold 80
    num-session tolerance 0
    active-rdt tolerance 20
    active-rdt difference 5
    active-rdt samples 8
    active-rdt single-shot
    active-rdt timeout 2
    active-rdt skip 5
!
```

Response as the HTTP body:

```

{
    "response":
    {
        "status" : "OK"
    }
}
```

Menus Privilege

Config Mode >> Service >> GSLB >> Policy

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.2.4 “gslb.policy.update” Method

This method is used to update one or more parameters in a GSLB policy.

Parameters

Parameter name	Description	Data Type	Range	Default
<code>session_id (*)</code>	user authenticated session id	String		
<code>method (*)</code>	gslb.policy.update	String		
<code>name(*)</code>	gslb policy name	String	1 - 31	
<code>metric</code>	Metric object	Object		
<code>session_capacity</code>	Session capacity object of metric object	Object		
<code>enabled</code>	Status of session capacity, enabled(1) or disabled (0)	Integer	0 - 1	0
<code>threshold</code>	Threshold of session capacity (%)	Integer	0 - 100	90
<code>active_rdt</code>	Active RDT object of metric object	Object		
<code>enabled</code>	Status of active RDT	Integer	0 - 1	0
<code>samples</code>	The number of samples of active RDT	Integer	1 - 8	5
<code>difference</code>	Difference of active RDT	Integer	0 - 16383	0
<code>tolerance</code>	Tolerance of active RDT	Integer	0 - 100	0
<code>timeout</code>	Timeout of active RDT	Integer	1 - 255	3
<code>skip</code>	Skip of active RDT	Integer	1 - 31	3
<code>single_shot</code>	Single-shot of active RDT, enabled (1) or disabled (0)	Integer	0 - 1	0
<code>connection_load</code>	Connection load object of metric object			
<code>enabled</code>	Status of connection load object, enabled (1) or disabled (0)	Integer	0 - 1	0
<code>limit</code>	Connection limit of connection load object	Integer	0 - 999999999	3
<code>samples_num</code>	The number of samples of connection load	Integer	1 - 8	5
<code>samples_interval</code>	Interval of samples of connection load	Integer	1 - 60	5
<code>num_session</code>	Number session object of metric object			
<code>enabled</code>	Status of number session, enabled (1) or disabled (0)	Integer	0 - 1	0
<code>tolerance</code>	Tolerance of number session	Integer	0 - 100	10
<code>geo_graphic</code>	Geographic object of metric object	Object		
<code>enabled</code>	Status of geographic, enabled (1) or disable (0)	Integer	0 - 1	1
<code>admin_ip</code>	Administrative IP of metric object	Object		
<code>enabled</code>	Status of ordered IP, enabled (1) or disabled (0)	Integer	0 - 1	0
<code>weighted_site</code>	Weighted site object of metric object	Object		
<code>enabled</code>	Status of weighted site object, enabled (1) or disabled (0)	Integer	0 - 1	0
<code>active_servers</code>	Active server object of metric object	Object		
<code>enabled</code>	Status of active server object, enabled (1) or disabled(0)	Integer	0 - 1	0
<code>weighted_ip</code>	Weighted IP object of metric object	Object		
<code>enabled</code>	Status of weight IP, enabled (1) or disabled (0)	Integer	0 - 1	0
<code>bandwidth_cost</code>	Bandwidth cost object of metric object	Object		
<code>enabled</code>	Status of bandwidth cost object, enabled (1) or disabled (0)	Integer	0 - 1	0
<code>health_check</code>	Health check object of metric object	Object		
<code>enabled</code>	Status of health check object, enabled (1) or disabled (0)	integer	0 - 1	1
<code>admin_preference</code>	Admin preference object of metric object	Object		
<code>enabled</code>	Status of administrator preference	Integer	0 - 1	0
<code>least_response</code>	Least response object of metric object	Object		
<code>enabled</code>	Status of least response object, enabled (1) or disabled (0)	Integer	0 - 1	0
<code>round_robin</code>	Round robin object of metric object	Object		
<code>enabled</code>	Status of round robin object	Integer	0 - 1	1
<code>dns_options</code>	DNS option object			
<code>action</code>	Action of DNS option object, enabled (1) or disabled (0)	Integer	0 - 1	0
<code>active_only</code>	Active only option of DNS options object, enabled (1) or disabled (0)	Integer	0 - 1	0
<code>best_only</code>	Best only object of DNS option object	Integer	0 - 1	0
<code>dns_cache</code>	DNS cache status object of DNS option	object		

enabled	Status of DNS cache object, enabled (1) or disabled (0)	Integer	0 – 1	0
dns_cache_aging_time	DNS cache aging time of DNS cache object, only when dns_cache is enabled. Value 0 means 'by TTL'.	Integer	0 - 1000000000	0
cname_detect	CName detect option of DNS option object, enabled (1) or disabled (0)	Integer	0 - 1	1
external_ip	External IP option of DNS option object, enabled (1) or disabled (0)	Integer	0 - 1	1
ip_replace	IP replace option of DNS option object, enabled (1) or disabled (0)	Integer	0 – 1	0
geo_location_alias	Geographic location alias option of DNS option object, enabled (1) or disabled (0)	Integer	0 - 1	0
geo_location_action	Geographic location action option of DNS option object, enabled (1) or disabled (0)	Integer	0 - 1	0
geo_location_policy	Geographic location policy option of DNS option object , enabled (1) or disabled (0)	Integer	0 – 1	0
mx_additional	MX record additional option of DNS option object, enabled (1) or disabled (0)	Integer	0 – 1	0
server_mode	Server mode object of DNS option	Object		
enabled	Status of server mode , enabled (1) or disabled (0)	Integer	0 - 1	0
authoritative_mode	Authoritative mode of server mode object, enabled (1) or disabled (0)	Integer	0 - 1	0
full_server_list	Full server list of server mode object, only when authoritative mode is enabled , enabled (1) or disabled (0)	Integer	0 - 1	0
server_mx	Server MX record of server mode object, enabled (1) or disabled (0)	Integer	0 – 1	0
server_mx_additional	MX record additional of server mode object, enabled (1) or disabled (0)	Integer	0 – 1	0
server_ns	Server NS record of server mode object, enabled (1) or disabled (0)	Integer	0 – 1	0
server_auto_ns	Server auto NS of server mode object, enabled (1) or disabled (0)	Integer	0 – 1	0
server_ptr	Server PRT of server mode, enabled (1) or disabled (0)	Integer	0 – 1	0
server_auto_ptr	Server auto PRT of server mode, enabled (1) or disabled (0)	Integer	0 – 1	0
sticky	sticky object of DNS option object	Object		
sticky_aging_time	Sticky aging time of sticky object	Integer	0 – 32	32
sticky_dns_client_ip_mask_len	Sticky dns client IP mask length of Sticky object	Integer	1 - 65535	5
ttl	TTL object of DNS option object	Object		
status	Status of TTL object, enabled (1) or disabled (0)	Integer	0 – 1	1
ttl_time_live	TTL time live of TTL object	Integer	0 - 1000000000	10
geo_location	Geographic location object	Object		
geo_location_match_first	Match first option of geographic location object, global (0) or policy (1)	Integer	0 - 1	0
geo_location_overlap	Overlap of geographic object, enabled (1) or disabled (0)	Integer	0 – 1	0

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
gslb policy default
!
gslb policy policy01
  dns action
  dns ttl 0
  capacity threshold 80
  num-session tolerance 0
  active-rdt tolerance 20
  active-rdt difference 5
  active-rdt samples 8
  active-rdt single-shot
  active-rdt timeout 2
  active-rdt skip 5
!
```

1. HTTP Request in URL Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=gslb.policy.update&policy=name%03policy01%02metric&metric=session_capacity%02active_rdt&session_capacity=enabled%03%02threshold%0380&active_rdt=enabled%031%02samples%038%02difference%035%02tolerance%0320%02timeout%032%02skip%035%02single_shot%031&dns_options=action%031%02action_only%031&format=url
```

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in JSON Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=gslb.policy.update&format=json
```

Request POST data (JSON Format):

```
{
  "policy": {
    "name": "policy01",
    "metric": {
      "session_capacity": {
        "sequence_num": 3,
        "enabled": 1,
        "threshold": 80
      },
      "active_rdt": {
        "sequence_num": 5,
        "enabled": 1,
        "samples": 8,
        "difference": 5,
        "tolerance": 20,
        "timeout": 2,
        "skip": 5,
        "single_shot": 1
      },
      "connection_load": {
        "sequence_num": 7,
        "enabled": 0,
        "limit": 0,
        "samples_num": 5,
        "samples_interval": 5
      }
    }
  }
}
```

```
},
"num_session":
{
    "sequence_num":8,
    "enabled":0,
    "tolerance":0
},
"geo_graphic":
{
    "sequence_num":6,
    "enabled":1
},
"admin_ip":
{
    "sequence_num":12,
    "enabled":0
},
"weighted_site":
{
    "sequence_num":2,
    "enabled":0
},
"active_servers":
{
    "sequence_num":4,
    "enabled":0
},
"weighted_ip":
{
    "sequence_num":1,
    "enabled":0
},
"bandwidth_cost":
{
    "sequence_num":10,
    "enabled":0
},
"health_check":
{
    "sequence_num":0,
    "enabled":1
},
"admin_preference":
{
    "sequence_num":9,
    "enabled":0
},
"least_response":
{
    "sequence_num":11,
    "enabled":0
},
"round_robin":
{
    "sequence_num":17,
    "enabled":1
}
},
"dns_options":
{
    "action":1,
    "active_only":0,
    "best_only":0,
    "dns_cache":
    {
        "enabled":0,
```

```
"dns_cache_agging_time":0
},
"cname_detect":1,
"external_ip":1,
"ip_replace":0,
"geo_location_alias":0,
"geo_location_action":0,
"geo_location_policy":0,
"mx_additional":0,
"server_mode":
{
    "enabled":0,
    "authoritative_mode":0,
    "full_server_list":0,
    "server_mx":0,
    "server_mx_additional":0,
    "server_ns":0,
    "server_auto_ns":0,
    "server_ptr":0,
    "server_auto_ptr":0
},
"sticky":
{
    "enabled":0,
    "sticky_dns_client_ip_mask_len":32,
    "sticky_agging_time":5
},
"ttl":
{
    "enabled":1,
    "ttl_time_live":0
}
},
"geo_location":
{
    "geo_location_match_first":0,
    "geo_location_overlap":0
}
}
```

CLI configuration after aXAPI call:

```
!
gslb policy default
!
gslb policy policy01
  dns action
  dns ttl 0
  capacity threshold 80
  num-session tolerance 0
  active-rdt tolerance 20
  active-rdt difference 5
  active-rdt samples 8
  active-rdt single-shot
  active-rdt timeout 2
  active-rdt skip 5
  metric-order health-check weighted-ip weighted-site capacity active-servers
    active-rdt geographic connection-load num-session admin-preference bw-cost
      least-response admin-ip
  capacity
  active-rdt
```

Response as the HTTP body:

```
{  
    "response":  
    {  
        "status" : "OK"  
    }  
}
```

Menus Privilege

Config Mode >> Service >> GSLB >> Policy

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.2.5 “gslb.policy.delete” Method

This method is used to delete a GSLB policy from the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	gslb.policy.delete	String		
name (*)	gslb policy name	String		

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
[https://\[AX_IP_ADDRESS\]/services/rest/V2/](https://[AX_IP_ADDRESS]/services/rest/V2/)
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!  
gslb policy default  
!  
gslb policy test01  
  dns active-only  
  dns geoloc-alias  
  metric-order health-check geographic weighted-ip weighted-site capacity active-  
    servers active-rdt connection-load num-session admin-preference bw-cost least-  
    response admin-ip  
!
```

1. **HTTP Request in URL Format:**

session_id=xxxxxxxxxxxxxxxxxxxxxx&method=gslb.policy.delete&name=test01&format=url

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>  
  <response status="ok">  
  </response>
```

2. **HTTP Request in JSON Format:**

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=gslb.policy.delete&format=json
```

Request POST data (JSON Format):

```
{  
    "name" : "test01"  
}
```

CLI configuration after aXAPI call:

```
!  
gslb policy default  
!
```

Response as the HTTP body:

```
{  
    "response":  
    {  
        "status" : "OK"  
    }  
}
```

Menus Privilege

Config Mode >> Service >> GSLB >> Policy

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.3 Service IP

9.3.1 “gslb.service_ip.getAll” Method

This method is used to get configuration information for all GLSB service IPs configured on the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	gslb.service_ip.getAll	String		

Note: (*) parameter is required in the API.

Response Fields

service_ip_list	XML tag of service IP list
service_ip	XML tag of service IP
name	name of service IP
ip_address	ip address of service IP
external_ip_address	external IP address of service IP
health_monitor	health monitor of service IP
status	status of service IP

port_list	XML tag of port list
port	XML tag of port
port_num	port number of port
protocol	protocol of port
health_monitor	health monitor of port
status	status of port

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
gslb service-ip service1 2.6.55.88
  port 21  tcp
  port 80  tcp
!
gslb service-ip service2 2.5.98.4
  external-ip 134.66.78.55
  port 21  tcp
  port 80  tcp
!
```

1. HTTP Request in URL Format:

session_id=xxxxxxxxxxxxxxxxxxxx&method=gslb.service_ip.getAll&format=url

Response as the HTTP body (URL Format):

```
<response status="ok">
  <service_ip_list>
    <service_ip>
      <name>service1</name>
      <ip_address>2.6.55.88</ip_address>
      <external_ip_address>0.0.0.0</external_ip_address>
      <health_monitor>(default)</health_monitor>
      <status>1</status>
      <port_list>
        <port>
          <port_num>21</port_num>
          <protocol>2</protocol>
          <health_monitor>(default)</health_monitor>
          <status>1</status>
        </port>
        <port>
          <port_num>80</port_num>
          <protocol>2</protocol>
          <health_monitor>(default)</health_monitor>
          <status>1</status>
        </port>
      </port_list>
    </service_ip>
    <service_ip>
      <name>service2</name>
      <ip_address>2.5.98.4</ip_address>
      <external_ip_address>134.66.78.55</external_ip_address>
      <health_monitor>(default)</health_monitor>
      <status>1</status>
      <port_list>
        <port>
          <port_num>21</port_num>
          <protocol>2</protocol>
```

```

<health_monitor>(default)</health_monitor>
<status>1</status>
</port>
<port>
<port_num>80</port_num>
<protocol>2</protocol>
<health_monitor>(default)</health_monitor>
<status>1</status>
</port>
</port_list>
</service_ip>
</service_ip_list>
</response>

```

2. HTTP Request in JSON Format:

session_id=xxxxxxxxxxxxxxxxxxxxxx&method=gslb.service_ip.getAll&format=json

Response as the HTTP body (JSON Format):

```
{
  "service_ip_list": [
    {
      "name": "service1",
      "ip_address": "2.6.55.88",
      "external_ip_address": "0.0.0.0",
      "health_monitor": "(default)",
      "status": 1,
      "port_list": [
        {
          "port_num": 80,
          "protocol": 2,
          "health_monitor": "(default)",
          "status": 1
        },
        {
          "port_num": 21,
          "protocol": 2,
          "health_monitor": "(default)",
          "status": 1
        }
      ]
    },
    {
      "name": "service2",
      "ip_address": "2.5.98.4",
      "external_ip_address": "134.66.78.55",
      "health_monitor": "(default)",
      "status": 1,
      "port_list": [
        {
          "port_num": 80,
          "protocol": 2,
          "health_monitor": "(default)",
          "status": 1
        },
        {
          "port_num": 21,
          "protocol": 2,
          "health_monitor": "(default)",
          "status": 1
        }
      ]
    }
  ]
}
```

Menus Privilege

Config Mode >> Service >> GSLB >> Service IP

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.3.2 “gslb.service_ip.search” Method

This method is used to search the GLSB service IPs configured on the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	gslb.service_ip.search	String		
name(*)	the name of the gslb dns proxy	String		

Note: (*) parameter is required in the API.

Response Fields

service_ip	XML tag of service IP
name	name of service IP
ip_address	IP address of service IP
external_ip_address	external IP address of service IP
health_monitor	health monitor of service IP
status	status of service IP
port_list	XML tag of port list
port	XML tag of port
port_num	port number of port
protocol	protocol of port
health_monitor	health monitor of port
status	status of port

Example Response

Request as:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
- **HTTP Action:**
POST

CLI configuration before aXAPI call:

```
!
gslb service-ip service1 2.6.55.88
    port 21  tcp
    port 80  tcp
!
```

1. HTTP Request in URL Format:

session_id=xxxxxxxxxxxxxxxxxxxx&method=gslb.service_ip.search&name=service1&format=url

Response as the HTTP body (URL Format):

```
<response status="ok">
    <service_ip>
        <name>service1</name>
        <ip_address>2.6.55.88</ip_address>
        <external_ip_address>0.0.0.0</external_ip_address>
```

```

<health_monitor>(default)</health_monitor>
<status>1</status>
<port_list>
  <port>
    <port_num>21</port_num>
    <protocol>2</protocol>
    <health_monitor>(default)</health_monitor>
    <status>1</status>
  </port>
  <port>
    <port_num>80</port_num>
    <protocol>2</protocol>
    <health_monitor>(default)</health_monitor>
    <status>1</status>
  </port>
</port_list>
</service_ip>
</response>

```

2. HTTP Request in JSON Format:

session_id=xxxxxxxxxxxxxxxxxxxx&method=gslb.service_ip.search&format=json

Request POST data (JSON Format):

```
{
  "name" : "service1"
}
```

Response as the HTTP body:

```
{
  "service_ip":
  {
    "name": "service1",
    "ip_address": "2.6.55.88",
    "external_ip_address": "0.0.0.0",
    "health_monitor": "(default)",
    "status": 1,
    "port_list": [
      {
        "port_num": 80,
        "protocol": 2,
        "health_monitor": "(default)",
        "status": 1
      },
      {
        "port_num": 21,
        "protocol": 2,
        "health_monitor": "(default)",
        "status": 1
      }
    ]
  }
}
```

Menus Privilege

Config Mode >> Service >> GSLB >> Service IP

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.3.3 “gslb.service_ip.create” Method

This method is used to create a GSLB service IP address.

Parameters

Parameter name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	gslb.service_ip.create	String		
service_ip	Service IP object	Object		
name(*)	Name of service IP object	String	1 - 31	--
ip_address(*)	IP address of service IP object	IPv4	0.0.0.0 – 255.255.255.255	--
external_ip_address	External IP address of service IP	IPv4	0.0.0.0 – 255.255.255.255	0.0.0.0
health_monitor	Name of health monitor	String	--	(default)
status	Status of service IP object, enabled (1) or disabled (0)	Integer	0 - 1	1
port_list	Port object list	List		
port	Port object	Object		
port_num(*)	Port number of the port object	Integer	0 - 65535	--
protocol	Protocol of port object, TCP (2) or UDP (3)	Integer	2 - 3	2
health_monitor	Health monitor (name) of service IP object	String	--	--
status	Status of service IP object	Integer	0 - 1	1

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
[https://\[AX_IP_ADDRESS\]/services/rest/V2/](https://[AX_IP_ADDRESS]/services/rest/V2/)
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
!
```

1. HTTP Request in URL Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=gslb.service_ip.create&service_ip=name%03service6%
02ip_address%031.6.61.41%02external_ip_address%03123.123.123.15%02health_monitor%03ping%0
2status%031%02port_list&port_list=p1%02p2%02p3&p1=port_num%038888&p2=port_num%038787%02he
alth_monitor%03ping&p3=port_num%038789%02health_monitor%03ping&format=url
```

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in JSON Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=gslb.service_ip.create&format=json
```

Request POST data (JSON Format):

```
{
  "service_ip": [
    {
      "name": "service6",
      "ip_address": "1.6.61.41",
      "external_ip_address": "123.123.123.15",
      "health_monitor": "ping",
      "status": 1,
      "port_list": [
        {
          "port_num": 8789,
          "protocol": 2,
          "health_monitor": "ping",
          "status": 1
        },
        {
          "port_num": 8787,
          "protocol": 2,
          "health_monitor": "ping",
          "status": 1
        },
        {
          "port_num": 8888,
          "protocol": 2,
          "health_monitor": "(default)",
          "status": 1
        }
      ]
    }
  ]
}
```

CLI configuration after aXAPI call:

```
!
gslb service-ip service6 1.6.61.41
  external-ip 123.123.123.15
  health-check ping
  port 8888  tcp
  port 8787  tcp
    health-check ping
  port 8789  tcp
    health-check ping
!
```

Response as the HTTP body:

```
{
  "response": [
    {
      "status": "OK"
    }
  ]
}
```

Menus Privilege

Config Mode >> Service >> GSLB >> Service IP

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.3.4 “gslb.service_ip.update” Method

This method is used to update a GSLB service IP address on the AX device.

Parameters

Parameter name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	gslb.service_ip.update	String		
service_ip	Service IP object	Object		
name(*)	Name of service IP object	String	1 - 31	--
ip_address(*)	IP address of service IP object	IPv4	0.0.0.0 – 255.255.255.255	--
external_ip_address	External IP address of service IP	IPv4	0.0.0.0 – 255.255.255.255	0.0.0.0
health_monitor	Name of health monitor	String	--	(default)
status	Status of service IP object, enabled (1) or disabled (0)	Integer	0 - 1	1
port_list	Port object list	List		
port	Port object	Object		
port_num(*)	Port number of the port object	Integer	0 - 65535	--
protocol	Protocol of port object, TCP (2) or UDP (3)	Integer	2 - 3	2
health_monitor	Health monitor (name) of service IP object	String	--	--
status	Status of service IP object	Integer	0 - 1	1

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
gslb service-ip service6 1.6.61.41
  external-ip 123.123.123.15
  health-check ping
  port 8888 tcp
  port 8787 tcp
    health-check ping
  port 8789 tcp
    health-check ping
!
!
```

1. HTTP Request in URL Format:

```
session_id=xxxxxxxxxxxxxxxxxxxxxx&method=gslb.service_ip.update&service_ip=name%03service6%02ip_address%031.6.61.41%02external_ip_address%03123.123.123.15%02health_monitor%03ping%02status%031%02port_list&port_list=p1%02p2%02p3%02p4&p1=port_num%038888&p2=port_num%038787%02health_monitor%03ping&p3=port_num%038789%02health_monitor%03ping&p4=port_num%038880&format=url
```

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in JSON Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=gslb.service_ip.update&format=json
```

Request POST data (JSON Format):

```
{  
    "service_ip":  
    {  
        "name": "service6",  
        "ip_address": "1.6.61.41",  
        "external_ip_address": "123.123.123.15",  
        "health_monitor": "ping",  
        "status": 1,  
        "port_list": [  
            {  
                "port_num": 8880,  
                "protocol": 2,  
                "health_monitor": "(default)",  
                "status": 1  
            },  
            {  
                "port_num": 8789,  
                "protocol": 2,  
                "health_monitor": "ping",  
                "status": 1  
            },  
            {  
                "port_num": 8787,  
                "protocol": 2,  
                "health_monitor": "ping",  
                "status": 1  
            },  
            {  
                "port_num": 8888,  
                "protocol": 2,  
                "health_monitor": "(default)",  
                "status": 1  
            }  
        ]  
    }  
}
```

CLI configuration after aXAPI call:

```
!  
gslb service-ip service6 1.6.61.41  
    external-ip 123.123.123.15  
    health-check ping  
    port 8888  tcp  
    port 8787  tcp  
        health-check ping  
    port 8789  tcp  
        health-check ping  
    port 8880  tcp  
!
```

Response as the HTTP body:

```
{  
    "response":  
    {  
        "status": "OK"  
    }  
}
```

```
}
```

Menus Privilege

Config Mode >> Service >> GSLB >> Service IP

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.3.5 “gslb.service_ip.delete” Method

This method is used to delete a GSLB service IP from the AX.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	gslb.service_ip.delete	String		
name (*)	gslb service IP name	String		

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
gslb service-ip test01 6.6.6.6
  external-ip 51.51.12.33
  port 80  tcp
  port 123  udp
!
```

1. HTTP Request in URL Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=gslb.service_ip.delete&name=test01&format=url
```

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in JSON Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=gslb.service_ip.delete&format=json
```

Request POST data (JSON Format):

```
{
  "name" : "test01"
}
```

CLI configuration after aXAPI call:

```
!
```

Response as the HTTP body:

```
{  
    "response":  
    {  
        "status" : "OK"  
    }  
}
```

Menus Privilege

Config Mode >> Service >> GSLB >> Service IP

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.3.6 “gslb.service_ip.port.create” Method

This method is used to add a port to a GSLB service IP.

Parameters

Parameter name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	gslb.service_ip.port.create	String		
name(*)	Name of service IP object	String	1 - 31	--
port	Port object	Object		
port_num(*)	Port number of the port object	Integer	0 - 65535	--
protocol	Protocol of port object, TCP (2) or UDP (3)	Integer	2 - 3	2
health_monitor	Health monitor (name) of service IP object	String	--	--
status	Status of service IP object	Integer	0 - 1	1

Notes: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
[https://\[AX_IP_ADDRESS\]/services/rest/V2/](https://[AX_IP_ADDRESS]/services/rest/V2/)
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!  
gslb service-ip test01 6.6.6.6  
  external-ip 51.51.12.33  
  port 80  tcp  
  port 123  udp  
!
```

1. HTTP Request in URL Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=gslb.service_ip.port.create&name=test01&port_num=8880&protocol=3&format=url
```

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in JSON Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=gslb.service_ip.port.create&format=json
```

Request POST data (JSON Format):

```
{
  "name": "test01",
  "port": {
    "port_num": 8880,
    "protocol": 3
  }
}
```

CLI configuration after aXAPI call:

```
!
gslb service-ip test01 6.6.6.6
  external-ip 51.51.12.33
  port 80  tcp
  port 123  udp
  port 8080  udp
!
```

Response as the HTTP body:

```
{
  "response": {
    "status": "OK"
  }
}
```

Privileges

Config Mode >> Service >> GSLB >> Service IP

Error Codes

TBD

9.3.7 “gslb.service_ip.port.update” Method

This method is used to update a GSLB service IP port.

Parameters

Parameter name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		

method (*)	gslb.service_ip.port.update	String		
name(*)	Name of service IP object	String	1 - 31	--
port	Port object	Object		
port_num(*)	Port number of the port object	Integer	0 - 65535	--
protocol	Protocol of port object, TCP (2) or UDP (3)	Integer	2 - 3	2
health_monitor	Health monitor (name) of service IP object	String	--	--
status	Status of service IP object	Integer	0 - 1	1

Notes: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
gslb service-ip test01 6.6.6.6
  external-ip 51.51.12.33
  port 80  tcp
  port 123  udp
  port 8080  udp
!
```

1. HTTP Request in URL Format:

```
session_id=xxxxxxxxxxxxxxxxxxxxxx&method=gslb.service_ip.port.update&name=test01&por
t=port_num%038080%02protocol%033&status=0&format=url
```

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in JSON Format:

```
session_id=xxxxxxxxxxxxxxxxxxxxxx&method=gslb.service_ip.port.update&format=json
```

Request POST data (JSON Format):

```
{
  "name": "test01",
  "port" :
  {
    "port_num": 8880,
    "protocol": 3,
    "status": 0
  }
}
```

CLI configuration after aXAPI call:

```
!
gslb service-ip test01 6.6.6.6
  external-ip 51.51.12.33
  port 80  tcp
  port 123  udp
  port 8080  udp
  disable
```

```
!
```

Response as the HTTP body:

```
{  
    "response":  
    {  
        "status" : "OK"  
    }  
}
```

Privileges

Config Mode >> Service >> GSLB >> Service IP

Error Codes

TBD

9.3.8 “gslb.service_ip.port.delete” Method

This method is used to delete a port from a GSLB service IP, using a given port number or protocol type.

Parameters

Parameter name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	gslb.service_ip.port.delete	String		
name(*)	Name of service IP object	String	1 - 31	--
port	Port object	Object		
port_num(*)	Port number of the port object	Integer	0 - 65535	--
protocol(*)	Protocol of port object, TCP (2) or UDP (3)	Integer	2 - 3	2

Notes: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
[https://\[AX_IP_ADDRESS\]/services/rest/V2/](https://[AX_IP_ADDRESS]/services/rest/V2/)
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!  
gslb service-ip test01 6.6.6.6  
  external-ip 51.51.12.33  
  port 80  tcp  
  port 123  udp  
  port 8080  udp  
    disable  
!
```

1. HTTP Request in URL Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=gslb.service_ip.port.delete&name=test01&port=port_num%038080%02protocol%03&format=url
```

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>  
<response status="ok">
```

```
</response>
```

2. HTTP Request in JSON Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=gslb.service_ip.port.delete&format=json
```

Request POST data (JSON Format):

```
{  
    "name": "test01",  
    "port" :  
    {  
        "port_num": 8880,  
        "protocol": 3  
    }  
}
```

CLI configuration after aXAPI call:

```
!  
gslb service-ip test01 6.6.6.6  
    external-ip 51.51.12.33  
    port 80  tcp  
    port 123  udp  
!
```

Response as the HTTP body:

```
{  
    "response":  
    {  
        "status" : "OK"  
    }  
}
```

Privileges

Config Mode >> Service >> GSLB >> Service IP

Error Codes

TBD

9.4 Site

9.4.1 “gslb.site.getAll” Method

This method is used to get configuration information for all GLSB sites configured on the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	gslb.site.getAll	String		

Note: (*) parameter is required in the API.

Response Fields

gslb_site_list	XML tag of GSLB site list
gslb_site	XML tag of GSLB site

name	name
weight	weight
template	name of GSLB template
status	site status is enabled(1) or disabled(0)
bandwidth_cost	XML tag of bandwidth cost
limit	bandwidth limit of bandwidth cost
threshold	threshold of bandwidth cost
active_rdt	XML tag of active RDT
aging_time	aging time of active RDT
bind_geoloc	bind geographic of active RDT
overlap	overlap of active RDT
limit	limit of active RDT
mask_len	mask length of active RDT
range_factor	range factor of active RDT
smooth_factor	smooth factor of active RDT
ip_server_list	XML tag of IP server list
ip_server	XML tag of IP server
name	name of IP server
ip_addr	IP address of SLB device
slb_device_list	XML tag of SLB device list
slb_device	XML tag of SLB device
name	SLB device name
ip_addr	IP address of SLB device
admin_preference	admin preference option
max_client	max num of clients
gateway	gateway
vip_server_list	XML tag of VIP server list
vip_server	XML tag of virtual server
name	virtual server name

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
gslb service-ip service1 2.6.55.88
    port 21  tcp
    port 80  tcp
gslb service-ip service2 2.5.98.4
    external-ip 134.66.78.55
    port 21  tcp
    port 80  tcp
!
gslb template snmp snmp_name
    host ccc
    username aaa
    community bbb
!
gslb site site1
    disable
    bw-cost limit 0 threshold 1
    template snmp_name
```

```

slb-dev device1 2.6.5.33
    vip-server service1
gslb site site2
    disable
    weight 100
    bw-cost limit 0 threshold 1
    template snmp_name
    active-rtt aging-time 15360
    active-rtt bind-geoloc overlap
    slb-dev aaa 55.66.99.55
        gateway 7.7.7.7
    ip-server service2
!

```

1. HTTP Request in URL Format:

session_id=xxxxxxxxxxxxxxxxxxxxxx&method=gslb.site.getAll&format=url

Response as the HTTP body (URL Format):

```

<response status="ok">
    <gslb_site_list>
        <gslb_site>
            <name>site1</name>
            <weight>1</weight>
            <template>snmp_name</template>
            <status>0</status>
            <bandwidth_cost>
                <limit>0</limit>
                <threshold>1</threshold>
            </bandwidth_cost>
            <active_rdt>
                <aging_time>10</aging_time>
                <bind_geoloc>0</bind_geoloc>
                <overlap>0</overlap>
                <limit>16383</limit>
                <mask_len>32</mask_len>
                <range_factor>25</range_factor>
                <smooth_factor>10</smooth_factor>
            </active_rdt>
        <ip_server_list/>
        <slb_device_list>
            <slb_device>
                <name>Device1</name>
                <ip_addr>2.6.5.33</ip_addr>
                <admin_preference>100</admin_preference>
                <max_client>32768</max_client>
                <gateway>0.0.0.0</gateway>
                <vip_server_list>
                    <vip_server>
                        <name>service1</name>
                    </vip_server>
                </vip_server_list>
            </slb_device>
        </slb_device_list>
    </gslb_site>
    <gslb_site>
        <name>site2</name>
        <weight>100</weight>
        <template>snmp_name</template>
        <status>0</status>
        <bandwidth_cost>
            <limit>0</limit>
            <threshold>1</threshold>
        </bandwidth_cost>
        <active_rdt>
            <aging_time>15360</aging_time>
            <bind_geoloc>1</bind_geoloc>
            <overlap>1</overlap>
            <limit>16383</limit>
            <mask_len>32</mask_len>
            <range_factor>25</range_factor>
        </active_rdt>
    </gslb_site>
</response>

```

```

        <smooth_factor>10</smooth_factor>
    </active_rdt>
    <ip_server_list>
        <ip_server>
            <name>service2</name>
        </ip_server>
    </ip_server_list>
    <slb_device_list>
        <slb_device>
            <name>aaa</name>
            <ip_addr>55.66.99.55</ip_addr>
            <admin_preference>100</admin_preference>
            <max_client>32768</max_client>
            <gateway>7.7.7.7</gateway>
            <vip_server_list/>
        </slb_device>
    </slb_device_list>
</gslb_site>
</gslb_site_list>
</response>

```

2. HTTP Request in JSON Format:

session_id=xxxxxxxxxxxxxxxxxxxx&method=gslb.site.getAll&format=json

Response as the HTTP body (JSON Format):

```
{
    "gslb_site_list": [
        {
            "name": "site1",
            "weight": 1,
            "template": "snmp_name",
            "status": 0,
            "bandwidth_cost": {
                "status": 1,
                "limit": 0,
                "threshold": 1
            },
            "active_rdt": {
                "aging_time": 10,
                "bind_geoloc": 0,
                "ignore_count": 5,
                "limit": 16383,
                "mask_len": 32,
                "range_factor": 25,
                "smooth_factor": 10
            },
            "ip_server_list": [],
            "slb_device_list": [
                {
                    "name": "device1",
                    "ip_addr": "2.6.5.33",
                    "admin_preference": 100,
                    "max_client": 32768,
                    "gateway": "0.0.0.0",
                    "passive_rtt_timer": 3,
                    "vip_server_list": [
                        {
                            "name": "service1"
                        }
                    ],
                    "protocol_aging_fast": 1,
                    "protocol_aging_time": 0,
                    "protocol_compatible": 0
                }
            ]
        }
    ]
}
```

```

        ],
    },
    {
        "name": "site2",
        "weight": 100,
        "template": "snmp_name",
        "status": 0,
        "bandwidth_cost": [
            {
                "status": 1,
                "limit": 0,
                "threshold": 1
            }
        ],
        "active_rdt": [
            {
                "aging_time": 15360,
                "bind_geoloc": 1,
                "ignore_count": 5,
                "limit": 16383,
                "mask_len": 32,
                "range_factor": 25,
                "smooth_factor": 10
            }
        ],
        "ip_server_list": [
            {
                "name": "service2"
            }
        ],
        "slb_device_list": [
            {
                "name": "aaa",
                "ip_addr": "55.66.99.55",
                "admin_preference": 100,
                "max_client": 32768,
                "gateway": "7.7.7.7",
                "passive_rtt_timer": 3,
                "vip_server_list": [],
                "protocol_aging_fast": 1,
                "protocol_aging_time": 0,
                "protocol_compatible": 0
            }
        ]
    }
}
]
}

```

Menus Privilege

Config Mode >> Service >> GSLB >> Site

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.4.2 “gslb.site.search” Method

This method is used to search GLSB sites configured on the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	gslb.site.search	String		
name(*)	the name of the gslb site	String		

Note: (*) parameter is required in the API.

Response Fields

gslb_site		XML tag of GSLB site
name	name	
weight	weight	
template	name of GSLB template	
status	site status is enabled(1) or disabled(0)	
bandwidth_cost		XML tag of bandwidth cost
limit	bandwidth limit of bandwidth cost	
threshold	threshold of bandwidth cost	
active_rdt		XML tag of active RDT
aging_time	aging time of active RDT	
bind_geoloc	bind geographic of active RDT	
overlap	overlap of active RDT	
limit	limit of active RDT	
mask_len	mask length of active RDT	
range_factor	range factor of active RDT	
smooth_factor	smooth factor of active RDT	
ip_server_list		XML tag of IP server list
ip_server		XML tag of IP server
name	name of IP server	
ip_addr	IP address of SLB device	
slb_device_list		XML tag of SLB device list
slb_device		XML tag of SLB device
name	SLB device name	
ip_addr	IP address of SLB device	
admin_preference	admin preference option	
max_client	max num of clients	
gateway	gateway	
vip_server_list	XML tag of VIP server list	
vip_server	XML tag of virtual server	
name	virtual server name	

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
gslb service-ip service1 2.6.55.88
    port 21  tcp
    port 80  tcp
!
gslb template snmp snmp_name
    host ccc
    username aaa
    community bbb
!
gslb site site1
    bw-cost limit 0 threshold 1
    template snmp_name
    slb-dev device1 2.6.5.33
    vip-server service1
!
```

1. HTTP Request in URL Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=gslb.site.search&name=site1&format=url
```

Response as the HTTP body (URL Format):

```
<response status="ok">
  <gslb_site>
    <name>site1</name>
    <weight>1</weight>
    <template>snmp_name</template>
    <status>0</status>
    <bandwidth_cost>
      <limit>0</limit>
      <threshold>1</threshold>
    </bandwidth_cost>
    <active_rdt>
      <aging_time>10</aging_time>
      <bind_geoloc>0</bind_geoloc>
      <overlap>0</overlap>
      <limit>16383</limit>
      <mask_len>32</mask_len>
      <range_factor>25</range_factor>
      <smooth_factor>10</smooth_factor>
    </active_rdt>
    <ip_server_list/>
    <slb_device_list>
      <slb_device>
        <name>Device1</name>
        <ip_addr>2.6.5.33</ip_addr>
        <admin_preference>100</admin_preference>
        <max_client>32768</max_client>
        <gateway>0.0.0.0</gateway>
        <vip_server_list>
          <vip_server>
            <name>service1</name>
          </vip_server>
        </vip_server_list>
      </slb_device>
    </slb_device_list>
  </gslb_site>
</response>
```

2. HTTP Request in JSON Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=gslb.site.search&format=json
```

Request POST data (JSON Format):

```
{
  "name" : "site1"
}
```

Response as the HTTP body:

```
{
  "gslb_site": {
    "name": "site1",
    "weight": 1,
    "template": "snmp_name",
    "status": 1,
    "bandwidth_cost": {
      "status": 1,
      "limit": 0,
      "threshold": 1
    }
  }
}
```

```

"active_rdt":
{
    "aging_time":10,
    "bind_geoloc":0,
    "ignore_count":5,
    "limit":16383,
    "mask_len":32,
    "range_factor":25,
    "smooth_factor":10
},
"ip_server_list":[],
"slb_device_list":[
{
    "name":"device1",
    "ip_addr":"2.6.5.33",
    "admin_preference":100,
    "max_client":32768,
    "gateway":"0.0.0.0",
    "passive_rtt_timer":3,
    "vip_server_list":[
    {
        "name":"service1"
    }
    ],
    "protocol_aging_fast":1,
    "protocol_aging_time":0,
    "protocol_compatible":0
}
]
}
}

```

Menus Privilege

Config Mode >> Service >> GSLB >> Site

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.4.3 “gslb.site.create” Method

This method is used to create a GSLB site.

Parameters

parameter name	Description	Data Type	Range	Default
session_id (*)	User-authenticated session ID	String		
method (*)	gslb.site.create	String		
site(*)	GSLB site object			
name(*)	Name of the GSLB site	String	1 - 31	--
weight	Weight of GSLB site	Integer	1 - 100	1
template	GSLB template of site	String	1 - 31	--
status	Site status, enabled (1) or disabled (0)	Bool	0 or 1	1
bandwidth_cost	Bandwidth cost object	Object		
status	Status of bandwidth cost, enabled (1) or disabled(0)	Integer	0 - 1	0
limit	Limitation of bandwidth cost	Integer	0 - 2147483647	0
threshold	Threshold of bandwidth cost	Integer	0 - 100	0
active_rdt	Active RDT object	Object		
aging_time	Aging time of active RDT	Integer	1 - 15360	10
bind_geoloc	Bind geographic of active RDT, enabled (1) or disabled (0)	Integer	0 - 1	0
overlap	Overlap of active RDT	Integer	0 - 1	0
limit	Limit of active RDT	Integer	1 - 16383	16383

mask_len	Mask length of active RDT	Integer	0 - 32	32
range_factor	Range factor of active RDT	Integer	0 - 1000	25
smooth_factor	Smooth factor of active RDT	Integer	0 - 100	10
slb_device_list	SLB device list	List		
slb_device	SLB device object	Object		
name(*)	Name of SLB device	String	1 - 31	--
ip_addr(*)	Up address of SLB device	IPv4	0.0.0.0 – 255.255.255.255	--
admin_preference(*)	Administrator preference option of SLB device	Integer	0 - 255	100
max_client	Max clients of SLB device	Integer	1 - 2147483647	32768
gateway	Gateway of SLB device	IPv4	0.0.0.0 – 255.255.255.255	0.0.0.0
vip_server_list	Virtual server object list	List		
vip_server	Virtual server object			
name	Name of virtual server	String	1 - 31	--

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
- **HTTP Action:**
POST

CLI configuration before aXAPI call:

```
!
```

```
!
```

1. HTTP Request in URL Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=gslb.site.create&gslb_site=name%03site31%02weight%035%02bandwidth_cost%02slb_device_list&bandwidth_cost=status%031%02limit%03100%02threshold%0310&slb_device_list=l1%02l2&l1=name%03ABC%02ip_addr%032.4.6.9%02admin_preference%0310&l2=name%03ABD%02ip_addr%032.4.6.10%02admin_preference%0310&format=url
```

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in JSON Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=gslb.site.create&format=json
```

Request POST data (JSON Format):

```
{
  "gslb_site": {
    "name": "site31",
    "weight": 5,
    "template": "",
    "status": 1,
    "bandwidth_cost": {}
```

```

    "status":1,
    "limit":100,
    "threshold":10
},
"active_rdt":
{
    "aging_time":10,
    "bind_geoloc":0,
    "ignore_count":5,
    "limit":16383,
    "mask_len":0
},
"ip_server_list":[],
"slb_device_list":[
{
    "name":"ABC",
    "ip_addr":"2.4.6.9",
    "admin_preference":10,
    "max_client":32768,
    "gateway":"0.0.0.0",
    "passive_rtt_timer":3,
    "vip_server_list":[],
    "protocol_aging_fast":1,
    "protocol_aging_time":0,
    "protocol_compatible":0
},
{
    "name":"ABD",
    "ip_addr":"2.4.6.10",
    "admin_preference":10,
    "max_client":32768,
    "gateway":"0.0.0.0",
    "passive_rtt_timer":3,
    "vip_server_list":[],
    "protocol_aging_fast":1,
    "protocol_aging_time":0,
    "protocol_compatible":0
}
]
}

```

CLI configuration after aXAPI call:

```

!
gslb site site31
    weight 5
    bw-cost limit 100 threshold 10
    no passive-rtt smooth-factor
    no passive-rtt range-factor
    passive-rtt mask /0
    no active-rtt smooth-factor
    no active-rtt range-factor
    active-rtt mask /0
    slb-dev ABC 2.4.6.9
        admin-preference 10
    slb-dev ABD 2.4.6.10
        admin-preference 10
!

```

Response as the HTTP body:

```
{

```

```

    "response":
    {
        "status" : "OK"
    }
}

```

Menus Privilege

Config Mode >> Service >> GSLB >> Site

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.4.4 “gslb.site.update” Method

This method is used to update one or more parameters associated with a given GSLB site.

Parameters

parameter name	Description	Data Type	Range	Default
session_id (*)	User-authenticated session ID	String		
method (*)	gslb.site.update	String		
site(*)	GSLB site object			
name(*)	Name of the GSLB site	String	1 - 31	--
weight	Weight of GSLB site	Integer	1 – 100	1
template	GSLB template of site	String	1 - 31	--
status	Site status, enabled(1) or disabled (0)	Bool	0 or 1	1
bandwidth_cost	Bandwidth cost object	Object		
status	Status of bandwidth cost, enabled (1) or disabled(0)	Integer	0 - 1	0
limit	Limitation of bandwidth cost	Integer	0 - 2147483647	0
threshold	Threshold of bandwidth cost	Integer	0 – 100	0
active_rdt	Active RDT object	Object		
aging_time	Aging time of active RDT	Integer	1 - 15360	10
bind_geoloc	Bind geographic of active RDT, enabled (1) or disabled (0)	Integer	0 - 1	0
overlap	Overlap of active RDT	Integer	0 – 1	0
limit	Limit of active RDT	Integer	1 – 16383	16383
mask_len	Mask length of active RDT	Integer	0 - 32	32
range_factor	Range factor of active RDT	Integer	0 - 1000	25
smooth_factor	Smooth factor of active RDT	Integer	0 – 100	10
slb_device_list	SLB device list	List		
slb_device	SLB device object	Object		
name(*)	Name of SLB device	String	1 - 31	--
ip_addr(*)	Up address of SLB device	IPv4	0.0.0.0 – 255.255.255.2 55	--
admin_preference(*)	Administrator preference option of SLB device	Integer	0 - 255	100
max_client	Max clients of SLB device	Integer	1 - 2147483647	32768
gateway	Gateway of SLB device	IPv4	0.0.0.0 – 255.255.255.2 55	0.0.0.0
vip_server_list	Virtual server object list	List		
vip_server	Virtual server object			
name	Name of virtual server	String	1 - 31	--

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
- **HTTP Action:**
POST

CLI configuration before aXAPI call:

```
!
gslb site site31
  weight 5
  bw-cost limit 100 threshold 10
  no passive-rtt smooth-factor
  no passive-rtt range-factor
  passive-rtt mask /0
  no active-rtt smooth-factor
  no active-rtt range-factor
  active-rtt mask /0
  slb-dev ABC 2.4.6.9
    admin-preference 10
  slb-dev ABD 2.4.6.10
    admin-preference 10
!
```

1. HTTP Request in URL Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=gslb.site.update&gslb_site=name%03site31%02
weight%035%02bandwidth_cost%02slb_device_list&bandwidth_cost=status%031%02limit%03100%02t
hreshold%0310&slb_device_list=l1%0212&l1=name%03ABC%02ip_addr%032.4.6.9%02admin_preferenc
e%0340&l2=name%03ABD%02ip_addr%032.4.6.10%02admin_preference%0360&format=url
```

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in JSON Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=gslb.site.update&format=json
```

Request POST data (JSON Format):

```
{
  "gslb_site": {
    "name": "site31",
    "weight": 5,
    "template": "",
    "status": 1,
    "bandwidth_cost": [
      {
        "status": 1,
        "limit": 100,
        "threshold": 10
      }
    ],
    "active_rdt": [
      {
        "aging_time": 10,
        "bind_geoloc": 0,
        "ignore_count": 5,
        "limit": 16383,
        "mask_len": 0
      }
    ]
  }
}
```

```

"ip_server_list":[] ,
"slb_device_list": [
{
  "name": "ABC",
  "ip_addr": "2.4.6.9",
  "admin_preference": 40,
  "max_client": 32768,
  "gateway": "0.0.0.0",
  "passive_rtt_timer": 3,
  "vip_server_list": [],
  "protocol_aging_fast": 1,
  "protocol_aging_time": 0,
  "protocol_compatible": 0
},
{
  "name": "ABD",
  "ip_addr": "2.4.6.10",
  "admin_preference": 60,
  "max_client": 32768,
  "gateway": "0.0.0.0",
  "passive_rtt_timer": 3,
  "vip_server_list": [],
  "protocol_aging_fast": 1,
  "protocol_aging_time": 0,
  "protocol_compatible": 0
}
]
}

```

CLI configuration after aXAPI call:

```

!
gslb site site31
  weight 5
  bw-cost limit 100 threshold 10
  no passive-rtt smooth-factor
  no passive-rtt range-factor
  passive-rtt mask /0
  no active-rtt smooth-factor
  no active-rtt range-factor
  active-rtt mask /0
  slb-dev ABC 2.4.6.9
    admin-preference 40
  slb-dev ABD 2.4.6.10
    admin-preference 60
!
```

Response as the HTTP body:

```
{
  "response": {
    "status": "OK"
  }
}
```

Menus Privilege

Config Mode >> Service >> GSLB >> Site

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.4.5 “gslb.site.delete” Method

This method is used to delete a GSLB site from the AX.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	gslb.site.delete	String		
name (*)	the name of the gslb site	String		

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
gslb site test01
  slb-dev 12.23.34.34 12.23.34.34
    admin-preference 1
!
```

1. HTTP Request in URL Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=gslb.site.delete&name=test01&format=url
```

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in JSON Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=gslb.site.delete&format=json
```

Request POST data (JSON Format):

```
{
  "name" : "test01"
}
```

CLI configuration after aXAPI call:

```
!
!
```

Response as the HTTP body:

```
{
  "response":
```

```

        "status" : "OK"
    }
}
```

Menus Privilege

Config Mode >> Service >> GSLB >> Site

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.4.6 “gslb.site.slb_device.create” Method

This method is used to add a server to a GSLB site.

Parameters

parameter name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	gslb.site.slb_device.create	String		
name(*)	Name of the GSLB site	String	1 - 31	--
slb_device(*)	SLB device object	Object		
name(*)	Name of SLB device	String	1 - 31	--
ip_addr(*)	Up address of SLB device	IPv4	0.0.0.0 – 255.255.255.255	--
admin_preference	Administrator preference option of SLB device	Integer	0 - 255	100
max_client	Max clients of SLB device	Integer	1 - 2147483647	32768
gateway	Gateway of SLB device	IPv4	0.0.0.0 – 255.255.255.255	0.0.0.0

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```

!
gslb service-ip test 10.10.10.11
    port 80  tcp
gslb service-ip test3 10.10.10.13
    port 21  tcp
!
gslb site yyy
!
```

1. HTTP Request in URL Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=gslb.site.slb_device.create&name=yyy&slb_device=name%031234%02ip_addr%038.8.8%02admin_preference%0310%02vip_server_list&vip_server_list=11%0212&l1=name%03test&l2=name%03test3&format=url
```

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in JSON Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=gslb.site.slb_device.create&format=json
```

Request POST data (JSON Format):

```
{
  "name": "yyy",
  "slb_device": {
    {
      "name": "1234",
      "ip_addr": "8.8.8.8",
      "admin_preference": 10,
      "vip_server_list": [
        {
          "name": "test"
        },
        {
          "name": "test3"
        }
      ]
    }
  }
}
```

CLI configuration after aXAPI call:

```
!
gslb service-ip test 10.10.10.11
  port 80  tcp
gslb service-ip test3 10.10.10.13
  port 21  tcp
!
gslb site yyy
  slb-dev 1234 8.8.8.8
    admin-preference 10
    vip-server test
    vip-server test3
!
```

Response as the HTTP body:

```
{
  "response": {
    {
      "status": "OK"
    }
  }
}
```

Menus Privilege

Config Mode >> Service >> GSLB >> Site

Please see “Admin Roles” on page 29 for the privileges associated with this method.

9.4.7 “gslb.site.slb_device.update” Method

This method is used to update one or more parameters configured for a server at a given GSLB site.

Parameters

parameter name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	gslb.site.slb_device.update	String		
name(*)	Name of the GSLB site	String	1 - 31	--
slb_device(*)	SLB device object	Object		
name(*)	Name of SLB device	String	1 - 31	--
ip_addr	Up address of SLB device	IPv4	0.0.0.0 – 255.255.255.255	--
admin_preference	Administrator preference option of SLB device	Integer	0 - 255	100
max_client	Max clients of SLB device	Integer	1 - 2147483647	32768
Gateway	Gateway of SLB device	IPv4	0.0.0.0 – 255.255.255.255	0.0.0.0

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
gslb service-ip test 10.10.10.11
    port 80  tcp
gslb service-ip test3 10.10.10.13
    port 21  tcp
!
gslb site yyy
    slb-dev 1234 8.8.8.8
        admin-preference 10
        vip-server test
        vip-server test3
!
```

1. HTTP Request in URL Format:

session_id=xxxxxxxxxxxxxxxxxxxx&method=gslb.site.slb_device.update&name=yyy&slb_device=name%031234%02ip_addr%038.8.8.8%02admin_preference%0350&max_client=65535%02vip_server_list&vip_server_list=11%0212&l1=name%03test&l2=name%03test3&format=url

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in JSON Format:

session_id=xxxxxxxxxxxxxxxxxxxx&method=gslb.site.slb_device.update&format=json

Request POST data (JSON Format):

```
{  
    "name": "yyy",  
    "slb_device":  
    {  
        "name": "1234",  
        "ip_addr": "8.8.8.8",  
        "admin_preference": 50,  
        "max_client": 65535,  
        "vip_server_list": [  
            {  
                "name": "test"  
            },  
            {  
                "name": "test3"  
            }  
        ]  
    }  
}
```

CLI configuration after aXAPI call:

```
!  
gslb service-ip test 10.10.10.11  
    port 80  tcp  
gslb service-ip test3 10.10.10.13  
    port 21  tcp  
!  
gslb site yyy  
    slb-dev 1234 8.8.8.8  
        admin-preference 50  
        max-client 65535  
        vip-server test  
        vip-server test3  
!
```

Response as the HTTP body:

```
{  
    "response":  
    {  
        "status": "OK"  
    }  
}
```

Menus Privilege

Config Mode >> Service >> GSLB >> Site

Please see “Admin Roles” on page 29 for the privileges associated with this method.

9.4.8 “gslb.site.slb_device.delete” Method

This method is used to delete a server from a GSLB site.

Parameters

parameter name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	gslb.site.slb_device.update	String		
name(*)	Name of the GSLB site	String	1 - 31	--

slb_device(*)	SLB device object	Object		
name(*)	Name of SLB device	String	1 - 31	--

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
gslb service-ip test 10.10.10.11
    port 80  tcp
gslb service-ip test3 10.10.10.13
    port 21  tcp
!
gslb site yyy
    slb-dev 1234 8.8.8.8
        admin-preference 50
        max-client 65535
        vip-server test
        vip-server test3
!
```

1. HTTP Request in URL Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=gslb.site.slb_device.delete&name=yyy&slb_device=name%031234&format=url
```

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in JSON Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=gslb.site.slb_device.delete&format=json
```

Request POST data (JSON Format):

```
{
    "name": "yyy",
    "slb_device":
    {
        "name": 1234
    }
}
```

CLI configuration after aXAPI call:

```
!
gslb service-ip test 10.10.10.11
    port 80  tcp
gslb service-ip test3 10.10.10.13
    port 21  tcp
!
gslb site yyy
```

```
!
```

Response as the HTTP body:

```
{  
    "response":  
    {  
        "status" : "OK"  
    }  
}
```

Menus Privilege

Config Mode >> Service >> GSLB >> Site

Please see “Admin Roles” on page 29 for the privileges associated with this method.

9.4.9 “gslb.site.ip_server.create” Method

This method is used to add a server IP to a GSLB site.

Parameters

parameter name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	gslb.site.ip_server.create	String		
name(*)	Name of the GSLB site	String	1 - 31	--
vip_server	Virtual server Object			
name	Name of virtual server	String	1 - 31	--

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!  
gslb service-ip test01 6.6.6.6  
    external-ip 51.51.12.33  
    port 80  tcp  
    port 123  udp  
!  
gslb site site01  
!
```

1. HTTP Request in URL Format:

session_id=xxxxxxxxxxxxxxxxxxxxx&method=gslb.site.ip_server.create&name=site01&vip_server=name%03test01&format=url

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>  
    <response status="ok">
```

```
</response>
```

2. HTTP Request in JSON Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=gslb.site.ip_server.create&format=json
```

Request POST data (JSON Format):

```
{  
    "name": "site01",  
    "vip_server":  
    {  
        "name": "test01"  
    }  
}
```

CLI configuration after aXAPI call:

```
!  
gslb service-ip test01 6.6.6.6  
    external-ip 51.51.12.33  
    port 80  tcp  
    port 123  udp  
!  
gslb site site01  
    ip-server test01  
!
```

Response as the HTTP body:

```
{  
    "response":  
    {  
        "status": "OK"  
    }  
}
```

Menus Privilege

Config Mode >> Service >> GSLB >> Site

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.4.10 “gslb.site.ip_server.delete” Method

This method is used to delete a service IP from a GSLB site.

Parameters

parameter name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	gslb.site.ip_server.delete	String		
name(*)	Name of the gslb site	String	1 - 31	--
vip_server	Virtual server Object			
name	Name of virtual server	String	1 - 31	--

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
gslb service-ip test01 6.6.6.6
  external-ip 51.51.12.33
  port 80  tcp
  port 123  udp
!
gslb site site01
  ip-server test01
!
```

1. HTTP Request in URL Format:

session_id=xxxxxxxxxxxxxxxxxxxxxx&method=gslb.site.ip_server.delete&name=site01&vip_server=name%03test01&format=url

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in JSON Format:

session_id=xxxxxxxxxxxxxxxxxxxxxx&method=gslb.site.ip_server.delete&format=json

Request POST data (JSON Format):

```
{
  "name": "site01",
  "vip_server": [
    {
      "name": "test01"
    }
}
```

CLI configuration after aXAPI call:

```
!
gslb service-ip test01 6.6.6.6
  external-ip 51.51.12.33
  port 80  tcp
  port 123  udp
!
gslb site site01
!
```

Response as the HTTP body:

```
{
  "response": [
    {
      "status": "OK"
    }
}
```

}

Menus Privilege

Config Mode >> Service >> GSLB >> Site

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.5 SNMP Template

9.5.1 “gslb.snmp_template.getAll” Method

This method is used to get configuration information for all SNMP templates configured on the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	gslb.snmp_template.getAll	String		

Note: (*) parameter is required in the API.

Response Fields

snmp_template_list	XML tag of SNMP template object list
snmp_template	XML tag of SNMP template object
name	SNMP template name
user_name	SNMP template user name
community	community
host	host name or host IP
port	SNMP port
version	SNMP version
oid	SNMP OID
interface	interface
security_level	secure level
security_engine_id	security engine ID
auth_key	authentication key
auth_protocol	authentication protocol
private_key	private key
private_protocol	private protocol
context_engine_id	context engine ID
context_name	context name
interval	interval

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
gslb template snmp snmp_name
    interface 25
    host 192.168.3.6
    username plumeer
    oid .1.3.6
    community public
    auth-key AAAA
    priv-key AAAA
    context-name AAA
    context-engine-id AAA
    security-engine-id AAAA
gslb template snmp template01
    interface 161
    host Plumeer-PC
```

```

username Alexander
oid .1.3.6
community USA
auth-key public
priv-key public
context-name 123
context-engine-id 123
security-engine-id 5
!
```

1. HTTP Request in URL Format:

session_id=xxxxxxxxxxxxxxxxxxxx&method=gslb.snmp_template.getAll&format=url

Response as the HTTP body (URL Format):

```

<response status="ok">
<snmp_template_list>
    <snmp_template>
        <name>template01</name>
        <user_name>Alexander</user_name>
        <community>USA</community>
        <host>Plumeer-PC</host>
        <port>161</port>
        <version>3</version>
        <oid>.1.3.6</oid>
        <interface>161</interface>
        <security_level>1</security_level>
        <security_engine_id>5</security_engine_id>
        <auth_key>public</auth_key>
        <auth_protocol>1</auth_protocol>
        <private_key>public</private_key>
        <private_protocol>1</private_protocol>
        <context_engine_id>123</context_engine_id>
        <context_name>123</context_name>
        <interval>3</interval>
    </snmp_template>
    <snmp_template>
        <name>snmp_name</name>
        <user_name>plumeer</user_name>
        <community>public</community>
        <host>192.168.3.6</host>
        <port>161</port>
        <version>3</version>
        <oid>.1.3.6</oid>
        <interface>25</interface>
        <security_level>1</security_level>
        <security_engine_id>AAAA</security_engine_id>
        <auth_key>AAAA</auth_key>
        <auth_protocol>1</auth_protocol>
        <private_key>AAAA</private_key>
        <private_protocol>1</private_protocol>
        <context_engine_id>AAA</context_engine_id>
        <context_name>AAA</context_name>
        <interval>3</interval>
    </snmp_template>
</snmp_template_list>
</response>
```

2. HTTP Request in JSON Format:

session_id=xxxxxxxxxxxxxxxxxxxx&method=gslb.snmp_template.getAll&format=json

Response as the HTTP body (JSON Format):

```
{
    "snmp_template_list": [
        {
            "name": "template01",
            "user_name": "Alexander",
            "community": "USA",
```

```

        "host":"Plumeer-PC",
        "port":161,
        "version":3,
        "oid":".1.3.6",
        "interface":161,
        "security_level":1,
        "security_engine_id":"5",
        "auth_key":"public",
        "auth_protocol":1,
        "private_key":"public",
        "private_protocol":1,
        "context_engine_id":"123",
        "context_name":"123",
        "interval":3
    },
    {
        "name":"snmp_name",
        "user_name":"plumeer",
        "community":"public",
        "host":"192.168.3.6",
        "port":161,
        "version":3,
        "oid":".1.3.6",
        "interface":25,
        "security_level":1,
        "security_engine_id":"AAAA",
        "auth_key":"AAAA",
        "auth_protocol":1,
        "private_key":"AAAA",
        "private_protocol":1,
        "context_engine_id":"AAA",
        "context_name":"AAA",
        "interval":3
    }
]
}

```

Menus Privilege

Config Mode >> Service >> GSLB >> Site

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.5.2 “gslb.snmp_template.search” Method

This method is used to search a GSLB SNMP template configured on the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	gslb.snmp_template.search	String		
name(*)	the name of the gslb snmp template	String		

Note: (*) parameter is required in the API.

Response Fields

snmp_template	XML tag of SNMP template object
name	SNMP template name
user_name	SNMP template user name
community	community
host	host name or host IP

port	SNMP port
version	SNMP version
oid	SNMP OID
interface	interface
security_level	secure level
security_engine_id	security engine ID
auth_key	authentication key
auth_protocol	authentication protocol
private_key	private key
private_protocol	private protocol
context_engine_id	context engine ID
context_name	context name
interval	interval

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
gslb template snmp template01
  interface 161
  host Plumeer-PC
  username Alexander
  oid .1.3.6
  community USA
  auth-key public
  priv-key public
  context-name 123
  context-engine-id 123
  security-engine-id 5
!
```

1. HTTP Request in URL Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=gslb.snmp_template.search&name=template01&format=url
```

Response as the HTTP body (URL Format):

```
<response status="ok">
  <snmp_template>
    <name>template01</name>
    <user_name>Alexander</user_name>
    <community>USA</community>
    <host>Plumeer-PC</host>
    <port>161</port>
    <version>3</version>
    <oid>.1.3.6</oid>
    <interface>161</interface>
    <security_level>1</security_level>
    <security_engine_id>5</security_engine_id>
    <auth_key>public</auth_key>
    <auth_protocol>1</auth_protocol>
    <private_key>public</private_key>
    <private_protocol>1</private_protocol>
    <context_engine_id>123</context_engine_id>
    <context_name>123</context_name>
    <interval>3</interval>
  </snmp_template>
```

```
</response>
```

2. HTTP Request in JSON Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=gslb.snmp_template.search&format=json
```

Request POST data (JSON Format):

```
{  
    "name" : "template01"  
}
```

Response as the HTTP body:

```
{  
    "snmp_template":  
    {  
        "name":"template01",  
        "user_name":"Alexander",  
        "community":"USA",  
        "host":"Plumeer-PC",  
        "port":161,  
        "version":3,  
        "oid":".1.3.6",  
        "interface":161,  
        "security_level":1,  
        "security_engine_id":5,  
        "auth_key":public,  
        "auth_protocol":1,  
        "private_key":public,  
        "private_protocol":1,  
        "context_engine_id":123,  
        "context_name":123,  
        "interval":3  
    }  
}
```

Menus Privilege

Config Mode >> Service >> GSLB >> Site

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.5.3 “gslb.snmp_template.create” Method

This method is used to create a GSLB SNMP template on the AX device.

Parameters

parameter name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	gslb.snmp_template.create	String		
Snmp_template(*)	GSLB snmp_template object	Object		
name	Name of SNMP template	String	1 - 31	--
user_name	Username of SNMP template	String	1 - 255	
community	Community of SNMP template	String	1 - 255	--
host	Host of SNMP template	String	1 - 255	--
port	Port of SNMP template	Integer	0 - 65535	--
version	SNMP version of SNMP template. 1: SNMP v2 2: SNMP v2c 3: SNMP v3	Integer	1 - 3	3

oid	SNMP Object ID of SNMP template	OID(String)	1 - 255	--
interface	Interface of SNMP	Integer	1 - 65535	161
security_level	Security level of SNMP template 0: default/empty 1: no auth no priv 2: auth no priv 3. auth priv	Integer	0 - 3	0
security_engine_id	Security level engine ID	String	1 - 255	--
auth_key	Authentication key	String	1 - 255	--
auth_protocol	Authentication protocol 0: default/empty 1: md5 2: sha	Integer	0 - 2	0
private_key	Private key string	String	1 - 255	--
private_protocol	Private protocol 0: default/empty 1: des 2: aes	Integer	0 - 2	0
context_engine_id	Context engine ID	String	1 - 255	--
context_name	Context name	String	1 - 255	--
interval	Interval	Integer	1 - 999	3

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
!
```

1. HTTP Request in URL Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=gslb.snmp_template.create&snmp_template=name%03template02%02user_name%03plumeer%02community%03public&format=url
```

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in JSON Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=gslb.snmp_template.create&format=json
```

Request POST data (JSON Format):

```
{
  "snmp_template": {
    "name": "template02",
    "user_name": "plumeer",
    "community": "public",
    "host": "",
    "port": 161,
    "version": 3,
```

```

        "oid":"",
        "interface":0,
        "security_level":1,
        "security_engine_id":"",
        "auth_key":"",
        "auth_protocol":1,
        "private_key":"",
        "private_protocol":1,
        "context_engine_id":"",
        "context_name":"",
        "interval":3
    }
}

```

CLI configuration after aXAPI call:

```
!
gslb template snmp template02
    username plumeer
    community public
!
```

Response as the HTTP body:

```
{
    "response":
    {
        "status" : "OK"
    }
}
```

Menus Privilege

Config Mode >> Service >> GSLB >> Site

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.5.4 “gslb.snmp_template.update” Method

This method is used to update the parameters configured in a GSLB SNMP template.

Parameters

parameter name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	gslb.snmp_template.update	String		
Snmp_template(*)	GSLB snmp_template object	Object		
name	Name of SNMP template	String	1 – 31	--
user_name	Username of SNMP template	String	1 - 255	
community	Community of SNMP template	String	1 - 255	--
host	Host of SNMP template	String	1 - 255	--
port	Port of SNMP template	Integer	0 - 65535	--
version	SNMP version of SNMP template. 1: SNMP v2 2: SNMP v2c 3: SNMP v3	Integer	1 - 3	3
oid	SNMP Object ID of SNMP template	OID(String)	1 - 255	--
interface	Interface of SNMP	Integer	1 - 65535	161
security_level	Security level of SNMP template 0: default/empty 1: no auth no priv	Integer	0 - 3	0

	2: auth no priv 3: auth priv			
security_engine_id	Security level engine ID	String	1 - 255	--
auth_key	Authentication key	String	1 - 255	--
auth_protocol	Authentication protocol 0: default/empty 1: md5 2: sha	Integer	0 - 2	0
private_key	Private key string	String	1 - 255	--
private_protocol	Private protocol 0: default/empty 1: des 2: aes	Integer	0 - 2	0
context_engine_id	Context engine ID	String	1 - 255	--
context_name	Context name	String	1 - 255	--
interval	Interval	Integer	1 - 999	3

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
gslb template snmp template02
    username plumeer
    community public
!
```

1. HTTP Request in URL Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=gslb.snmp_template.update&snmp_template=name%03template02%02user_name%03plumeer%02community%03public&format=url
```

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in JSON Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=gslb.snmp_template.update&format=json
```

Request POST data (JSON Format):

```
{
  "snmp_template": {
    "name": "template02",
    "user_name": "plumeer_2",
    "community": "public_2",
    "host": "",
    "port": 161,
    "version": 3,
    "oid": "",
    "interface": 0,
```

```

    "security_level":1,
    "security_engine_id":"",
    "auth_key":"",
    "auth_protocol":1,
    "private_key":"",
    "private_protocol":1,
    "context_engine_id":"",
    "context_name":"",
    "interval":3
}
}

```

CLI configuration after aXAPI call:

```
!
gslb template snmp template02
    username plumeer_2
    community public_2
!
```

Response as the HTTP body:

```
{
    "response":
    {
        "status" : "OK"
    }
}
```

Menus Privilege

Config Mode >> Service >> GSLB >> Site

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.5.5 “gslb.snmp_template.delete” Method

This method is used to delete a GSLB SNMP template from the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	gslb.snmp_template.delete	String		
name (*)	the name of the snmp template	String		

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
```

```

gslb template snmp template01
  interface 123
  host ccc
  username aaa
  oid .1.1.1
  community bbb
!

```

1. HTTP Request in URL Format:

```

session_id=xxxxxxxxxxxxxxxxxxxx&method=gslb.snmp_template.delete&name=template01&format=url

```

Response as the HTTP body (URL Format):

```

<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>

```

2. HTTP Request in JSON Format:

```

session_id=xxxxxxxxxxxxxxxxxxxx&method=gslb.snmp_template.delete&format=json

```

Request POST data (JSON Format):

```
{
  "name" : "template01"
}
```

CLI configuration after aXAPI call:

```
!
```

Response as the HTTP body:

```
{
  "response": {
    {
      "status" : "OK"
    }
  }
}
```

Menus Privilege

Config Mode >> Service >> GSLB >> Site

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.6 Zone

9.6.1 “gslb.zone.getAll” Method

This method is used to get configuration information for all GSLB zones configured on the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		

method (*)	gslb.zone.getAll	String		
-------------------	-------------------------	--------	--	--

Note: (*) parameter is required in the API.

Response Fields

zone_list	XML tag of GSLB zone list
zone	XML tag of GSLB zone
name	name of GSLB zone
ttl	TTL of GSLB zone
policy	policy of GSLB zone
status	zone status is enabled(1) or disabled(0)
dns_mx_record_list	XML tag of DNS MX record list
dns_mx_record	XML tag of DNS MX record
name	name of DNS MX record
priority	priority of DNS MX record
dns_ns_record_list	XML tag of DNS NS record list
dns_ns_record	XML tag of DNS DS record
name	name of DNS DN record
service_list	XML tag of service list
service	XML tag of service
name	name of service
port	port of service
policy	policy of service
action	action of service
dns_address_record_list	XML tag of DNS address record list
dns_address_record	XML tag of DNS address record
as_replace	as replace option of DNS address record
no_response	no replace option of DNS address record
static	static option of DNS address record
weight	weight option of DNS address record
dns_mx_record_list	XML tag of DNS MX record list
dns_mx_record	XML tag of DNS MX record
name	name of DNS MX record
priority	priority of DNS MX record
dns_cname_record_list	XML tag of DNS CName record list
dns_cname_record	XML tag of DNS CName record
name	name of DNS CName
dns_ns_record_list	XML tag of DNS NS record list
dns_ns_record	XML tag of DNS NS record
name	name of DNS NS record
dns_ptr_record_list	XML tag of DNS PTR record list
dns_ptr_record	XML tag of DNS PTR record
name	name of DNS PTR record
dns_srv_record_list	XML tag of DNS server record list
dns_srv_record	XML tag of DNS server record
name	DNS server name
port	DNS server port (0 – 65535) (65535 means ‘not set’)
priority	priority of this DNS service
weight	weight of this DNS service
dns_txt_record_list	DNS TXT record list
dns_txt_record	DNS TXT record

txt	DNS TXT record (string length: 1 - 511)
geo_location_list	Geography location list
geo_location	Geography location
geo_location	geography location string (string length: 1 - 127)
alias	DNS alias
geo_action	(string length: 1 – 1024, lower case only) Selected action for this geo-location
0: do nothing	
1: forward response	
2: forward query and response	
3: forward query	
4: drop	
5: reject	
6: allow	
policy	Name of the profile for this GSLB policy. (Policy used to handle this geo-location (string length, 1 - 127))

Note: The geo-location parameter ‘geo_action’ and ‘policy’ are mutually exclusive. If both are configured at the same time, the ‘policy’ parameter takes effect.

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
gslb zone aaa10
    disable
    service ftp service2
        action forward response
        dns-cname-record fff
        dns-cname-record ggg
        dns-a-record test01 as-replace static
        ip-order test01
        dns-mx-record vvv 123
        dns-mx-record bbb 234
        dns-ns-record hhh
        dns-ns-record jjj
        dns-ptr-record asd
        dns-ptr-record qwe
    !
gslb zone aaa11
!
```

1. HTTP Request in URL Format:

session_id=xxxxxxxxxxxxxxxxxxxx&method=gslb.zone.getAll&format=url

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
    <zone_list>
        <zone>
```

```

<name>aaa10</name>
<ttl>10</ttl>
<policy>default</policy>
<status>0</status>
<dns_mx_record_list>
    <dns_mx_record>
        <name>pc-02</name>
        <priority>50</priority>
    </dns_mx_record>
    <dns_mx_record>
        <name>pc-01</name>
        <priority>100</priority>
    </dns_mx_record>
</dns_mx_record_list>
<dns_ns_record_list>
    <dns_ns_record>
        <name>nnn</name>
    </dns_ns_record>
    <dns_ns_record>
        <name>mmm</name>
    </dns_ns_record>
</dns_ns_record_list>
<service_list>
    <service>
        <name>service2</name>
        <port>21</port>
        <policy>default</policy>
        <action>1</action>
        <dns_address_record_list>
            <dns_address_record>
                <as_replace>1</as_replace>
                <no_response>1</no_response>
                <static>1</static>
                <weight>0</weight>
            </dns_address_record>
            <dns_address_record>
                <as_replace>1</as_replace>
                <no_response>1</no_response>
                <static>1</static>
                <weight>0</weight>
            </dns_address_record>
            <dns_address_record>
                <as_replace>1</as_replace>
                <no_response>1</no_response>
                <static>1</static>
                <weight>0</weight>
            </dns_address_record>
        </dns_address_record_list>
    </service>
</service_list>
<dns_mx_record_list>
    <dns_mx_record>
        <name>vvv</name>
        <priority>123</priority>
    </dns_mx_record>
    <dns_mx_record>
        <name>bbb</name>
        <priority>234</priority>
    </dns_mx_record>
</dns_mx_record_list>
<dns cname_record_list>
    <dns_cname_record>
        <name>fff</name>
    </dns_cname_record>
    <dns_cname_record>
        <name>ggg</name>
    </dns_cname_record>
</dns cname_record_list>
<dns_ns_record_list>
    <dns_ns_record>
        <name>hhh</name>
    </dns_ns_record>
</dns_ns_record_list>
<dns_ns_record>

```

```

                <name>jjj</name>
            </dns_ns_record>
        </dns_ns_record_list>
        <dns_ptr_record_list>
            <dns_ptr_record>
                <name>asd</name>
            </dns_ptr_record>
            <dns_ptr_record>
                <name>qwe</name>
            </dns_ptr_record>
        </dns_ptr_record_list>
        <geo_location_list/>
        </service>
        </service_list>
    </zone>
    <zone>
        <name>aaa11</name>
        <ttl>10</ttl>
        <policy>default</policy>
        <dns_mx_record_list/>
        <dns_ns_record_list/>
        <service_list/>
    </zone>
</zone_list>
</response>

```

2. HTTP Request in JSON Format:

session_id=xxxxxxxxxxxxxxxxxxxxxx&method=gslb.zone.getAll&format=json

Response as the HTTP body (JSON Format):

```
{
    "zone_list": [
        {
            "name": "aaa10",
            "ttl": 10,
            "policy": "default",
            "disable_all_services": 1,
            {
                "dns_mx_record_list": []
            },
            {
                "dns_ns_record_list": []
            },
            "service_list": [
                {
                    "name": "service2",
                    "port": 21,
                    "policy": "default",
                    "action": 1,
                    "dns_address_record_list": [
                        {
                            "vip_order": "test01",
                            "as_replace": 1,
                            "no_response": 1,
                            "static": 1,
                            "weight": 0
                        }
                    ],
                    "dns_mx_record_list": [
                        {
                            "name": "vvv",
                            "priority": 123
                        }
                    ]
                }
            ]
        }
    ]
}
```

```

        },
        {
            "name":"bbb",
            "priority":234
        }
    ],
    "dns cname record list": [
    {
        "name":"fff"
    },
    {
        "name":"ggg"
    }
],
    "dns ns record list": [
    {
        "name":"hhh"
    },
    {
        "name":"jjj"
    }
],
    "dns ptr record list": [
    {
        "name":"asd"
    },
    {
        "name":"qwe"
    }
],
    "geo location list": []
}
]
},
{
    "name":"aaa11",
    "ttl":10,
    "policy":"default",
    "disable_all_services":0,
    {
        "dns mx record list": []
    },
    {
        "dns ns record list": []
    },
    "service list": []
}
]
}

```

Menus Privilege

Config Mode >> Service >> GSLB >> Zone

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.6.2 “gslb.zone.search” Method

This method is used to search configuration information for a given GSLB zone configured on the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	gslb.zone.search	String		
name(*)	the name of the gslb zone	String		

Note: (*) parameter is required in the API.

Response Fields

zone	XML tag of GSLB zone
name	name of GSLB zone
ttl	TTL of GSLB zone
policy	policy of GSLB zone
status	zone status is enabled(1) or disabled(0)
dns_mx_record_list	XML tag of DNS MX record list
dns_mx_record	XML tag of DNS MX record
name	name of DNS MX record
priority	priority of DNS MX record
dns_ns_record_list	XML tag of DNS NS record list
dns_ns_record	XML tag of DNS DS record
name	name of DNS DN record
service_list	XML tag of service list
service	XML tag of service
name	name of service
port	port of service
policy	policy of service
action	action of service
dns_address_record_list	XML tag of DNS address record list
dns_address_record	XML tag of DNS address record
as_replace	as replace option of DNS address record
no_response	no replace option of DNS address record
static	static option of DNS address record
weight	weight option of DNS address record
dns_mx_record_list	XML tag of DNS MX record list
dns_mx_record	XML tag of DNS MX record
name	name of DNS MX record
priority	priority of DNS MX record
dns_cname_record_list	XML tag of DNS CName record list
dns_cname_record	XML tag of DNS CName record
name	name of DNS CName
dns_ns_record_list	XML tag of DNS NS record list
dns_ns_record	XML tag of DNS NS record
name	name of DNS NS record
dns_ptr_record_list	XML tag of DNS PTR record list
dns_ptr_record	XML tag of DNS PTR record
name	name of DNS PTR record

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
gslb zone aaa10
  disable
  service ftp service2
    action forward response
    dns-cname-record fff
    dns-cname-record ggg
    dns-a-record test01 as-replace static
    ip-order test01
    dns-mx-record vvv 123
    dns-mx-record bbb 234
    dns-ns-record hhh
    dns-ns-record jjj
    dns-ptr-record asd
    dns-ptr-record qwe
!
```

1. HTTP Request in URL Format:

```
session_id=xxxxxxxxxxxxxxxxxxxxxx&method=gslb.zone.search&name=aaa10&format=url
```

Response as the HTTP body (URL Format):

```
<response status="ok">
  <zone_list>
    <zone>
      <name>aaa10</name>
      <ttl>10</ttl>
      <policy>default</policy>
      <status>0</status>
      <dns_mx_record_list>
        <dns_mx_record>
          <name>pc-02</name>
          <pri>50</pri>
        </dns_mx_record>
        <dns_mx_record>
          <name>pc-01</name>
          <pri>100</pri>
        </dns_mx_record>
      </dns_mx_record_list>
      <dns_ns_record_list>
        <dns_ns_record>
          <name>nnn</name>
        </dns_ns_record>
        <dns_ns_record>
          <name>mmm</name>
        </dns_ns_record>
      </dns_ns_record_list>
      <service_list>
        <service>
          <name>service2</name>
          <port>21</port>
          <policy>default</policy>
          <action>1</action>
          <dns_address_record_list>
            <dns_address_record>
              <as_replace>1</as_replace>
              <no_response>1</no_response>
              <static>1</static>
              <weight>0</weight>
            </dns_address_record>
          </dns_address_record_list>
        </service>
      </service_list>
    </zone>
  </zone_list>
</response>
```

```

<as_replace>1</as_replace>
<no_response>1</no_response>
<static>1</static>
<weight>0</weight>
</dns_address_record>
<dns_address_record>
<as_replace>1</as_replace>
<no_response>1</no_response>
<static>1</static>
<weight>0</weight>
</dns_address_record>
</dns_address_record_list>
<dns_mx_record_list>
<dns_mx_record>
<name>www</name>
<priority>123</priority>
</dns_mx_record>
<dns_mx_record>
<name>bbb</name>
<priority>234</priority>
</dns_mx_record>
</dns_mx_record_list>
<dns cname_record_list>
<dns cname_record>
<name>fff</name>
</dns cname_record>
<dns cname_record>
<name>ggg</name>
</dns cname_record>
</dns cname_record_list>
<dns_ns_record_list>
<dns_ns_record>
<name>hhh</name>
</dns_ns_record>
<dns_ns_record>
<name>jjj</name>
</dns_ns_record>
</dns_ns_record_list>
<dns_ptr_record_list>
<dns_ptr_record>
<name>asd</name>
</dns_ptr_record>
<dns_ptr_record>
<name>qwe</name>
</dns_ptr_record>
</dns_ptr_record_list>
<geo_location_list/>
</service>
</service_list>
</zone>
</zone_list>
</response>

```

2. HTTP Request in JSON Format:

session_id=xxxxxxxxxxxxxxxxxxxxx&method=gslb.zone.search&format=json

Request POST data (JSON Format):

```
{
  "name" : "aaa10"
}
```

Response as the HTTP body:

```
{
  "zone":
```

```
{
    "name": "aaa10",
    "ttl": 10,
    "policy": "default",
    "disable_all_services": 1,
    {
        "dns_mx_record_list": []
    },
    {
        "dns_ns_record_list": []
    },
    "service_list": [
        {
            "name": "service2",
            "port": 21,
            "policy": "default",
            "action": 1,
            "dns_address_record_list": [
                {
                    "vip_order": "test01",
                    "as_replace": 1,
                    "no_response": 1,
                    "static": 1,
                    "weight": 0
                }
            ],
            "dns_mx_record_list": [
                {
                    "name": "vvv",
                    "priority": 123
                },
                {
                    "name": "bbb",
                    "priority": 234
                }
            ],
            "dns cname record_list": [
                {
                    "name": "fff"
                },
                {
                    "name": "ggg"
                }
            ],
            "dns_ns_record_list": [
                {
                    "name": "hhh"
                },
                {
                    "name": "jjj"
                }
            ],
            "dns_ptr_record_list": [
                {
                    "name": "asd"
                },
                {
                    "name": "qwe"
                }
            ],
            "geo_location_list": []
        }
    ]
}
```

Menus Privilege

Config Mode >> Service >> GSLB >> Zone

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.6.3 “gslb.zone.create” Method

This method is used to create a GSLB zone on the AX device.

Parameters

parameter name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	gslb.zone.create	String		
zone(*)	GSLB ZONE object	Object		
name(*)	Name of GSLB zone	String	1 - 31	--
ttl	TTL object of GSLB zone	Object		
Status	Status of TTL, enabled (1) or disabled (0)	Integer	0 - 1	1
ttl_timer	TTL timer of TTL object	Integer	0 - 1000000000	10
status	Status of zone, enabled(1) or disabled(0)	Bool	0 - 1	1
policy	Policy of GSLB zone	String	1 - 31	default
dns_mx_record_list	DNS MX record list of of GSLB zone	List		
dns_mx_record	DNS MX record	Object		
 name	Name of DNS MX record	String	1 - 127	--
 priority	Priority of DNS MX record	Integer	0 - 65535	--
dns_ns_record_list	DNS NS record list of GSLB zone	List		
dns_ns_record	DNS NS record object	Object		
 name	Name of NS record object	String	1 - 127	--
service_list	Service list of GSLB zone	List		
service	Service object	Object		
 name	Name of service	String	1 - 31	--
port	Port of service	Integer	0 - 65535	--
policy	Policy of service	String	1 - 31	--
action	Action of service 0: do nothing 1: forward response 2: forward both 3: forward query 4: drop 5: reject	Integer	0 - 5	0
dns_address_record_list	DNS address record list of GSLB zone	List	--	--
dns_address_record	DNS address record object	Object		
 as_replace	As replace option of DNS address record, enabled (1) or disabled (0)	Integer	0 - 1	0
 no_response	No response option of DNS address record , enabled (1) or disabled (0)	Integer	0 - 1	0
 static	Static option of DNS address record, enabled (1) or disabled (0)	Integer	0 - 1	0
 weight	Weight of DNS address record, 0 means disabled	Integer	0 - 100	0
dns_mx_record_list	DNS MX record list of of GSLB zone	List		
dns_mx_record	DNS MX record object	Object		
 name	Name of DNS MX record	String	1 - 127	--
 priority	Priority of DNS MX record	Integer	0 - 65535	0
dns_cname_record_list	DNS CName record list of GSLB zone	List		
dns_cname_record	DNS CName record object	Object		
 name	Name of DNS CName record object	String	1 - 127	--
dns_ns_record_list	DNS NS record list of of GSLB zone	List		
dns_ns_record	DNS NS record object	Object		
 name	Name of DNS NS record	String	1 - 127	--
dns_ptr_record_list	DNS PTR record list	List		
dns_ptr_record	DNS PTR record object	Object		
 name	Name of DNS PTR record object	String	1 - 127	--

dns_srv_record_list	DNS server record list	List	--	--
dns_srv_record	DNS server record	Object	--	--
name	DNS server name	String	1 – 127	--
port	DNS server port (value scope is 0 – 65535, 65535 means not set)	Integer	0 - 65535	65535 (not set)
priority	Priority of this DNS service	Integer	0 - 65535	0
weight	Weight of this DNS service	Integer	0 - 65535	0
dns_txt_record_list	DNS TXT record list	List	--	--
dns_txt_record	DNS TXT record	Object	--	--
txt	Text of this DNS TXT record	String	1 - 511	--
geo_location_list(**)	Geography location list	List	--	--
geo_location	Geography location	Object	--	--
geo_location	Geography location string	String	1 - 127	--
alias	DNS alias	String	1 - 1024	--
geo_action	Action when this geo-location is hit (0: do nothing, 1: forward response, 2: forward both query and response, 3: forward query, 4: drop, 5: reject, 6: allow)	Integer	0 – 6	0
policy	Profile name for the GSLB policy used to handle this geo-location	String	1 – 127	--

Note: (*) Parameter is required in the API.

(**) The geo-location parameter ‘geo_action’ and ‘policy’ are mutually exclusive. If both are configured at the same time, the ‘policy’ parameter takes effect.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
!
```

1. HTTP Request in URL Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=gslb.zone.create&zone=name%03CCC1%02ttl%031
%02policy%03default%02dns_mx_record_list%02dns_ns_record_list%02service_list&dns_mx_recor
d_list=m1%02m2%02m3&m1=name%031234%02priority%03123&m2=name%031235%02priority%03321&m3=na
me%031237%02priority%0332&dns_ns_record_list=n1%02n2&n1=name%03dddddd&n2=name%03fffff&se
rvice_list=s11%02s12&s11=name%03fffffd%02port%0345%02policy%03default&s12=name%03ffffeee
%02port%0321%02policy%03default%02dns_mx_record_list%02dns_address_record_list&dns_mx_rec
ord_list=mx1%02mx2&mx1=name%03ffff%02priority%0312&mx2=name%03fffff1%02priority%0312&dns_a
ddress_record_list=j1%02j2&j1=vip_order%03service1&j2=vip_order%03service2%02as_replace%0
31&format=url
```

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in JSON Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=gslb.zone.create&format=json
```

Request POST data (JSON Format):

```
{  
  "zone":  
  {  
    "name":"ccc1",  
    "ttl":1,  
    "policy":"default",  
    "disable_all_services":0,  
    {  
      "dns_mx_record_list":[]  
    },  
    {  
      "dns_ns_record_list":[]  
    },  
    "service_list": [  
      {  
        "name":"ffdfdd",  
        "port":45,  
        "policy":"default",  
        "action":0,  
        "dns_address_record_list":[],  
        "dns_mx_record_list":[],  
        "dns cname_record_list":[],  
        "dns_ns_record_list":[],  
        "dns_ptr_record_list":[],  
        "geo_location_list":[]  
      },  
      {  
        "name":"ffffeeee",  
        "port":21,  
        "policy":"default",  
        "action":0,  
        "dns_address_record_list": [  
          {  
            "vip_order":"service1",  
            "as_replace":0,  
            "no_response":0,  
            "static":0,  
            "weight":0  
          },  
          {  
            "vip_order":"service2",  
            "as_replace":0,  
            "no_response":1,  
            "static":1,  
            "weight":0  
          }  
        ],  
        "dns_mx_record_list": [  
          {  
            "name":"fffff",  
            "priority":12  
          },  
          {  
            "name":"ffff1",  
            "priority":12  
          }  
        ],  
        "dns cname_record_list":[],  
        "dns_ns_record_list":[],  
        "dns_ptr_record_list":[],  
        "geo_location_list":[]  
      }  
    ]  
  }  
}
```

CLI configuration after aXAPI call:

```
!
gslb zone CCC1
  ttl 1
  service 45 ffdffdd
  service ftp fffffeeee
    dns-a-record service2 as-replace
    ip-order service1 service2
  dns-mx-record ffff 12
  dns-mx-record ffff1 12
!
```

Response as the HTTP body:

```
{
  "response": {
    "status" : "OK"
  }
}
```

Menus Privilege

Config Mode >> Service >> GSLB >> Zone

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.6.4 “gslb.zone.update” Method

This method is used to update one or more parameters configured for a given GSLB zone.

Parameters

parameter name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	gslb.zone.update	String		
zone(*)	GSLB ZONE object	Object		
name(*)	Name of GSLB zone	String	1 - 31	--
ttl	TTL object of GSLB zone	Object		
status	Status of TTL, enabled (1) or disabled (0)	Integer	0 - 1	1
ttl_timer	TTL timer of TTL object	Integer	0 - 1000000000	10
status	Zone status is enabled(1) or disabled(0)	Bool	0 - 1	1
policy	Policy of GSLB zone	String	1 - 31	default
dns_mx_record_list	DNS MX record list of of GSLB zone	List		
dns_mx_record	DNS MX record	Object		
 name	Name of DNS MX record	String	1 - 127	--
 priority	Priority of DNS MX record	Integer	0 - 65535	--
dns_ns_record_list	DNS NS record list of GSLB zone	List		
dns_ns_record	DNS NS record object	Object		
 name	Name of NS record object	String	1 - 127	--
service_list	Service list of GSLB zone	List		
service	Service object	Object		
 name	Name of service	String	1 - 31	--
 port	Port of service	Integer	0 - 65535	--
 policy	Policy of service	String	1 - 31	--
 action	Action of service 0: do nothing	Integer	0 - 5	0

	1: forward response 2: forward both 3: forward query 4: drop 5: reject			
dns_address_record_list	DNS address record list of GSLB zone	List		
dns_address_record	DNS address record object	Object		
as_replace	As replace option of DNS address record, enabled (1) or disabled (0)	Integer	0 - 1	0
no_response	No response option of DNS address record , enabled (1) or disabled (0)	Integer	0 - 1	0
static	Static option of DNS address record, enabled (1) or disabled (0)	Integer	0 - 1	0
weight	Weight of DNS address record, 0 means disabled	Integer	0 - 100	0
dns_mx_record_list	DNS MX record list of of GSLB zone	List		
dns_mx_record	DNS MX record object	Object		
name	Name of DNS MX record	String	1 - 127	--
priority	Priority of DNS MX record	Integer	0 - 65535	0
dns_cname_record_list	DNS CName record list of GSLB zone	List		
dns_cname_record	DNS CName record object	Object		
name	Name of DNS CName record object	String	1 - 127	--
dns_ns_record_list	DNS NS record list of of GSLB zone	List		
dns_ns_record	DNS NS record object	Object		
name	Name of DNS NS record	String	1 - 127	--
dns_ptr_record_list	DNS PTR record list	List		
dns_ptr_record	DNS PTR record object	Object		
name	Name of DNS PTR record object	String	1 - 127	--

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- **URL:**
[https://\[AX_IP_ADDRESS\]/services/rest/V2/](https://[AX_IP_ADDRESS]/services/rest/V2/)
- **HTTP Action:**
POST

CLI configuration before aXAPI call:

```
!
gslb zone CCC1
  ttl 1
  service 45 ffdffdd
  service ftp fffffeeee
    dns-a-record service2 as-replace
    ip-order service1 service2
    dns-mx-record ffff 12
    dns-mx-record ffff1 12
!
!
```

1. HTTP Request in URL Format:

```
session_id=xxxxxxxxxxxxxxxxxxxxx&method=gslb.zone.update&zone=name%03CCC1%02ttl%031
00%02policy%03default%02dns_mx_record_list%02dns_ns_record_list%02service_list&dns_mx_rec
ord_list=m1%02m2%02m3&m1=name%031234%02priority%03123&m2=name%031235%02priority%03321&m3=
name%031237%02priority%0332&dns_ns_record_list=n1%02n2&n1=name%03dddddd&n2=name%03fffff&
service_list=s11%02s12&s11=name%03ffffd%02port%0345%02policy%03default&s12=name%03ffffee
ee%02port%0321%02policy%03default%02dns_mx_record_list%02dns_address_record_list&dns_mx_r
ecord_list=mx1%02mx2&mx1=name%03ffff%02priority%0320&mx2=name%03ffff1%02priority%0330&dns
address_record_list=j1%02j2&j1=vip_order%03service1&j2=vip_order%03service2%02as_replace
%031&format=url
```

Response as the HTTP body (URL Format):

```

<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>

```

2. HTTP Request in JSON Format:

session_id=xxxxxxxxxxxxxxxxxxxx&method=gslb.zone.update&format=json

Request POST data (JSON Format):

```

{
  "zone": [
    {
      "name": "ccc1",
      "ttl": 100,
      "policy": "default",
      "disable_all_services": 0,
      {
        "dns_mx_record_list": []
      },
      {
        "dns_ns_record_list": []
      },
      "service_list": [
        {
          "name": "ffdfdd",
          "port": 45,
          "policy": "default",
          "action": 0,
          "dns_address_record_list": [],
          "dns_mx_record_list": [],
          "dns_cname_record_list": [],
          "dns_ns_record_list": [],
          "dns_ptr_record_list": [],
          "geo_location_list": []
        },
        {
          "name": "ffffeeee",
          "port": 21,
          "policy": "default",
          "action": 0,
          "dns_address_record_list": [
            {
              "vip_order": "service1",
              "as_replace": 0,
              "no_response": 0,
              "static": 0,
              "weight": 0
            },
            {
              "vip_order": "service2",
              "as_replace": 0,
              "no_response": 1,
              "static": 1,
              "weight": 0
            }
          ],
          "dns_mx_record_list": [
            {
              "name": "fffff",
              "priority": 20
            },
            {

```

```

        "name":"ffff1",
        "priority":30
    },
],
"dns cname record list":[],
"dns ns record list":[],
"dns ptr record list":[],
"geo location list":[]
}
]
}
}

```

CLI configuration after aXAPI call:

```
!
gslb zone CCC1
  ttl 100
  service 45 ffdffdd
  service ftp fffffeeee
    dns-a-record service2 as-replace
    ip-order service1 service2
  dns-mx-record ffff 20
  dns-mx-record ffff1 30
!
```

Response as the HTTP body:

```
{
  "response":
  {
    "status" : "OK"
  }
}
```

Menus Privilege

Config Mode >> Service >> GSLB >> Zone

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.6.5 “gslb.zone.delete” Method

This method is used to delete a GSLB zone from the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	gslb.zone.delete	String		
name (*)	the name of the gslb zone	String		

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:

- [https://\[AX_IP_ADDRESS\]/services/rest/V2/](https://[AX_IP_ADDRESS]/services/rest/V2/)
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
gslb zone zone01
  ttl 1
  service 45 ffdffffd
  service ftp fffffeee
    dns-a-record service2 as-replace
    ip-order service1 service2
  dns-mx-record fffff 12
  dns-mx-record fffff1 120
!
```

1. HTTP Request in URL Format:

session_id=xxxxxxxxxxxxxxxxxxxx&method=gslb.zone.delete&name=zone01&format=url

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in JSON Format:

session_id=xxxxxxxxxxxxxxxxxxxx&method=gslb.zone.delete&format=json

Request POST data (JSON Format):

```
{
  "name" : "zone01"
}
```

CLI configuration after aXAPI call:

```
!
!
```

Response as the HTTP body:

```
{
  "response": {
    {
      "status" : "OK"
    }
  }
}
```

Menus Privilege

Config Mode >> Service >> GSLB >> Zone

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.6.6 “**gslb.zone.service.create**” Method

This method is used to add a service object to a GSLB zone.

Parameters

parameter name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	gslb.zone.service.create	String		
name(*)	Name of GSLB zone	String	1 - 31	--
service(*)	Service object	Object		
name(*)	Name of service	String	1 - 31	--
port	Port of service	Integer	0 – 65535	--
policy	Policy of service	String	1 - 31	--
action	Action of service 0: do nothing 1: forward response 2: forward both 3: forward query 4: drop 5: reject	Integer	0 - 5	0
dns_address_record_list	DNS address record list of GSLB zone	List		
dns_address_record	DNS address record object	Object		
as_replace	As replace option of DNS address record, enabled (1) or disabled (0)	Integer	0 - 1	0
no_response	No response option of DNS address record , enabled (1) or disabled (0)	Integer	0 - 1	0
static	Static option of DNS address record, enabled (1) or disabled (0)	Integer	0 - 1	0
weight	Weight of DNS address record, 0 means disabled	Integer	0 - 100	0
dns_mx_record_list	DNS MX record list of of GSLB zone	List		
dns_mx_record	DNS MX record object	Object		
name	Name of DNS MX record	String	1 - 127	--
priority	Priority of DNS MX record	Integer	0 - 65535	0
dns cname_record_list	DNS CName record list of GSLB zone	List		
dns cname_record	DNS CName record object	Object		
name	Name of DNS CName record object	String	1 - 127	--
dns_ns_record_list	DNS NS record list of of GSLB zone	List		
dns_ns_record	DNS NS record object	Object		
name	Name of DNS NS record	String	1 – 127	--
dns_ptr_record_list	DNS PTR record list	List		
dns_ptr_record	DNS PTR record object	Object		
name	Name of DNS PTR record object	String	1 - 127	--
dns_srv_record_list	DNS server record list	List	--	--
dns_srv_record	DNS server record	Object	--	--
name	DNS server name	String	1 – 127	--
port	DNS server port (value scope is 0 – 65535, 65535 means not set)	Integer	0 - 65535	65535 (not set)
priority	Priority of this DNS service	Integer	0 - 65535	0
weight	Weight of this DNS service	Integer	0 - 65535	0
dns_txt_record_list	DNS TXT record list	List	--	--
dns_txt_record	DNS TXT record	Object	--	--
txt	Text of this DNS TXT record	String	1 - 511	--
geo_location_list(**)	Geography location list	List	--	--
geo_location	Geography location	Object	--	--
geo_location	geography location string	String	1 - 127	--
alias	DNS alias	String	1 - 1024	--
geo_action	Action when this geo-location is hit (0: do nothing, 1: forward response, 2: forward both query and response, 3: forward query, 4: drop, 5: reject, 6: allow)	Integer	0 – 6	0
policy	Profile name for this GSLB policy used to handle this geo-location.	String	1 – 127	--

Note: (*) parameter is required in the API.

(**) The geo-location parameter ‘geo_action’ and ‘policy’ are mutually exclusive. If both are configured at the same time, the ‘policy’ parameter takes effect.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
gslb zone abc
!
```

1. HTTP Request in URL Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=gslb.zone.service.create&name=abc&service=name%03sv08%02port%0325%02dns_mx_record_list&dns_mx_record_list=11%0212%0213&11=name%03bbb%02priority%0310&12=name%03bbc%02priority%039&13=name%03bbec%02priority%039&format=url
```

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in JSON Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=gslb.zone.service.create&format=json
```

Request POST data (JSON Format):

```
{
  "name": "abc",
  "service": {
    {
      "name": "sv08",
      "port": 25,
      "dns_mx_record_list": [
        {
          "name": "bbb",
          "priority": 10
        },
        {
          "name": "bbc",
          "priority": 9
        },
        {
          "name": "bbec",
          "priority": 9
        }
      ]
    }
  }
}
```

CLI configuration after aXAPI call:

```
!
```

```

gslb zone abc
    service smtp sv08
        dns-mx-record bbc 9
        dns-mx-record bbec 9
        dns-mx-record bbb 10
!

```

Response as the HTTP body:

```
{
    "response":
    {
        "status" : "OK"
    }
}
```

Menus Privilege

Config Mode >> Service >> GSLB >> Zone

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.6.7 “gslb.zone.service.update” Method

This method is used to update one or more service parameters for a given GSLB zone.

Parameters

parameter name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	gslb.zone.service.update	String		
name(*)	Name of GSLB zone	String	1 - 31	--
service(*)	Service object	Object		
name(*)	Name of service	String	1 - 31	--
port	Port of service	Integer	0 - 65535	--
policy	Policy of service	String	1 - 31	--
action	Action of service 0: do nothing 1: forward response 2: forward both 3: forward query 4: drop 5: reject	Integer	0 - 5	0
dns_address_record_list	DNS address record list of GSLB zone	List		
dns_address_record	DNS address record object	Object		
as_replace	As replace option of DNS address record, enabled (1) or disabled (0)	Integer	0 - 1	0
no_response	No response option of DNS address record , enabled (1) or disabled (0)	Integer	0 - 1	0
static	Static option of DNS address record, enabled (1) or disabled (0)	Integer	0 - 1	0
weight	Weight of DNS address record, 0 means disabled	Integer	0 - 100	0
dns_mx_record_list	DNS MX record list of of GSLB zone	List		
dns_mx_record	DNS MX record object	Object		
name	Name of DNS MX record	String	1 - 127	--
priority	Priority of DNS MX record	Integer	0 - 65535	0
dns_cname_record_list	DNS CName record list of GSLB zone	List		
dns_cname_record	DNS CName record object	Object		
name	Name of DNS CName record object	String	1 - 127	--
dns_ns_record_list	DNS NS record list of of GSLB zone	List		

dns_ns_record	DNS NS record object	Object		
name	Name of DNS NS record	String	1 - 127	--
dns_ptr_record_list	DNS PTR record list	List		
dns_ptr_record	DNS PTR record object	Object		
name	Name of DNS PTR record object	String	1 - 127	--
dns_srv_record_list	DNS server record list	List	--	--
dns_srv_record	DNS server record	Object	--	--
name	DNS server name	String	1 - 127	--
port	DNS server port	Integer	0 - 65535	65535 (not set)
priority	Priority of this DNS service	Integer	0 - 65535	0
weight	Weight of this DNS service	Integer	0 - 65535	0
dns_txt_record_list	DNS TXT record list	List	--	--
dns_txt_record	DNS TXT record	Object	--	--
txt	Text of this DNS TXT record	String	1 - 511	--
geo_location_list(**)	Geography location list	List	--	--
geo_location	Geography location	Object	--	--
geo_location	geography location string	String	1 - 127	--
alias	DNS alias	String	1 - 1024	--
geo_action	Action when this geo-location is hit (0: do nothing, 1: forward response, 2: forward both query and response, 3: forward query, 4: drop, 5: reject, 6: allow)	Integer	0 - 6	0
policy	Profile name of GSLB policy used to handle this geo-location	String	1 - 127	--

Note: (*) parameter is required in the API.

(**) The geo-location parameter ‘geo_action’ and ‘policy’ are mutually exclusive. If both are configured at the same time, the ‘policy’ parameter takes effect.

Response Fields

General XML status response

Example Response

Request as:

- **URL:**
https://[AX_IP_ADDRESS]/services/rest/V2/
- **HTTP Action:**
POST

CLI configuration before aXAPI call:

```
!
gslb zone abc
    service smtp sv08
        dns-mx-record bbc 9
        dns-mx-record bbec 9
        dns-mx-record bbb 10
!
```

1. HTTP Request in URL Format:

session_id=xxxxxxxxxxxxxxxxxxxx&method=gslb.zone.service.update&name=abc&service=name%03sv08%02port%0325%02dns_mx_record_list&dns_mx_record_list=11%0212%0213%0214&l1=name%03bbb%02priority%0310&l2=name%03bbc%02priority%039&l3=name%03bbec%02priority%039&l4=name%03bbd%02priority%0311&format=url

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in JSON Format:

session_id=xxxxxxxxxxxxxxxxxxxx&method=gslb.zone.service.update&format=json

Request POST data (JSON Format):

```
{  
    "name": "abc",  
    "service":  
    {  
        "name": "sv08",  
        "port": 25,  
        "dns_mx_record_list": [  
            {  
                "name": "bbb",  
                "priority": 10  
            },  
            {  
                "name": "bbc",  
                "priority": 9  
            },  
            {  
                "name": "bbec",  
                "priority": 9  
            },  
            {  
                "name": "bbd",  
                "priority": 11  
            }  
        ]  
    }  
}
```

CLI configuration after aXAPI call:

```
!  
gslb zone abc  
    service smtp sv08  
        dns-mx-record bbc 9  
        dns-mx-record bbec 9  
        dns-mx-record bbb 10  
        dns-mx-record bbd 11  
!  
!
```

Response as the HTTP body:

```
{  
    "response":  
    {  
        "status": "OK"  
    }  
}
```

Menus Privilege

Config Mode >> Service >> GSLB >> Zone

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.6.8 “gslb.zone.service.search” Method

This method is used to search for a service in a specific GSLB zone.

Parameters

parameter name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	gslb.zone.service.delete	String		
zone_name (*)	Name of GSLB zone	String	1 – 63	--
service_name (*)	Name of service of GSLB zone	string	1 - 63	--

Note: (*) parameter is required in the API.

Response Fields

service	Service object	Object		
name	Name of service	String	1 - 31	--
port	Port of service	Integer	0 - 65535	--
policy	Policy of service	String	1 - 31	--
action	Action of service 0: do nothing 1: forward response 2: forward both 3: forward query 4: drop 5: reject	Integer	0 - 5	0
dns_address_record_list	DNS address record list of GSLB zone	List		
dns_address_record	DNS address record object	Object		
as_replace	As replace option of DNS address record, enabled (1) or disabled (0)	Integer	0 - 1	0
no_response	No response option of DNS address record , enabled (1) or disabled (0)	Integer	0 - 1	0
static	Static option of DNS address record, enabled (1) or disabled (0)	Integer	0 - 1	0
weight	Weight of DNS address record, 0 means disabled	Integer	0 - 100	0
dns_mx_record_list	DNS MX record list of of GSLB zone	List		
dns_mx_record	DNS MX record object	Object		
name	Name of DNS MX record	String	1 - 127	--
priority	Priority of DNS MX record	Integer	0 - 65535	0
dns_cname_record_list	DNS CName record list of GSLB zone	List		
dns_cname_record	DNS CName record object	Object		
name	Name of DNS CName record object	String	1 - 127	--
dns_ns_record_list	DNS NS record list of of GSLB zone	List		
dns_ns_record	DNS NS record object	Object		
name	Name of DNS NS record	String	1 – 127	--
dns_ptr_record_list	DNS PTR record list	List	--	--
dns_ptr_record	DNS PTR record object	Object	--	--
name	Name of DNS PTR record object	String	1 - 127	--
dns_srv_record_list	DNS server record list	List	--	--
dns_srv_record	DNS server record	Object	--	--
name	DNS server name	String	1 – 127	--
port	DNS server port	Integer	0 - 65535	65535 (not set)
priority	Priority of this DNS service	Integer	0 - 65535	0
weight	Weight of this DNS service	Integer	0 - 65535	0
dns_txt_record_list	DNS TXT record list	List	--	--
dns_txt_record	DNS TXT record	Object	--	--
txt	Text of this DNS TXT record	String	1 - 511	--
geo_location_list(*)	Geography location list	List	--	--
geo_location	Geography location	Object	--	--
geo_location	Geography location string	String	1 - 127	--
alias	DNS alias	String	1 - 1024	--
geo_action	Action when this geo-location is hit (0: do nothing, 1: forward response, 2: forward both query and response, 3: forward query, 4: drop, 5: reject, 6: allow)	Integer	0 – 6	0

policy	Profile name of GSLB policy used for this geo-location	String	1 – 127	--
---------------	--	--------	---------	----

Note: (*)The geo-location parameter ‘geo_action’ and ‘policy’ are mutually exclusive. If both are configured at the same time, the ‘policy’ parameter takes effect.

Example

Request as:

- URL:
- https://192.168.101.134:443/services/rest/V2/?session_id=14cd29a17e4bac85f26c16bb45d7ed&format=json&method=gslb.zone.service.search&zone_name=h1&service_name=s_http1
- HTTP Action:
GET

Response

```
{
    "service": {
        "name": "service01",
        "port": 69,
        "policy": "default",
        "action": 1,
        "status": 1,
        "dns_address_record_list": [
            {
                "vip_order": "aaa",
                "as_replace": 0,
                "no_response": 0,
                "static": 0,
                "as_backup": 0,
                "ttl": 0,
                "weight": 0
            }
        ],
        "dns_mx_record_list": [
            {
                "name": "a10.com",
                "priority": 1
            }
        ],
        "dns cname_record_list": [
            {
                "name": "a10.com",
                "admin_preference": 100,
                "as_backup": 0,
                "weight": 1
            }
        ],
        "dns_ns_record_list": [
            {
                "name": "a10.com"
            }
        ],
        "dns_ptr_record_list": [
            {
                "name": "a10.com"
            }
        ],
        "geo_location_list": [
            {
                "geo_location": "a10.com",
                "alias": "",
                "action": 1,
                "policy": ""
            }
        ],
        "dns_srv_record_list": [
            {
                "name": "a10.com",
                "priority": 2,
                "weight": 1
            }
        ]
    }
}
```

```

        "weight": 10,
        "port_defined": 0
    }
],
"dns_txt_record_list": []
}
}

```

CLI config before call

```

gslb zone zone01
dns-mx-record abc 0
dns-ns-record a10.com
service tftp service01
action forward response
dns-cname-record a10.com
ip-order aaa
dns-mx-record a10.com 1
dns-ns-record a10.com
dns-ptr-record a10.com
dns-srv-record a10.com 2
geo-location a10.com action forward response

```

Menus Privilege

Config Mode >> Service >> GSLB >> Zone

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.6.9 “gslb.zone.service.delete” Method

This method is used to delete a service from a specific GSLB zone.

Parameters

parameter name	Description	Data Type	Range	Default
session_id (*)	User authenticated session id	String		
method (*)	gslb.zone.service.delete	String		
name(*)	Name of GSLB zone	String	1 - 31	--
service(*)	Service object	Object		
name(*)	Name of service	String	1 - 31	--

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://[AX_IP_ADDRESS]/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
gslb zone abc
  service smtp sv08
    dns-mx-record bbc 9
    dns-mx-record bbec 9
    dns-mx-record bbb 10
    dns-mx-record bbd 11
!
```

1. HTTP Request in URL Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=gslb.zone.service.delete&name=abc&service=name%03sv08&format=url
```

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in JSON Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=gslb.zone.service.delete&format=json
```

Request POST data (JSON Format):

```
{
  "name": "abc",
  "service": [
    {
      "name": "sv08"
    }
}
```

Response as the HTTP body (JSON Format):

```
{
  "response": [
    {
      "status": "OK"
    }
}
```

CLI configuration after aXAPI call:

```
!
gslb zone abc
!
```

Menus Privilege

Config Mode >> Service >> GSLB >> Zone

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.6.10 “gslb.zone.service.dns_txt_record.create” Method

Method name

gslb.zone.service.dns_txt_record.create

Description

This method offers a quick way to add DNS txt records into the service of a GSLB zone. You can achieve the same goal by calling methods ‘gslb.zone.*’ or ‘gslb.service.*’.

Parameters

Table Parameters of main

parameter name	Description	Data Type	Range	Default
session_id (*)	User authenticated session id	String		
method (*)	gslb.zone.service.dns_txt_record.create	String		
zone_name(*)	Name of GSLB zone	String	1 - 31	--
service_name(*)	Name of Service	String	1 - 31	
dns_txt_record_list (*)	List of DNS txt record objects, see ‘Table Parameters of dns txt record’	List	1 - 31	--

Table Parameters of dns txt record

Parameter name	Description	Data Type	Range	Default
txt (*)	Text string of dns txt record.	String	1 - 255	

Note: (*) parameter is required in the API.

Response Fields

General AXAPI status response

Example

URL	https://[AX host name or IP address]:[service port]/services/rest/V2/?session_id=[your session id]&format=json&method=gslb.zone.service.dns_txt_record.create
HTTP Action	POST
HTTP Post Body	{ "zone_name": "zone01", "service_name": "service01", "dns_txt_record_list": [{ "txt": "AAA=aaa a10.com=1.0" }, { "txt": "This is a sample" }] }
Response	{"response": {"status": "OK"}}
CLI Config Before Call	gslb zone zone01 service ftp service01
CLI Config After Call	gslb zone zone01 service ftp service01 dns-txt-record AAA=aaa a10.com=1.0 dns-txt-record This is a sample

Note: By default we use JSON format.

Menus Privilege

Config Mode >> Service >> GSLB >> Zone

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.6.11 “gslb.zone.service.dns_txt_record.update” Method

Method name

gslb.zone.service.dns_txt_record.update

Description

This method offers a quick way to update DNS txt records for a service in a GSLB zone. You can achieve the same goal by calling methods ‘gslb.zone.*’ or ‘gslb.service.*’.

Note: This method completely replaces the old list with the new list. If you only want to update specific items in the list, please delete the items by gslb.zone.service.dns_txt_record.delete and then add the new items with gslb.zone.service.dns_txt_record.add.

Parameters

Table Parameters of main

parameter name	Description	Data Type	Range	Default
session_id (*)	User authenticated session id	String		
method (*)	gslb.zone.service.dns_txt_record.update	String		
zone_name(*)	Name of GSLB zone	String	1 - 31	--
service_name(*)	Name of Service	String	1 - 31	
dns_txt_record_list (*)	List of DNS txt record objects, see 'Table Parameters of dns txt record'	List	1 - 31	--

Table Parameters of dns txt record

Parameter name	Description	Data Type	Range	Default
txt (*)	Text string of dns txt record.	String	1 - 255	

Note: (*) parameter is required in the API.

Response Fields

General AXAPI status response

Example

URL	https://[AX host name or IP address]:[service port]/services/rest/V2/?session_id=[your session id]&format=json&method=gslb.zone.service.dns_txt_record.update
HTTP Action	POST
HTTP Post Body	{ "zone_name": "zone01", "service_name": "service01", "dns_txt_record_list": [{ "txt": "AAA=aaa a10.com=1.0" }, { "txt": "This is a sample" }] }
Response	{"response": {"status": "OK"}}
CLI Config Before Call	gslb zone zone01 service ftp service01 dns-txt-record abcdef
CLI Config After Call	gslb zone zone01 service ftp service01 dns-txt-record AAA=aaa a10.com=1.0

	dns-txt-record This is a sample
--	---------------------------------

Note: By default we use JSON format.

Menus Privilege

Config Mode >> Service >> GSLB >> Zone

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.6.12 “gslb.zone.service.dns_txt_record.delete” Method

Method name

gslb.zone.service.dns_txt_record.delete

Description

This method offers a quick way to delete DNS txt records from the service of a GSLB zone. You can achieve the same goal by calling methods ‘gslb.zone.*’ or ‘gslb.service.*’.

Parameters

Table Parameters of main

Parameter name	Description	Data Type	Range	Default
session_id (*)	User authenticated session id	String		
method (*)	gslb.zone.service.dns_txt_record.delete	String		
zone_name(*)	Name of GSLB zone	String	1 - 31	--
service_name(*)	Name of Service	String	1 - 31	--
dns_txt_record_list (*)	List of DNS txt record objects, see ‘Table Parameters of dns txt record’	List	1 - 31	--

Table Parameters of dns txt record

Parameter name	Description	Data Type	Range	Default
txt (*)	Text string of DNS txt record.	String	1 - 255	

Note: (*) parameter is required in the API.

Response Fields

General AXAPI status response

Example

URL	https://[AX host name or IP address]:[service port]/services/rest/V2/?session_id=[your session id]&format=json&method=gslb.zone.service.dns_txt_record.delete
HTTP Action	POST
HTTP Post Body	{ "zone_name": "zone01", "service_name": "service01", "dns_txt_record_list": [{ "txt": "AAA=aaa a10.com=1.0" }, { "txt": "This is a sample" }] }
Response	{"response": {"status": "OK"}}
CLI Config Before Call	gslb zone zone01

	service ftp service01 dns-txt-record AAA=aaa a10.com=1.0 dns-txt-record This is a sample
CLI Config After Call	gslb zone zone01 service ftp service01

Note: By default we use JSON format.

Menus Privilege

Config Mode >> Service >> GSLB >> Zone

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.6.13 “gslb.zone.service.dns_txt_record.getAll” Method

Method name

gslb.zone.service.dns_txt_record.getAll

Description

This method offers a quick way to get all DNS txt records for a service in a GSLB zone. You can achieve the same goal by calling methods ‘gslb.zone.*’ or ‘gslb.service.*’.

Parameters

parameter name	Description	Data Type	Range	Default
session_id (*)	User authenticated session id	String		
method (*)	gslb.zone.service.dns_txt_record.getAll	String		
zone_name(*)	Name of GSLB zone	String	1 - 31	--
service_name(*)	Name of Service	String	1 - 31	

Response Fields

Table Response fields of main

field name	Description	Data Type	Range	Default
dns_txt_record_list (*)	List of DNS txt record objects, see ‘Response fields of DNS txt record’	List	1 - 31	--

Table Response fields of dns txt record

Parameter name	Description	Data Type	Range	Default
txt (*)	Text string of DNS txt record.	String	1 - 255	

Example

URL	https://[AX host name or IP address]:[service port]/services/rest/V2/?session_id=[your session id]&format=json&method=gslb.zone.service.dns_txt_record.getAll
HTTP Action	POST
HTTP Post Body	{ "zone_name": "zone01", "service_name": "service01" }
Response	{ "dns_txt_record_list": [{ "txt": "AAA=aaa a10.com=1.0" }, { "txt": "This is a sample" }] }

	<pre> }] } </pre>
CLI Config Before Call	gslb zone zone01 service ftp service01 dns-txt-record AAA=aaa a10.com=1.0 dns-txt-record This is a sample
CLI Config After Call	No change

Note: By default we use JSON format.

Menus Privilege

Config Mode >> Service >> GSLB >> Zone

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.6.14 “gslb.zone.service.dns_txt_record.deleteAll” Method

Method name

gslb.zone.service.dns_txt_record.deleteAll

Description

This method offers a quick way to delete all DNS txt records for a service in a GSLB zone. You can achieve the same goal by calling methods ‘gslb.zone.*’ or ‘gslb.service.*’.

Parameters

parameter name	Description	Data Type	Range	Default
session_id (*)	User authenticated session id	String		
method (*)	gslb.zone.service.dns_txt_record.deleteAll	String		
zone_name(*)	Name of GSLB zone	String	1 - 31	--
service_name(*)	Name of Service	String	1 - 31	

Response Fields

General AXAPI status response

Example

URL	https://[AX host name or IP address]:[service port]/services/rest/V2/?session_id=[your session id]&format=json&method=gslb.zone.service.dns_txt_record.deleteAll
HTTP Action	POST
HTTP Post Body	{ "zone_name": "zone01", "service_name": "service01" }
Response	{"response": {"status": "OK"}}
CLI Config Before Call	gslb zone zone01 service ftp service01 dns-txt-record AAA=aaa a10.com=1.0 dns-txt-record This is a sample
CLI Config After Call	gslb zone zone01 service ftp service01

Note: By default we use JSON format.

Menus Privilege

Config Mode >> Service >> GSLB >> Zone

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.6.15 “gslb.zone.service.dns_mx_record.create” Method

Method name

gslb.zone.service.dns_mx_record.create

Description

This method offers a quick way to add DNS MX records into a service for a GSLB zone. You can achieve the same goal by calling methods ‘gslb.zone.*’ or ‘gslb.service.*’.

Parameters

Table Parameters of main

parameter name	Description	Data Type	Range	Default
session_id (*)	User authenticated session id	String		
method (*)	gslb.zone.service.dns_mx_record.create	String		
zone_name(*)	Name of GSLB zone	String	1 - 31	--
service_name(*)	Name of Service	String	1 - 31	--
dns_mx_record_list (*)	List of DNS mx record objects, see ‘Table Parameters of DNS mx record’	List	1 - 31	--

Table Parameters of dns mx record

Parameter name	Description	Data Type	Range	Default
name (*)	Name of DNS mx record.	String	1 – 127	
priority(*)	Priority of DNS mx record	Integer	0 - 65535	

Note: (*) parameter is required in the API.

Response Fields

General AXAPI status response

Example

URL	https://[AX host name or IP address]:[service port]/services/rest/V2/?session_id=[your session id]&format=json&method=gslb.zone.service.dns_mx_record.create
HTTP Action	POST
HTTP Post Body	{ "zone_name": "zone03", "service_name": "111", "dns_mx_record_list": [{ "name": "ABC", "priority": 12 }, { "name": "CDE", "priority": 20 }] }
Response	{"response": {"status": "OK"}}

CLI Config Before Call	gslb zone zone01 service ftp service01
CLI Config After Call	gslb zone zone01 service ftp service01 dns-mx-record abc 12 dns-mx-record cde 20

Note: By default we use JSON format.

Menus Privilege

Config Mode >> Service >> GSLB >> Zone

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.6.16 “gslb.zone.service.dns_mx_record.update” Method

Method name

gslb.zone.service.dns_mx_record.update

Description

This method offers a quick way to update DNS MX records for a service in a GSLB zone. You can achieve the same goal by calling methods ‘gslb.zone.*’ or ‘gslb.service.*’.

Note: This method completely replaces the old list by the new list. If you only want to update specific items in the list, please delete the items with gslb.zone.service.dns_mx_record.delete and then add new items with gslb.zone.service.dns_mx_record.add.

Parameters

Table Parameters of main

parameter name	Description	Data Type	Range	Default
session_id (*)	User authenticated session id	String		
method (*)	gslb.zone.service.dns_mx_record.update	String		
zone_name(*)	Name of GSLB zone	String	1 - 31	--
service_name(*)	Name of Service	String	1 - 31	
dns_mx_record_list (*)	List of DNS mx record objects, see ‘Table Parameters of DNS mx record’	List	1 - 31	--

Table Parameters of dns mx record

Parameter name	Description	Data Type	Range	Default
name (*)	Name of DNS mx record.	String	1 – 127	
priority(*)	Priority of DNS mx record	Integer	0 - 65535	

Note: (*) parameter is required in the API.

Response Fields

General AXAPI status response

Example

URL	https://[AX host name or IP address]:[service port]/services/rest/V2/?session_id=[your session id]&format=json&method=gslb.zone.service.dns_mx_record.update
HTTP Action	POST
HTTP Post Body	{

	<pre>"zone_name": "zone03", "service_name": "111", "dns_mx_record_list": [{ "name": "ABC", "priority": 12 }, { "name": "CDE", "priority": 20 }]</pre>
Response	{"response": {"status": "OK"}}
CLI Config Before Call	<pre>gslb zone zone01 service ftp service01 dns-mx-record abc 10 dns-mx-record xyz 10 dns-mx-record opq 10</pre>
CLI Config After Call	<pre>gslb zone zone01 service ftp service01 dns-mx-record abc 12 dns-mx-record cde 20</pre>

Note: By default we use JSON format.

Menus Privilege

Config Mode >> Service >> GSLB >> Zone

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.6.17 “gslb.zone.service.dns_mx_record.delete” Method

Method name

gslb.zone.service.dns_mx_record.delete

Description

This method offers a quick way to delete DNS MX records for a service in a GSLB zone. You can achieve the same goal by calling methods ‘gslb.zone.*’ or ‘gslb.service.*’.

Parameters

Table Parameters of main

Parameter name	Description	Data Type	Range	Default
session_id (*)	User authenticated session id	String		
method (*)	gslb.zone.service.dns_mx_record.delete	String		
zone_name(*)	Name of GSLB zone	String	1 - 31	--
service_name(*)	Name of Service	String	1 - 31	
dns_mx_record_list (*)	List of DNS MX record objects, see ‘Table Parameters of DNS MX record’	List	1 - 31	--

Table Parameters of dns mx record

Parameter name	Description	Data Type	Range	Default
name (*)	Name of DNS MX record.	String	1 – 127	
priority(*)	Priority of DNS MX record	Integer	0 - 65535	

Note: (*) parameter is required in the API.

Response Fields

General AXAPI status response

Example

URL	https://[AX host name or IP address]:[service port]/services/rest/V2/?session_id=[your session id]&format=json&method=gslb.zone.service.dns_mx_record.delete
HTTP Action	POST
HTTP Post Body	{ "zone_name": "zone03", "service_name": "111", "dns_mx_record_list": [{ "name": "ABC", "priority": 12 }, { "name": "CDE", "priority": 20 }] }
Response	{"response": {"status": "OK"}}
CLI Config Before Call	gslb zone zone01 service ftp service01 dns-mx-record abc 12 dns-mx-record cde 20 dns-mx-record xyz 10 dns-mx-record opq 10
CLI Config After Call	gslb zone zone01 service ftp service01 dns-mx-record xyz 10 dns-mx-record opq 10

Note: By default we use JSON format.

Menus Privilege

Config Mode >> Service >> GSLB >> Zone

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.6.18 “gslb.zone.service.dns_mx_record.getAll” Method

Method name

gslb.zone.service.dns_mx_record.getAll

Description

This method offers a quick way to get all DNS MX records for a service in a GSLB zone. You can achieve the same goal by calling methods ‘gslb.zone.*’ or ‘gslb.service.*’.

Parameters

parameter name	Description	Data Type	Range	Default
session_id (*)	User authenticated session id	String		
method (*)	gslb.zone.service.dns_mx_record.getAll	String		

zone_name(*)	Name of GSLB zone	String	1 - 31	--
service_name(*)	Name of Service	String	1 - 31	

Response Fields

Table Response fields of main

Field name	Description	Data Type	Range	Default
dns_mx_record_list (*)	List of DNS MX record objects, see 'Response fields of dns mx record'	List	1 - 31	--

Table Response fields of dns mx record

Fields name	Description	Data Type	Range	Default
name (*)	Name of DNS MX record.	String	1 – 127	
priority(*)	Priority of DNS MX record	Integer	0 - 65535	

Example

URL	https://[AX host name or IP address]:[service port]/services/rest/V2/?session_id=[your session id]&format=json&method=gslb.zone.service.dns_mx_record.getAll
HTTP Action	POST
HTTP Post Body	{ "zone_name": "zone01", "service_name": "service01" }
Response	{ "dns_mx_record_list": [{ "name": "abc", "priority": 12 }, { "name": "cde", "priority": 20 }] }
CLI Config Before Call	gslb zone zone01 service ftp service01 dns-mx-record abc 12 dns-mx-record cde 20
CLI Config After Call	No change

Note: By default we use JSON format.

Menus Privilege

Config Mode >> Service >> GSLB >> Zone

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.6.19 “gslb.zone.service.dns_mx_record.deleteAll” Method

Method name

gslb.zone.service.dns_mx_record.deleteAll

Description

This method offers a quick way to delete all DNS MX records for a service in a GSLB zone. You can achieve the same goal by calling methods ‘gslb.zone.*’ or ‘gslb.service.*’.

Parameters

parameter name	Description	Data Type	Range	Default
session_id (*)	User authenticated session id	String		
method (*)	gslb.zone.service.dns_mx_record.deleteAll	String		
zone_name(*)	Name of GSLB zone	String	1 - 31	--
service_name(*)	Name of Service	String	1 - 31	

Response Fields

General AXAPI status response

Example

URL	https://[AX host name or IP address]:[service port]/services/rest/V2/?session_id=[your session id]&format=json&method=gslb.zone.service.dns_mx_record.deleteAll
HTTP Action	POST
HTTP Post Body	{ "zone_name": "zone01", "service_name": "service01" }
Response	{"response": {"status": "OK"}}
CLI Config Before Call	gslb zone zone01 service ftp service01 dns-mx-record abc 12 dns-mx-record cde 20
CLI Config After Call	gslb zone zone01 service ftp service01

Note: By default we use JSON format.

Menus Privilege

Config Mode >> Service >> GSLB >> Zone

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.6.20 “gslb.zone.service.dns_mx_record.search” Method

Method name

gslb.zone.service.dns_mx_record.search

Description

This method offers a quick way to get specific DNS MX records for a service in a GSLB zone. You can achieve the same goal by calling methods ‘gslb.zone.*’ or ‘gslb.service.*’.

Parameters

parameter name	Description	Data Type	Range	Default
session_id (*)	User authenticated session id	String		
method (*)	gslb.zone.service.dns_mx_record.search	String		
zone_name(*)	Name of GSLB zone	String	1 - 31	--
service_name(*)	Name of Service	String	1 - 31	
mx_name(*)	Name of the MX record	String	1 - 127	

Response Fields

General AXAPI status response

Example

URL	https://[AX host name or IP address]:[service port]/services/rest/V2/?session_id=[your session id]&format=json&method=gslb.zone.service.dns_mx_record.search
HTTP Action	POST
HTTP Post Body	{ "zone_name": "zone01", "service_name": "service01", "mx_name": "abc" }
Response	{ "dns_mx_record": { "name": "abc", "priority": 12 } }
CLI Config Before Call	gslb zone zone01 service ftp service01 dns-mx-record abc 12 dns-mx-record cde 20
CLI Config After Call	No change

Note: By default we use JSON format.

Menus Privilege

Config Mode >> Service >> GSLB >> Zone

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.6.21 “gslb.zone.service.dns_cname_record.create” **Method**

Method name

gslb.zone.service.dns_cname_record.create

Description

This method offers a quick way to add DNS CNAME records for a service in a GSLB zone. You can achieve the same goal by calling methods ‘gslb.zone.*’ or ‘gslb.service.*’.

Parameters

Table Parameters of main

parameter name	Description	Data Type	Range	Default
session_id (*)	User authenticated session id	String		
method (*)	gslb.zone.service.dns_cname_record.create	String		
zone_name(*)	Name of GSLB zone	String	1 - 31	--
service_name(*)	Name of Service	String	1 - 31	
dns_cname_record_list (*)	List of DNS cname record objects, see ‘Table Parameters of dns cname record’	List	1 - 31	--

Table Parameters of dns cname record

Parameter name	Description	Data Type	Range	Default
name (*)	Name of DNS CNAME record.	String	1 – 127	
admin_preference	Admin preference of DNS CNAME record	Integer	0 - 255	100
as_backup	As backup of DNS CNAME record	Boolean	0 - 1	0
weight	Weight of DNS CNAME record	Integer	1 – 100	1

Note: (*) parameter is required in the API.

Response Fields

General AXAPI status response

Example

URL	https://[AX host name or IP address]:[service port]/services/rest/V2/?session_id=[your session id]&format=json&method=gslb.zone.service.dns cname_record.create
HTTP Action	POST
HTTP Post Body	{ "zone_name": "zone01", "service_name": "service01", "dns_cname_record_list": [{"name": "test1", "admin_preference": 50, "as_backup": 1, "weight": 60}, {"name": "test2", "admin_preference": 50, "as_backup": 1, "weight": 60}] }
Response	{"response": {"status": "OK"}}
CLI Config Before Call	gslb zone zone01 service ftp service01
CLI Config After Call	gslb zone zone01 service ftp service01 dns-cname-record test1 admin-preference 50 weight 60 as-backup dns-cname-record test2 admin-preference 50 weight 60 as-backup

Note: By default we use JSON format.

Menus Privilege

Config Mode >> Service >> GSLB >> Zone

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.6.22 “gslb.zone.service.dns cname_record.update” Method

Method name

gslb.zone.service.dns cname_record.update

Description

This method offers a quick way to update DNS CNAME records for a service in a GSLB zone. You can achieve the same goal by calling methods ‘gslb.zone.*’ or ‘gslb.service.*’.

Note: This method completely replaces the old list with the new list. If you only want to update specific items in the list, please delete the items using gslb.zone.service.dns cname_record.delete and then add the new items using gslb.zone.service.dns cname_record.add.

Parameters

Table Parameters of main

parameter name	Description	Data Type	Range	Default
session_id (*)	User authenticated session id	String		
method (*)	gslb.zone.service.dns_cname_record.update	String		
zone_name(*)	Name of GSLB zone	String	1 - 31	--
service_name(*)	Name of Service	String	1 - 31	
dns_cname_record_list (*)	List of DNS cname record objects, see 'Table Parameters of dns cname record'	List	1 - 31	--

Table Parameters of dns cname record

Parameter name	Description	Data Type	Range	Default
name (*)	Name of DNS CNAME record.	String	1 – 127	
admin_preference	Admin preferenceof DNS CNAME record	Integer	0 - 255	100
as_backup	As backup of DNS CNAME record	Boolean	0 - 1	0
weight	Weight of DNS CNAME record	Integer	1 – 100	1

Note: (*) parameter is required in the API.

Response Fields

General AXAPI status response

Example

URL	https://[AX host name or IP address]:[service port]/services/rest/V2/?session_id=[your session id]&format=json&method=gslb.zone.service.dns_cname_record.update
HTTP Action	POST
HTTP Post Body	{ "zone_name": "zone01", "service_name": "service01", "dns_cname_record_list": [{"name": "test1", "admin_preference": 66, "as_backup": 1, "weight": 55}, {"name": "test3", "admin_preference": 50, "as_backup": 1, "weight": 60}, {"name": "test4", "admin_preference": 50, "as_backup": 0, "weight": 60}] }
Response	{"response": {"status": "OK"}}
CLI Config Before Call	gslb zone zone01 service ftp service01 dns-cname-record test1 admin-preference 50 weight 60 as-backup dns-cname-record test2 admin-preference 50 weight 60 as-backup
CLI Config After Call	gslb zone zone01 service ftp service01 dns-cname-record test1 admin-preference 66 weight 55 as-backup dns-cname-record test3 admin-preference 50 weight 60 as-backup dns-cname-record test4 admin-preference 50 weight 60

Note: By default we use JSON format.

Menus Privilege

Config Mode >> Service >> GSLB >> Zone

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.6.23 “gslb.zone.service.dns cname record.delete” Method

Method name

gslb.zone.service.dns_cname_record.delete

Description

This method offers a quick way to delete DNS cname records from service of GSLB zone. You can achieve the same goal by calling methods ‘gslb.zone.*’ or ‘gslb.service.*’.

Parameters

Table Parameters of main

Parameter name	Description	Data Type	Range	Default
session_id (*)	User authenticated session id	String		
method (*)	gslb.zone.service.dns cname record.delete	String		
zone_name(*)	Name of GSLB zone	String	1 - 31	--
service_name(*)	Name of Service	String	1 - 31	
dns cname record_list (*)	List of DNS cname record objects, see ‘Table Parameters of dns cname record’	List	1 - 31	--

Table Parameters of dns cname record

Parameter name	Description	Data Type	Range	Default
name (*)	Name of DNS CNAME record.	String	1 – 127	
admin_preference	Admin preferenceof DNS CNAME record	Integer	0 - 255	100
as_backup	As backup of DNS CNAME record	Boolean	0 - 1	0
weight	Weight of DNS CNAME record	Integer	1 – 100	1

Note: (*) parameter is required in the API.

Response Fields

General AXAPI status response

Example

URL	https://[AX host name or IP address]:[service port]/services/rest/V2/?session_id=[your session id]&format=json&method=gslb.zone.service.dns cname record.delete
HTTP Action	POST
HTTP Post Body	{ "zone_name": "zone01", "service_name": "service01", "dns cname record_list": [{"name": "test1", "admin_preference": 66, "as_backup": 1, "weight": 55}] }
Response	{"response": {"status": "OK"}}
CLI Config Before Call	<pre> gslb zone zone01 service ftp service01 dns-cname-record test1 admin-preference 66 weight 55 as-backup dns-cname-record test3 admin-preference 50 weight 60 as-backup dns-cname-record test4 admin-preference 50 weight 60 </pre>
CLI Config After Call	<pre> gslb zone zone01 service ftp service01 dns-cname-record test3 admin-preference 50 weight 60 as-backup dns-cname-record test4 admin-preference 50 weight 60 </pre>

Note: By default we use JSON format.

Menus Privilege

Config Mode >> Service >> GSLB >> Zone

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.6.24 “gslb.zone.service.dns_cname_record.getAll” Method

Method name

gslb.zone.service.dns_cname_record.getAll

Description

This method offers a quick way to get all DNS CNAME records for a service of a GSLB zone. You can achieve the same goal by calling methods ‘gslb.zone.*’ or ‘gslb.service.*’.

Parameters

parameter name	Description	Data Type	Range	Default
session_id (*)	User authenticated session id	String		
method (*)	gslb.zone.service.dns_cname_record.get All	String		
zone_name(*)	Name of GSLB zone	String	1 - 31	--
service_name(*)	Name of Service	String	1 - 31	

Response Fields

Table Response fields of main

Field name	Description	Data Type	Range	Default
dns_cname_record_list (*)	List of DNS cname record objects, see ‘Response fields of dns cname record’	List	1 - 31	--

Table Response fields of dns cname record

Field name	Description	Data Type	Range	Default
name (*)	Name of DNS CNAME record.	String	1 – 127	
admin_preference	Admin preferenceof DNS CNAME record	Integer	0 - 255	100
as_backup	As backup of DNS CNAME record	Boolean	0 - 1	0
weight	Weight of DNS CNAME record	Integer	1 – 100	1

Example

URL	https://[AX host name or IP address]:[service port]/services/rest/V2/?session_id=[your session id]&format=json&method=gslb.zone.service.dns_cname_record.getAll
HTTP Action	POST
HTTP Post Body	{ "zone_name": "zone01", "service_name": "service01" }
Response	{ "dns_cname_record_list": [{ "name": "test1", "admin_preference": 66, "as_backup": 1, "weight": 55 }, { "name": "test3", "admin_preference": 50, "as_backup": 1, "weight": 60 }] }

	<pre> }, { "name": "test4", "admin_preference": 50, "as_backup": 0, "weight": 60 }] } </pre>
CLI Config Before Call	<pre> gslb zone zone01 service ftp service01 dns-cname-record test1 admin-preference 66 weight 55 as-backup dns-cname-record test3 admin-preference 50 weight 60 as-backup dns-cname-record test4 admin-preference 50 weight 60 </pre>
CLI Config After Call	No change

Note: By default we use JSON format.

Menus Privilege

Config Mode >> Service >> GSLB >> Zone

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.6.25 “gslb.zone.service.dns_cname_record.deleteAll” Method

Method name

gslb.zone.service.dns_cname_record.deleteAll

Description

This method offers a quick way to delete all DNS CNAME records for a service in a GSLB zone. You can achieve the same goal by calling methods ‘gslb.zone.*’ or ‘gslb.service.*’.

Parameters

parameter name	Description	Data Type	Range	Default
session_id (*)	User authenticated session id	String		
method (*)	gslb.zone.service.dns_cname_record.deleteAll	String		
zone_name(*)	Name of GSLB zone	String	1 - 31	--
service_name(*)	Name of Service	String	1 - 31	

Response Fields

General AXAPI status response

Example

URL	https://[AX host name or IP address]:[service port]/services/rest/V2/?session_id=[your session id]&format=json&method=gslb.zone.service.dns_cname_record.deleteAll
HTTP Action	POST
HTTP Post Body	<pre> { "zone_name": "zone01", "service_name": "service01" } </pre>
Response	{"response": {"status": "OK"}}
CLI Config Before Call	gslb zone zone01

	service ftp service01 dns-cname-record test1 admin-preference 66 weight 55 as-backup dns-cname-record test3 admin-preference 50 weight 60 as-backup dns-cname-record test4 admin-preference 50 weight 60
CLI Config After Call	gslb zone zone01 service ftp service01

Note: By default we use JSON format.

Menus Privilege

Config Mode >> Service >> GSLB >> Zone

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.6.26 “gslb.zone.service.dns_cname_record.search” Method

Method name

gslb.zone.service.dns_cname_record.search

Description

This method offers a quick way to get specific DNS CNAME records for a service in a GSLB zone. You can achieve the same goal by calling methods ‘gslb.zone.*’ or ‘gslb.service.*’.

Parameters

parameter name	Description	Data Type	Range	Default
session_id (*)	User authenticated session id	String		
method (*)	gslb.zone.service.dns_cname_record.search	String		
zone_name(*)	Name of GSLB zone	String	1 - 31	--
service_name(*)	Name of Service	String	1 - 31	
cname_name(*)	Name of the cname record	String	1 - 127	

Response Fields

General AXAPI status response

Example

URL	https://[AX host name or IP address]:[service port]/services/rest/V2/?session_id=[your session id]&format=json&method=gslb.zone.service.dns_cname_record.search
HTTP Action	POST
HTTP Post Body	{ "zone_name": "zone01", "service_name": "service01", "cname_name": "test1" }
Response	{ "dns_cname_record": { "name": "test1", "admin_preference": 66, "as_backup": 1, "weight": 55 } }

	}
CLI Config Before Call	gslb zone zone01 service ftp service01 dns-cname-record test1 admin-preference 66 weight 55 as-backup dns-cname-record test3 admin-preference 50 weight 60 as-backup dns-cname-record test4 admin-preference 50 weight 60
CLI Config After Call	No change

Note: By default we use JSON format.

Menus Privilege

Config Mode >> Service >> GSLB >> Zone

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.6.27 “gslb.zone.service.dns_ns_record.create” Method

Method name

gslb.zone.service.dns_ns_record.create

Description

This offers a quick way to add DNS NS records into service of GSLB zone. You can achieve the same goal by calling methods ‘gslb.zone.*’ or ‘gslb.service.*’.

Parameters

Table Parameters of main

parameter name	Description	Data Type	Range	Default
session_id (*)	User authenticated session id	String		
method (*)	gslb.zone.service.dns_ns_record.create	String		
zone_name(*)	Name of GSLB zone	String	1 - 31	--
service_name(*)	Name of Service	String	1 - 31	
dns_ns_record_list (*)	List of DNS NS record objects, see ‘Table Parameters of DNS NS record’	List	1 - 31	--

Table Parameters of dns ns record

Parameter name	Description	Data Type	Range	Default
name (*)	Name of DNS NS record.	String	1 – 127	

Note: (*) parameter is required in the API.

Response Fields

General AXAPI status response

Example

URL	https://[AX host name or IP address]:[service port]/services/rest/V2/?session_id=[your session id]&format=json&method=gslb.zone.service.dns_ns_record.create
HTTP Action	POST
HTTP Post Body	<pre>{ "zone_name": "zone01", "service_name": "service01", "dns_ns_record_list": [{"name": "QQQ"}, {"name": "OOO"}] }</pre>

Response	{"response": {"status": "OK"}}
CLI Config Before Call	gslb zone zone01 service ftp service01
CLI Config After Call	gslb zone zone01 service ftp service01 dns-ns-record qqq dns-ns-record ooo

Note: By default we use JSON format.

Menus Privilege

Config Mode >> Service >> GSLB >> Zone

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.6.28 “gslb.zone.service.dns_ns_record.update” Method

Method name

gslb.zone.service.dns_ns_record.update

Description

This method offers a quick way to update DNS NS records for a service in a GSLB zone. You can achieve the same goal by calling methods ‘gslb.zone.*’ or ‘gslb.service.*’.

Note: This method completely replaces the old list with the new list. If you only want to update specific items in the list, please delete the items using gslb.zone.service.dns_ns_record.delete and then add new items using gslb.zone.service.dns_ns_record.add.

Parameters

Table Parameters of main

parameter name	Description	Data Type	Range	Default
session_id (*)	User authenticated session id	String		
method (*)	gslb.zone.service.dns_ns_record.update	String		
zone_name(*)	Name of GSLB zone	String	1 - 31	--
service_name(*)	Name of Service	String	1 - 31	
dns_ns_record_list (*)	List of DNS ns record objects, see 'Table Parameters of DNS NS record'	List	1 - 31	--

Table Parameters of dns ns record

Parameter name	Description	Data Type	Range	Default
name (*)	Name of DNS NS record.	String	1 – 127	

Note: (*) parameter is required in the API.

Response Fields

General AXAPI status response

Example

URL	https://[AX host name or IP address]:[service port]/services/rest/V2/?session_id=[your session id]&format=json&method=gslb.zone.service.dns_ns_record.update
HTTP Action	POST
HTTP Post Body	{ "zone_name": "zone01",

	<pre>"service_name": "service01", "dns_ns_record_list": [{"name": "111"}, {"name": "222"}] }</pre>
Response	{"response": {"status": "OK"}}
CLI Config Before Call	gslb zone zone01 service ftp service01 dns-ns-record qqq
CLI Config After Call	gslb zone zone01 service ftp service01 dns-ns-record 111 dns-ns-record 222

Note: By default we use JSON format.

Menus Privilege

Config Mode >> Service >> GSLB >> Zone

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.6.29 “gslb.zone.service.dns_ns_record.delete” Method

Method name

gslb.zone.service.dns_ns_record.delete

Description

This method offers a quick way to delete DNS NS records for a service in a GSLB zone. You can achieve the same goal by calling methods ‘gslb.zone.*’ or ‘gslb.service.*’.

Parameters

Table Parameters of main

Parameter name	Description	Data Type	Range	Default
session_id (*)	User authenticated session id	String		
method (*)	gslb.zone.service.dns_ns_record.delete	String		
zone_name(*)	Name of GSLB zone	String	1 - 31	--
service_name(*)	Name of Service	String	1 - 31	--
dns_ns_record_list (*)	List of DNS NS record objects, see ‘Table Parameters of DNS NS record’	List	1 - 31	--

Table Parameters of dns ns record

Parameter name	Description	Data Type	Range	Default
name (*)	Name of DNS NS record.	String	1 – 127	

Note: (*) parameter is required in the API.

Response Fields

General AXAPI status response

Example

URL	https://[AX host name or IP address]:[service port]/services/rest/V2/?session_id=[your session id]&format=json&method=gslb.zone.service.dns_ns_record.delete
HTTP Action	POST
HTTP Post Body	{

	<pre>"zone_name": "zone01", "service_name": "service01", "dns_ns_record_list": [{ "name": "QQQ" }]</pre>
Response	{"response": {"status": "OK"}}
CLI Config Before Call	gslb zone zone01 service ftp service01 dns-ns-record QQQ dns-ns-record OOO
CLI Config After Call	gslb zone zone01 service ftp service01 dns-ns-record OOO

Note: By default we use JSON format.

Menus Privilege

Config Mode >> Service >> GSLB >> Zone

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.6.30 “gslb.zone.service.dns_ns_record.getAll” Method

Method name

gslb.zone.service.dns_ns_record.getAll

Description

This offers a quick way to get all DNS NS records for a service in a GSLB zone. You can achieve the same goal by calling methods ‘gslb.zone.*’ or ‘gslb.service.*’.

Parameters

parameter name	Description	Data Type	Range	Default
session_id (*)	User authenticated session id	String		
method (*)	gslb.zone.service.dns_ns_record.getAll	String		
zone_name(*)	Name of GSLB zone	String	1 - 31	--
service_name(*)	Name of Service	String	1 - 31	

Response Fields

Table Response fields of main

Field name	Description	Data Type	Range	Default
dns_ns_record_list (*)	List of DNS NS record objects, see ‘Response fields of DNS NS record’	List	1 - 31	--

Table Response fields of dns ns record

Parameter name	Description	Data Type	Range	Default
name (*)	Name of DNS NS record.	String	1 – 127	

Example

URL	https://[AX host name or IP address]:[service port]/services/rest/V2/?session_id=[your session id]&format=json&method=gslb.zone.service.dns_ns_record.getAll
HTTP Action	POST

HTTP Post Body	{ "zone_name": "zone01", "service_name": "service01" }
Response	{ "dns_ns_record_list": [{ "name": "qqq" }, { "name": "ooo" }] }
CLI Config Before Call	gslb zone zone01 service ftp service01 dns-ns-record QQQ dns-ns-record OOO
CLI Config After Call	No change

Note: By default we use JSON format.

Menus Privilege

Config Mode >> Service >> GSLB >> Zone

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.6.31 “gslb.zone.service.dns_ns_record.deleteAll” Method

Method name

gslb.zone.service.dns_ns_record.deleteAll

Description

This method offers a quick way to delete all DNS NS records for a service in a GSLB zone. You can achieve the same goal by calling methods ‘gslb.zone.*’ or ‘gslb.service.*’.

Parameters

parameter name	Description	Data Type	Range	Default
session_id (*)	User authenticated session id	String		
method (*)	gslb.zone.service.dns_ns_record.deleteAll	String		
zone_name(*)	Name of GSLB zone	String	1 - 31	--
service_name(*)	Name of Service	String	1 - 31	

Response Fields

General AXAPI status response

Example

URL	https://[AX host name or IP address]:[service port]/services/rest/V2/?session_id=[your session id]&format=json&method=gslb.zone.service.dns_ns_record.deleteAll
HTTP Action	POST
HTTP Post Body	{ "zone_name": "zone01", "service_name": "service01"

	}
Response	{"response": {"status": "OK"}}
CLI Config Before Call	gslb zone zone01 service ftp service01 dns-ns-record QQQ dns-ns-record OOO
CLI Config After Call	gslb zone zone01 service ftp service01

Note: By default we use JSON format.

Menus Privilege

Config Mode >> Service >> GSLB >> Zone

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.6.32 “gslb.zone.service.dns_srv_record.create” Method

Method name

gslb.zone.service.dns_srv_record.create

Description

This method offers a quick way to add DNS SRV records for a service into a GSLB zone. You can achieve the same goal by calling methods ‘gslb.zone.*’ or ‘gslb.service.*’.

Parameters

Table Parameters of main

parameter name	Description	Data Type	Range	Default
session_id (*)	User authenticated session id	String		
method (*)	gslb.zone.service.dns_srv_record.create	String		
zone_name(*)	Name of GSLB zone	String	1 - 31	--
service_name(*)	Name of Service	String	1 - 31	--
dns_srv_record_list (*)	List of DNS SRV record objects, see ‘Table Parameters of DNS SRV record’	List	1 - 31	--

Table Parameters of dns srv record

Parameter name	Description	Data Type	Range	Default
name (*)	Name of DNS SRV record.	String	1 – 127	--
priority(*)	Priority of DNS SRV record	Integer	0 - 65535	--
port	Port number of DNS SRV record, value scope is 1 – 15535, 1 – 65534 is normal value of port number, 65535 means ‘not set’	Integer	0 - 65535	65535 (not set)
weight	Weight of DNS SRV record	Integer	1 – 65535	10

Note: (*) parameter is required in the API.

Response Fields

General AXAPI status response

Example

URL	https://[AX host name or IP address]:[service port]/services/rest/V2/?session_id=[your]
-----	---

	session id]&format=json&method=gslb.zone.service.dns_srv_record.create
HTTP Action	POST
HTTP Post Body	{ "zone_name": "zone01", "service_name": "service01", "dns_srv_record_list": [{ "name": "AAAA", "port": 80, "priority": 10, "weight": 10 }, { "name": "BBBB", "port": 90, "port_defined": 1, "priority": 5, "weight": 5 }] }
Response	{"response": {"status": "OK"}}
CLI Config Before Call	gslb zone zone01 service ftp service01
CLI Config After Call	gslb zone zone01 service ftp service01 dns-srv-record bbbb 5 weight 5 port 90 dns-srv-record aaaa 10 port 80

Note: By default we use JSON format.

Menus Privilege

Config Mode >> Service >> GSLB >> Zone

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.6.33 “gslb.zone.service.dns_srv_record.update” Method

Method name

gslb.zone.service.dns_srv_record.update

Description

This method offers a quick way to update DNS SRV records for a service in a GSLB zone. You can achieve the same goal by calling methods ‘gslb.zone.*’ or ‘gslb.service.*’.

Note: This method completely replaces the old list with the new list. If you only want to update specific items in the list, please delete the items using gslb.zone.service.dns_srv_record.delete and then add the new items using gslb.zone.service.dns_srv_record.add.

Parameters

Table Parameters of main

parameter name	Description	Data Type	Range	Default
session_id (*)	User authenticated session id	String		
method (*)	gslb.zone.service.dns_srv_record.update	String		

zone_name(*)	Name of GSLB zone	String	1 - 31	--
service_name(*)	Name of Service	String	1 - 31	--
dns_srv_record_list (*)	List of DNS srv record objects, see 'Table Parameters of DNS SRV record'	List	1 - 31	--

Table Parameters of dns srv record

Parameter name	Description	Data Type	Range	Default
name (*)	Name of DNS SRV record.	String	1 – 127	--
priority(*)	Priority of DNS SRV record	Integer	0 - 65535	--
port	Port number of DNS SRV record, value scope is 1 – 15535, 1 – 65534 is normal value of port number, 65535 means 'not set'	Integer	0 - 65535	65535 (not set)
weight	Weight of DNS SRV record	Integer	1 – 65535	10

Note: (*) parameter is required in the API.

Response Fields

General AXAPI status response

Example

URL	https://[AX host name or IP address]:[service port]/services/rest/V2/?session_id=[your session id]&format=json&method=gslb.zone.service.dns_srv_record.update
HTTP Action	POST
HTTP Post Body	{ "zone_name": "zone01", "service_name": "service01", "dns_srv_record_list": [{ "name": "AAAA", "port": 80, "port_defined": 1, "priority": 10, "weight": 10 }, { "name": "BBBB", "port": 90, "port_defined": 1, "priority": 5, "weight": 5 }] }
Response	{"response": {"status": "OK"}}
CLI Config Before Call	gslb zone zone01 service ftp service01 dns-srv-record cccc 5 weight 1 port 900 dns-srv-record dddd 10 port 80
CLI Config After Call	gslb zone zone01 service ftp service01 dns-srv-record bbbb 5 weight 5 port 90 dns-srv-record aaaa 10 port 80

Note: By default we use JSON format.

Menus Privilege

Config Mode >> Service >> GSLB >> Zone

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.6.34 “gslb.zone.service.dns_srv_record.delete” Method

Method name

gslb.zone.service.dns_srv_record.delete

Description

This method offers a quick way to delete DNS SRV records for a service in a GSLB zone. You can achieve the same goal by calling methods ‘gslb.zone.*’ or ‘gslb.service.*’.

Parameters

Table Parameters of main

Parameter name	Description	Data Type	Range	Default
session_id (*)	User authenticated session id	String		
method (*)	gslb.zone.service.dns_srv_record.delete	String		
zone_name(*)	Name of GSLB zone	String	1 - 31	--
service_name(*)	Name of Service	String	1 - 31	
dns_srv_record_list (*)	List of DNS SRV record objects, see ‘Table Parameters of DNS SRV record’	List	1 - 31	--

Table Parameters of dns srv record

Parameter name	Description	Data Type	Range	Default
name (*)	Name of DNS SRV record.	String	1 – 127	--
priority(*)	Priority of DNS SRV record	Integer	0 - 65535	--
port	Port number of DNS SRV record, value scope is 1 – 15535, 1 – 65534 is normal value of port number, 65535 means ‘not set’	Integer	0 - 65535	65535 (not set)
weight	Weight of DNS SRV record	Integer	1 – 65535	10

Note: (*) parameter is required in the API.

Response Fields

General AXAPI status response

Example

URL	https://[AX host name or IP address]:[service port]/services/rest/V2/?session_id=[your session id]&format=json&method=gslb.zone.service.dns_srv_record.delete
HTTP Action	POST
HTTP Post Body	{ "zone_name": "zone01", "service_name": "service01", "dns_srv_record_list": [{ "name": "AAAA", "port": 80, "port_defined": 1, "priority": 10, "weight": 10 }] }
Response	{"response": {"status": "OK"}}
CLI Config Before Call	gslb zone zone01 service ftp service01 dns-srv-record bbbb 5 weight 5 port 90 dns-srv-record aaaa 10 port 80
CLI Config After Call	gslb zone zone01 service ftp service01 dns-srv-record bbbb 5 weight 5 port 90

Note: By default we use JSON format.

Menus Privilege

Config Mode >> Service >> GSLB >> Zone

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.6.35 “gslb.zone.service.dns_srv_record.getAll” Method

Method name

gslb.zone.service.dns_srv_record.getAll

Description

This method offers a quick way to get all DNS SRV records for a service in a GSLB zone. You can achieve the same goal by calling methods ‘gslb.zone.*’ or ‘gslb.service.*’.

Parameters

parameter name	Description	Data Type	Range	Default
session_id (*)	User authenticated session id	String		
method (*)	gslb.zone.service.dns_srv_record.getAll	String		
zone_name(*)	Name of GSLB zone	String	1 - 31	--
service_name(*)	Name of Service	String	1 - 31	

Response Fields

Table Response fields of main

Field name	Description	Data Type	Range	Default
dns_srv_record_list (*)	List of DNS SRV record objects, see ‘Response fields of dns srv record’	List	1 - 31	--

Table Response fields of dns srv record

Field name	Description	Data Type	Range	Default
name (*)	Name of DNS SRV record.	String	1 – 127	--
priority(*)	Priority of DNS SRV record	Integer	0 - 65535	--
port	Port number of DNS SRV record, value scope is 1 – 15535, 1 – 65534 is normal value of port number, 65535 means ‘not set’	Integer	0 - 65535	65535 (not set)
weight	Weight of DNS SRV record	Integer	1 – 65535	10

Example

URL	https://[AX host name or IP address]:[service port]/services/rest/V2/?session_id=[your session id]&format=json&method=gslb.zone.service.dns_srv_record.getAll
HTTP Action	POST
HTTP Post Body	{ "zone_name": "zone01", "service_name": "service01" }
Response	{ "dns_srv_record_list": [{ "name": "bbbb", "priority": 5, "port": 12345, "weight": 10 }] }

	<pre> "weight": 5, "port_defined": 1, "port": 90 }, { "name": "aaaa", "priority": 10, "weight": 10, "port_defined": 1, "port": 80 }] } </pre>
CLI Config Before Call	<pre> gslb zone zone01 service ftp service01 dns-srv-record bbbb 5 weight 5 port 90 dns-srv-record aaaa 10 port 80 </pre>
CLI Config After Call	No change

Note: By default we use JSON format.

Menus Privilege

Config Mode >> Service >> GSLB >> Zone

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.6.36 “gslb.zone.service.dns_srv_record.deleteAll” Method

Method name

gslb.zone.service.dns_srv_record.deleteAll

Description

This offers a quick way to delete all DNS SRV records for a service in a GSLB zone. You can achieve the same goal by calling methods ‘gslb.zone.*’ or ‘gslb.service.*’.

Parameters

parameter name	Description	Data Type	Range	Default
session_id (*)	User authenticated session id	String		
method (*)	gslb.zone.service.dns_srv_record.deleteAll	String		
zone_name(*)	Name of GSLB zone	String	1 - 31	--
service_name(*)	Name of Service	String	1 - 31	

Response Fields

General AXAPI status response

Example

URL	https://[AX host name or IP address]:[service port]/services/rest/V2/?session_id=[your session id]&format=json&method=gslb.zone.service.dns_srv_record.deleteAll
HTTP Action	POST
HTTP Post Body	{ "zone_name": "zone01", "service_name": "service01" }

Response	{"response": {"status": "OK"}}
CLI Config Before Call	gslb zone zone01 service ftp service01 dns-srv-record bbbb 5 weight 5 port 90 dns-srv-record aaaa 10 port 80
CLI Config After Call	gslb zone zone01 service ftp service01

Note: By default we use JSON format.

Menus Privilege

Config Mode >> Service >> GSLB >> Zone

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.6.37 “gslb.zone.service.dns_srv_record.search” Method

Method name

gslb.zone.service.dns_srv_record.search

Description

This method offers a quick way to get specific DNS SRV records for a service in a GSLB zone. You can achieve the same goal by calling methods ‘gslb.zone.*’ or ‘gslb.service.*’ .

Parameters

parameter name	Description	Data Type	Range	Default
session_id (*)	User authenticated session id	String		
method (*)	gslb.zone.service.dns_srv_record.search	String		
zone_name(*)	Name of GSLB zone	String	1 - 31	--
service_name(*)	Name of Service	String	1 - 31	
srv_name(*)	Name of the SRV record	String	1 - 127	

Response Fields

General AXAPI status response

Example

URL	https://[AX host name or IP address]:[service port]/services/rest/V2/?session_id=[your session id]&format=json&method=gslb.zone.service.dns_srv_record.search
HTTP Action	POST
HTTP Post Body	{ "zone_name": "zone01", "service_name": "service01", "srv_name": "aaaa" }
Response	{ "dns_srv_record": { "name": "aaaa", "priority": 10, "weight": 10, "port_defined": 1, "port": 80 } }

CLI Config Before Call	gslb zone zone01 service ftp service01 dns-srv-record bbbb 5 weight 5 port 90 dns-srv-record aaaa 10 port 80
CLI Config After Call	No change

Note: By default we use JSON format.

Menus Privilege

Config Mode >> Service >> GSLB >> Zone

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.6.38 “gslb.zone.service.dns_ptr_record.create” Method

Method name

gslb.zone.service.dns_ptr_record.create

Description

This method offers a quick way to add DNS PTR records for a service in a GSLB zone. You can achieve the same goal by calling methods ‘gslb.zone.*’ or ‘gslb.service.*’.

Parameters

Table Parameters of main

parameter name	Description	Data Type	Range	Default
session_id (*)	User authenticated session id	String		
method (*)	gslb.zone.service.dns_ptr_record.create	String		
zone_name(*)	Name of GSLB zone	String	1 - 31	--
service_name(*)	Name of Service	String	1 - 31	--
dns_ptr_record_list (*)	List of DNS PTR record objects, see ‘Table Parameters of DNS PTR record’	List	1 - 31	--

Table Parameters of dns ptr record

Parameter name	Description	Data Type	Range	Default
name (*)	Name of DNS PTR record.	String	1 – 127	

Note: (*) parameter is required in the API.

Response Fields

General AXAPI status response

Example

URL	https://[AX host name or IP address]:[service port]/services/rest/V2/?session_id=[your session id]&format=json&method=gslb.zone.service.dns_ptr_record.create
HTTP Action	POST
HTTP Post Body	{ "zone_name": "zone01", "service_name": "service01", "dns_ptr_record_list": [{ "name": "AAA" }, { "name": "BBB" }] }

	<pre> "name": "BBB" }] } </pre>
Response	{"response": {"status": "OK"}}
CLI Config Before Call	gslb zone zone01 service ftp service01
CLI Config After Call	gslb zone zone01 service ftp service01 dns-ptr-record aaa dns-ptr-record bbb

Note: By default we use JSON format.

Menus Privilege

Config Mode >> Service >> GSLB >> Zone

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.6.39 “gslb.zone.service.dns_ptr_record.update” Method

Method name

gslb.zone.service.dns_ptr_record.update

Description

This method offers a quick way to update DNS PTR records for a service in a GSLB zone. You can achieve the same goal by calling methods ‘gslb.zone.*’ or ‘gslb.service.*’.

Note: This method completely replaces the old list with the new list. If you only want to update specific items in the list, please delete the items using gslb.zone.service.dns_ptr_record.delete and then add the new items using gslb.zone.service.dns_ptr_record.add.

Parameters

Table Parameters of main

parameter name	Description	Data Type	Range	Default
session_id (*)	User authenticated session id	String		
method (*)	gslb.zone.service.dns_ptr_record.update	String		
zone_name(*)	Name of GSLB zone	String	1 - 31	--
service_name(*)	Name of Service	String	1 - 31	--
dns_ptr_record_list (*)	List of DNS PTR record objects, see ‘Table Parameters of DNS PTR record’	List	1 - 31	--

Table Parameters of dns ptr record

Parameter name	Description	Data Type	Range	Default
name (*)	Name of DNS PTR record.	String	1 – 127	

Note: (*) parameter is required in the API.

Response Fields

General AXAPI status response

Example

URL	https://[AX host name or IP address]:[service port]/services/rest/V2/?session_id=[your]
-----	---

	session id&format=json&method=gslb.zone.service.dns_ptr_record.update
HTTP Action	POST
HTTP Post Body	{ "zone_name": "zone01", "service_name": "service01", "dns_ptr_record_list": [{ "name": "AAA" }, { "name": "BBB" }] }
Response	{"response": {"status": "OK"}}
CLI Config Before Call	gslb zone zone01 service ftp service01 dns-ptr-record ccc dns-ptr-record ddd
CLI Config After Call	gslb zone zone01 service ftp service01 dns-ptr-record aaa dns-ptr-record bbb

Note: By default we use JSON format.

Menus Privilege

Config Mode >> Service >> GSLB >> Zone

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.6.40 “gslb.zone.service.dns_ptr_record.delete” Method

Method name

gslb.zone.service.dns_ptr_record.delete

Description

This method offers a quick way to delete DNS PTR records from a service in a GSLB zone. You can achieve the same goal by calling methods ‘gslb.zone.*’ or ‘gslb.service.*’.

Parameters

Table Parameters of main

Parameter name	Description	Data Type	Range	Default
session_id (*)	User authenticated session id	String		
method (*)	gslb.zone.service.dns_ptr_record.delete	String		
zone_name(*)	Name of GSLB zone	String	1 - 31	--
service_name(*)	Name of Service	String	1 - 31	
dns_ptr_record_list (*)	List of DNS PTR record objects, see ‘Table Parameters of DNS PTR record’	List	1 - 31	--

Table Parameters of dns ptr record

Parameter name	Description	Data Type	Range	Default
name (*)	Name of DNS PTR record.	String	1 – 127	

Note: (*) parameter is required in the API.

Response Fields

General AXAPI status response

Example

URL	<code>https://[AX host name or IP address]:[service port]/services/rest/V2/?session_id=[your session id]&format=json&method=gslb.zone.service.dns_ptr_record.delete</code>
HTTP Action	POST
HTTP Post Body	{ "zone_name": "zone01", "service_name": "service01", "dns_ptr_record_list": [{ "name": "AAA" }] }
Response	{"response": {"status": "OK"}}
CLI Config Before Call	gslb zone zone01 service ftp service01 dns-ptr-record aaa dns-ptr-record bbb
CLI Config After Call	gslb zone zone01 service ftp service01 dns-ptr-record bbb

Note: By default we use JSON format.

Menus Privilege

Config Mode >> Service >> GSLB >> Zone

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.6.41 “gslb.zone.service.dns_ptr_record.getAll” Method

Method name

`gslb.zone.service.dns_ptr_record.getAll`

Description

This method offers a quick way to get all DNS PTR records for a service in a GSLB zone. You can achieve the same goal by calling methods ‘gslb.zone.*’ or ‘gslb.service.*’.

Parameters

parameter name	Description	Data Type	Range	Default
<code>session_id (*)</code>	User authenticated session id	String		
<code>method (*)</code>	<code>gslb.zone.service.dns_ptr_record.getAll</code>	String		
<code>zone_name(*)</code>	Name of GSLB zone	String	1 - 31	--
<code>service_name(*)</code>	Name of Service	String	1 - 31	

Response Fields

Table Response fields of main

Field name	Description	Data Type	Range	Default
<code>dns_ptr_record_list (*)</code>	List of DNS ptr record objects, see	List	1 - 31	--

	'Response fields of DNS ptr record'			
--	-------------------------------------	--	--	--

Table Response fields of dns ptr record

Parameter name	Description	Data Type	Range	Default
name (*)	Name of DNS PTR record.	String	1 – 127	

Example

URL	https://[AX host name or IP address]:[service port]/services/rest/V2/?session_id=[your session id]&format=json&method=gslb.zone.service.dns_ptr_record.getAll
HTTP Action	POST
HTTP Post Body	{ "zone_name": "zone01", "service_name": "service01" }
Response	{ "dns_ptr_record_list": [{ "name": "aaa" }, { "name": "bbb" }] }
CLI Config Before Call	gslb zone zone01 service ftp service01 dns-ptr-record aaa dns-ptr-record bbb
CLI Config After Call	No change

Note: By default we use JSON format.

Menus Privilege

Config Mode >> Service >> GSLB >> Zone

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.6.42 “gslb.zone.service.dns_ptr_record.deleteAll” Method

Method name

gslb.zone.service.dns_ptr_record.deleteAll

Description

This method offers a quick way to delete all DNS PTR records from a service of a GSLB zone. You can achieve the same goal by calling methods ‘gslb.zone.*’ or ‘gslb.service.*’.

Parameters

parameter name	Description	Data Type	Range	Default
session_id (*)	User authenticated session id	String		
method (*)	gslb.zone.service.dns_ptr_record.delete All	String		
zone_name(*)	Name of GSLB zone	String	1 - 31	--
service_name(*)	Name of Service	String	1 - 31	

Response Fields

General AXAPI status response

Example

URL	https://[AX host name or IP address]:[service port]/services/rest/V2/?session_id=[your session id]&format=json&method=gslb.zone.service.dns_ptr_record.deleteAll
HTTP Action	POST
HTTP Post Body	{ "zone_name": "zone01", "service_name": "service01" }
Response	{"response": {"status": "OK"}}
CLI Config Before Call	gslb zone zone01 service ftp service01 dns-ptr-record aaa dns-ptr-record bbb
CLI Config After Call	gslb zone zone01 service ftp service01

Note: By default we use JSON format.

Menus Privilege

Config Mode >> Service >> GSLB >> Zone

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.6.43 “gslb.zone.service.dns_ptr_record.search” Method

Method name

gslb.zone.service.dns_ptr_record.search

Description

This method offers a quick way to get specific DNS PTR records for a service in a GSLB zone. You can achieve the same goal by calling methods ‘gslb.zone.*’ or ‘gslb.service.*’.

Parameters

parameter name	Description	Data Type	Range	Default
session_id (*)	User authenticated session id	String		
method (*)	gslb.zone.service.dns_ptr_record.search	String		
zone_name(*)	Name of GSLB zone	String	1 - 31	--
service_name(*)	Name of Service	String	1 - 31	
ptr_name(*)	Name of the PTRr record	String	1 - 127	

Response Fields

General AXAPI status response

Example

URL	https://[AX host name or IP address]:[service port]/services/rest/V2/?session_id=[your session id]&format=json&method=gslb.zone.service.dns_ptr_record.search
HTTP Action	POST
HTTP Post Body	{

	<pre>"zone_name": "zone01", "service_name": "service01", "ptr_name": "aaa" }</pre>
Response	<pre>{ "dns_ptr_record": { "name": "aaa" } }</pre>
CLI Config Before Call	<pre>gslb zone zone01 service ftp service01 dns-ptr-record aaa dns-ptr-record bbb</pre>
CLI Config After Call	No change

Note: By default we use JSON format.

Menus Privilege

Config Mode >> Service >> GSLB >> Zone

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.6.44 “gslb.zone.service.dns_address_record.create” Method

Method name

gslb.zone.service.dns_address_record.create

Description

This method offers a quick way to add DNS address records to the service of a GSLB zone. You can achieve the same goal by calling methods ‘gslb.zone.*’ or ‘gslb.service.*’.

Parameters

Table Parameters of main

parameter name	Description	Data Type	Range	Default
session_id (*)	User authenticated session id	String		
method (*)	gslb.zone.service.dns_address_record.create	String		
zone_name(*)	Name of GSLB zone	String	1 - 31	--
service_name(*)	Name of Service	String	1 - 31	
dns_address_record_list (*)	List of DNS address record objects, see ‘Table Parameters of DNS address record’	List	1 - 31	--

Table Parameters of dns address record

Parameter name	Description	Data Type	Range	Default
vip_order (*)	The virtual server host name or IP address. If you specify the host name here, the VIP should be pre-configured.	IP (V4&V6)		
as_replace	As replace status of DNS address record	Boolean	0 - 1	0
no_response	No response status of DNS address record	Boolean	0 - 1	0
static	Is static record?	Boolean	0 - 1	0
as_backup	As backup status of DNS address record	Boolean	0 - 1	0
ttl	TTL	Integer	0 - 2147483647	

weight	Weight of DNS address record	Integer	1 - 100	
---------------	------------------------------	---------	---------	--

Note: (*) parameter is required in the API.

Response Fields

General AXAPI status response

Example

URL	https://[AX host name or IP address]:[service port]/services/rest/V2/?session_id=[your session id]&format=json&method=gslb.zone.service.dns_address_record.create
HTTP Action	POST
HTTP Post Body	{ "zone_name": "zone01", "service_name": "service01", "dns_address_record_list": [{ "vip_order": "3ff3::2234", "as_replace": 1, "no_response": 0, "static": 1, "as_backup": 1, "ttl": 1223, "weight": 80 }] }
Response	{"response": {"status": "OK"}}
CLI Config Before Call	gslb zone zone01 service ftp service01
CLI Config After Call	gslb zone zone01 service ftp service01 dns-a-record ip 3ff3::2234 weight 80 ttl 1223 as-replace static as-backup

Note: By default we use JSON format.

Menus Privilege

Config Mode >> Service >> GSLB >> Zone

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.6.45 “gslb.zone.service.dns_address_record.update” **Method**

Method name

gslb.zone.service.dns_address_record.update

Description

This method offers a quick way to update DNS address records for a service in a GSLB zone. You can achieve the same goal by calling methods ‘gslb.zone.*’ or ‘gslb.service.*’.

Note: This method completely replaces the old list with the new list. If you only want to update specific items in the list, please delete the items using

`gslb.zone.service.dns_address_record.delete` and then add new items using `gslb.zone.service.dns_address_record.add`.

Parameters

Table Parameters of main

parameter name	Description	Data Type	Range	Default
<code>session_id (*)</code>	User authenticated session id	String		
<code>method (*)</code>	<code>gslb.zone.service.dns_address_record.update</code>	String		
<code>zone_name(*)</code>	Name of GSLB zone	String	1 - 31	--
<code>service_name(*)</code>	Name of Service	String	1 - 31	
<code>dns_address_record_list (*)</code>	List of DNS address record objects, see 'Table Parameters of dns address record'	List	1 - 31	--

Table Parameters of dns address record

Parameter name	Description	Data Type	Range	Default
<code>vip_order (*)</code>	The virtual server host name or IP address. If you specify the host name here, the VIP should be pre-configured.	IP (V4&V6)		
<code>as_replace</code>	As replace status of DNS address record	Boolean	0 - 1	0
<code>no_response</code>	No response status of DNS address record	Boolean	0 - 1	0
<code>static</code>	Is static record?	Boolean	0 - 1	0
<code>as_backup</code>	As backup status of DNS address record	Boolean	0 - 1	0
<code>ttl</code>	TTL	Integer	0 - 2147483647	
<code>weight</code>	Weight of DNS address record	Integer	1 - 100	

Note: (*) parameter is required in the API.

Response Fields

General AXAPI status response

Example

URL	<code>https://[AX host name or IP address]:[service port]/services/rest/V2/?session_id=[your session id]&format=json&method=gslb.zone.service.dns_address_record.update</code>
HTTP Action	POST
HTTP Post Body	<pre>{ "zone_name": "zone01", "service_name": "service01", "dns_address_record_list": [{ "vip_order": "3ff3::1234", "as_replace": 1, "no_response": 0, "static": 1, "as_backup": 1, "ttl": 1223, "weight": 80 }, { "vip_order": "3ff3::2234", "as_replace": 1, "no_response": 0, "static": 1, "as_backup": 1, "ttl": 1223, "weight": 80 }] }</pre>
Response	<code>{"response": {"status": "OK"}}</code>
CLI Config Before Call	<code>gslb zone zone01 service ftp service01</code>

	dns-a-record ip 3ff3::2234 weight 80 ttl 1223 as-replace static as-backup
CLI Config After Call	gslb zone zone01 service ftp service01 dns-a-record ip 3ff3::2234 weight 80 ttl 1223 as-replace static as-backup dns-a-record ip 3ff3::1234 weight 80 ttl 1223 as-replace static as-backup

Note: By default we use JSON format.

Menus Privilege

Config Mode >> Service >> GSLB >> Zone

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.6.46 “gslb.zone.service.dns_address_record.delete” Method

Method name

gslb.zone.service.dns_address_record.delete

Description

This method offers a quick way to delete DNS address records from the service of a GSLB zone. You can achieve the same goal by calling methods ‘gslb.zone.*’ or ‘gslb.service.*’.

Parameters

Table Parameters of main

Parameter name	Description	Data Type	Range	Default
session_id (*)	User authenticated session id	String		
method (*)	gslb.zone.service.dns_address_record.delete	String		
zone_name(*)	Name of GSLB zone	String	1 - 31	--
service_name(*)	Name of Service	String	1 - 31	--
dns_address_record_list (*)	List of DNS address record objects, see ‘Table Parameters of dns address record’	List	1 - 31	--

Table Parameters of dns address record

Parameter name	Description	Data Type	Range	Default
vip_order (*)	The virtual server host name or IP address. If you specify the host name here, the VIP should be pre-configured.	IP (V4&V6)		
as_replace	As replace status of DNS address record	Boolean	0 - 1	0
no_response	No response status of DNS address record	Boolean	0 - 1	0
static	Is static record?	Boolean	0 - 1	0
as_backup	As backup status of DNS address record	Boolean	0 - 1	0
ttl	TTL	Integer	0 - 2147483647	
weight	Weight of DNS address record	Integer	1 - 100	

Note: (*) parameter is required in the API.

Response Fields

General AXAPI status response

Example

URL	https://[AX host name or IP address]:[service port]/services/rest/V2/?session_id=[your session id]&format=json&method=gslb.zone.service.dns_address_record.delete
HTTP Action	POST
HTTP Post Body	{ "zone_name": "zone01", "service_name": "service01", "dns_address_record_list": [{ "vip_order": "3ff3::2234", "as_replace": 1, "no_response": 0, "static": 1, "as_backup": 1, "ttl": 1223, "weight": 80 }] }
Response	{"response": {"status": "OK"}}
CLI Config Before Call	gslb zone zone01 service ftp service01 dns-a-record ip 3ff3::2234 weight 80 ttl 1223 as-replace static as-backup dns-a-record ip 3ff3::1234 weight 80 ttl 1223 as-replace static as-backup
CLI Config After Call	gslb zone zone01 service ftp service01 dns-a-record ip 3ff3::1234 weight 80 ttl 1223 as-replace static as-backup

Note: By default we use JSON format.

Menus Privileqe

Config Mode >> Service >> GSLB >> Zone

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.6.47 “gslb.zone.service.dns_address_record.getAll” **Method**

Method name

gslb.zone.service.dns_address_record.getAll

Description

This method offers a quick way to get all DNS address records for a service in a GSLB zone. You can achieve the same goal by calling methods ‘gslb.zone.*’ or ‘gslb.service.*’.

Parameters

parameter name	Description	Data Type	Range	Default
session_id (*)	User authenticated session id	String		
method (*)	gslb.zone.service.dns_address_record.getAll	String		
zone_name(*)	Name of GSLB zone	String	1 - 31	--
service_name(*)	Name of Service	String	1 - 31	

Response Fields

Table Response fields of main

Field name	Description	Data Type	Range	Default
dns_address_record_list (*)	List of DNS address record objects, see 'Response fields of dns address record'	List	1 - 31	--

Table Response fields of dns address record

Field name	Description	Data Type	Range	Default
vip_order (*)	The virtual server host name or IP address. If you specify the host name here, the vip should be pre-configured.	IP (V4&V6)		
as_replace	As replace status of dns address record	Boolean	0 - 1	0
no_response	No response status of dns address record	Boolean	0 - 1	0
static	Is static record?	Boolean	0 - 1	0
as_backup	As backup status of dns address record	Boolean	0 - 1	0
ttl	TTL	Integer	0 - 2147483647	
weight	Weight of dns address record	Integer	1 - 100	

Example

URL	https://[AX host name or IP address]:[service port]/services/rest/V2/?session_id=[your session id]&format=json&method=gslb.zone.service.dns_address_record.getAll
HTTP Action	POST
HTTP Post Body	{ "zone_name": "zone01", "service_name": "service01" }
Response	{ "dns_address_record_list": [{ "vip_order": "3ff3::2234", "as_replace": 1, "no_response": 0, "static": 1, "as_backup": 1, "ttl": 1223, "weight": 80 }, { "vip_order": "3ff3::1234", "as_replace": 1, "no_response": 0, "static": 1, "as_backup": 1, "ttl": 1223, "weight": 80 }] }
CLI Config Before Call	gslb zone zone01 service ftp service01 dns-a-record ip 3ff3::2234 weight 80 ttl 1223 as-replace static as-backup dns-a-record ip 3ff3::1234 weight 80 ttl 1223 as-replace static as-backup
CLI Config After Call	No change

Note: By default we use JSON format.

Menus Privilege

Config Mode >> Service >> GSLB >> Zone

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.6.48 “gslb.zone.service.dns_address_record.deleteAll” Method

Method name

gslb.zone.service.dns_address_record.deleteAll

Description

This method offers a quick way to delete all DNS address records for a service in a GSLB zone. You can achieve the same goal by calling methods ‘gslb.zone.*’ or ‘gslb.service.*’.

Parameters

parameter name	Description	Data Type	Range	Default
session_id (*)	User authenticated session id	String		
method (*)	gslb.zone.service.dns_address_record.deleteAll	String		
zone_name(*)	Name of GSLB zone	String	1 - 31	--
service_name(*)	Name of Service	String	1 - 31	

Response Fields

General AXAPI status response

Example

URL	https://[AX host name or IP address]:[service port]/services/rest/V2/?session_id=[your session id]&format=json&method=gslb.zone.service.dns_address_record.deleteAll
HTTP Action	POST
HTTP Post Body	{ "zone_name": "zone01", "service_name": "service01" }
Response	{"response": {"status": "OK"}}
CLI Config Before Call	gslb zone zone01 service ftp service01 dns-a-record ip 3ff3::2234 weight 80 ttl 1223 as-replace static as-backup dns-a-record ip 3ff3::1234 weight 80 ttl 1223 as-replace static as-backup
CLI Config After Call	gslb zone zone01 service ftp service01

Note: By default we use JSON format.

Menus Privilege

Config Mode >> Service >> GSLB >> Zone

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.6.49 “gslb.zone.service.dns_address_record.search” Method

Method name

gslb.zone.service.dns_address_record.search

Description

This method offers a quick way to get specific DNS address records for a service in a GSLB zone. You can achieve the same goal by calling methods ‘gslb.zone.*’ or ‘gslb.service.*’.

Parameters

parameter name	Description	Data Type	Range	Default
session_id (*)	User authenticated session id	String		
method (*)	gslb.zone.service.dns_address_record.search	String		
zone_name(*)	Name of GSLB zone	String	1 - 31	--
service_name(*)	Name of Service	String	1 - 31	
vip_order(*)	Host name or IP address if virtual server	IP (v4 v6)		

Response Fields

General AXAPI status response

Example

URL	https://[AX host name or IP address]:[service port]/services/rest/V2/?session_id=[your session id]&format=json&method=gslb.zone.service.dns_address_record.search
HTTP Action	POST
HTTP Post Body	{ "zone_name": "zone01", "service_name": "service01", "vip_order": "3ff3::1234" }
Response	{ "dns_address_record": { "vip_order": "3ff3::1234", "as_replace": 1, "no_response": 0, "static": 1, "as_backup": 1, "ttl": 1223, "weight": 80 } }
CLI Config Before Call	gslb zone zone01 service ftp service01 dns-a-record ip 3ff3::2234 weight 80 ttl 1223 as-replace static as-backup dns-a-record ip 3ff3::1234 weight 80 ttl 1223 as-replace static as-backup
CLI Config After Call	No change

Note: By default we use JSON format.

Menus Privilege

Config Mode >> Service >> GSLB >> Zone

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.6.50 “gslb.zone.service.geo_location.create” Method

Method name

gslb.zone.service.geo_location.create

Description

This method offers a quick way to add DNS GEO location records into the service of a GSLB zone. You can achieve the same goal by calling methods ‘gslb.zone.*’ or ‘gslb.service.*’.

Parameters

Table Parameters of main

parameter name	Description	Data Type	Range	Default
session_id (*)	User authenticated session id	String		
method (*)	gslb.zone.service.geo_location.create	String		
zone_name(*)	Name of GSLB zone	String	1 - 31	--
service_name(*)	Name of Service	String	1 - 31	
geo_location_list (*)	List of DNS address record objects, see ‘Table Parameters of dns address record’	List	1 - 31	--

Table Parameters of dns address record

Parameter name	Description	Data Type	Range	Default
geo_location (*)	Geo location. The string include several sections which be separated by ‘.’. The length of each section must be no more than 15.	String (with special format)	1 - 127	
alias	It refers pre-defined DNS CNAME record.	String	0 - 127	
action	Action when geo location matched can be ‘allow’, ‘drop’, ‘forward query’, ‘forward response’, ‘forward both’ or ‘reject’. 0: none 1: forward response 2: forward both 3: forward query 4: drop 5: reject 6: allow This parameter and parameter ‘policy’ are mutual exclusion.	Enumeration	0 – 6	0
policy	Policy used for this geo location. The policy must be pre-defined. This parameter and parameter ‘action’ are mutual exclusion.	string	0 - 63	

Note: (*) parameter is required in the API.

Response Fields

General AXAPI status response

Example

URL	https://[AX host name or IP address]:[service port]/services/rest/V2/?session_id=[your session id]&format=json&method=gslb.zone.service.geo_location.create
HTTP Action	POST
HTTP Post Body	{ "zone_name": "zone01", "service_name": "service01", "geo_location_list": [{ "geo_location": "aaa", "alias": "test1", "action": 0, "policy": "" }] }

	<pre> }, { "geo_location": "bbb", "alias": "test1", "action": 0, "policy": "policy01" }] } </pre>
Response	{"response": {"status": "OK"}}
CLI Config Before Call	gslb zone zone01 service ftp service01 dns-cname-record test1 admin-preference 66 weight 55 as-backup dns-cname-record test3 admin-preference 50 weight 60 as-backup dns-cname-record test4 admin-preference 50 weight 60
CLI Config After Call	gslb zone zone01 service ftp service01 dns-cname-record test1 admin-preference 66 weight 55 as-backup dns-cname-record test3 admin-preference 50 weight 60 as-backup dns-cname-record test4 admin-preference 50 weight 60 geo-location bbb alias test1 geo-location aaa alias test1 geo-location bbb policy policy01

Note: By default we use JSON format.

Menus Privilege

Config Mode >> Service >> GSLB >> Zone

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.6.51 “gslb.zone.service.geo_location.update” Method

Method name

gslb.zone.service.geo_location.update

Description

This method offers a quick way to update the DNS address records for a service in a GSLB zone. You can achieve the same goal by calling methods ‘gslb.zone.*’ or ‘gslb.service.*’.

Note: This method only updates the items listed in the request and does not delete any old ones or any new ones. Therefore, you should verify that the items you want to update really do exist.

Parameters

Table Parameters of main

parameter name	Description	Data Type	Range	Default
session_id (*)	User authenticated session id	String		
method (*)	gslb.zone.service.geo_location.update	String		
zone_name(*)	Name of GSLB zone	String	1 - 31	--
service_name(*)	Name of Service	String	1 - 31	
geo_location_list (*)	List of DNS address record objects, see ‘Table Parameters of dns address record’	List	1 - 31	--

Table Parameters of dns address record

Parameter name	Description	Data Type	Range	Default
geo_location (*)	Geo location. The string include several	String (with	1 - 127	

	sections which be separated by '..'. The length of each section must be no more than 15.	special format)		
alias	It refers pre-defined DNS cname record.	String	0 - 127	
action	Action when geo location matched can be 'allow', 'drop', 'forward query', 'forward response', 'forward both' or 'reject'. 0: none 1: forward response 2: forward both 3: forward query 4: drop 5: reject 6: allow This parameter and parameter 'policy' are mutual exclusion.	Enumeration	0 – 6	0
policy	Policy used for this geo location. The policy must be pre-defined. This parameter and parameter 'action' are mutual exclusion.	string	0 - 63	

Note: (*) parameter is required in the API.

Response Fields

General AXAPI status response

Example

URL	https://[AX host name or IP address]:[service port]/services/rest/V2/?session_id=[your session id]&format=json&method=gslb.zone.service.geo_location.update
HTTP Action	POST
HTTP Post Body	{ "zone_name": "zone01", "service_name": "service01", "geo_location_list": [{ "geo_location": "aaa", "alias": "test3", "action": 0, "policy": "policy01" }, { "geo_location": "bbb", "alias": "test4", "action": 6, "policy": "" }] }
Response	{"response": {"status": "OK"}}
CLI Config Before Call	gslb zone zone01 service ftp service01 dns-cname-record test1 admin-preference 66 weight 55 as-backup dns-cname-record test3 admin-preference 50 weight 60 as-backup dns-cname-record test4 admin-preference 50 weight 60 geo-location bbb alias test1 geo-location aaa alias test1 geo-location bbb policy policy01
CLI Config After Call	gslb zone zone01 service ftp service01 dns-cname-record test1 admin-preference 66 weight 55 as-backup dns-cname-record test3 admin-preference 50 weight 60 as-backup dns-cname-record test4 admin-preference 50 weight 60 geo-location aaa alias test3 geo-location bbb alias test4 geo-location bbb action allow geo-location aaa policy policy01

Note: By default we use JSON format.

Menus Privilege

Config Mode >> Service >> GSLB >> Zone

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.6.52 “gslb.zone.service.geo_location.delete” Method

Method name

gslb.zone.service.geo_location.delete

Description

This method offers a quick way to delete DNS address records from a service in a GSLB zone. You can achieve the same goal by calling methods ‘gslb.zone.*’ or ‘gslb.service.*’.

Parameters

Table Parameters of main

Parameter name	Description	Data Type	Range	Default
session_id (*)	User authenticated session id	String		
method (*)	gslb.zone.service.geo_location.delete	String		
zone_name(*)	Name of GSLB zone	String	1 - 31	--
service_name(*)	Name of Service	String	1 - 31	
geo_location_list (*)	List of DNS address record objects, see ‘Table Parameters of dns address record’	List	1 - 31	--

Table Parameters of dns address record

Parameter name	Description	Data Type	Range	Default
geo_location (*)	Geo location. The string include several sections which be separated by ‘.’. The length of each section must be no more than 15.	String (with special format)	1 - 127	

Note: (*) parameter is required in the API.

Response Fields

General AXAPI status response

Example

URL	https://[AX host name or IP address]:[service port]/services/rest/V2/?session_id=[your session id]&format=json&method=gslb.zone.service.geo_location.delete
HTTP Action	POST
HTTP Post Body	{ "zone_name": "zone01", "service_name": "service01", "geo_location_list": [{ "geo_location": "aaa" }] }
Response	{"response": {"status": "OK"}}
CLI Config Before Call	gslb zone zone01

	<pre> service ftp service01 dns-cname-record test1 admin-preference 66 weight 55 as-backup dns-cname-record test3 admin-preference 50 weight 60 as-backup dns-cname-record test4 admin-preference 50 weight 60 geo-location aaa alias test3 geo-location bbb alias test4 geo-location bbb action allow geo-location aaa policy policy01 </pre>
CLI Config After Call	<pre> gslb zone zone01 service ftp service01 dns-cname-record test1 admin-preference 66 weight 55 as-backup dns-cname-record test3 admin-preference 50 weight 60 as-backup dns-cname-record test4 admin-preference 50 weight 60 geo-location bbb alias test4 geo-location bbb action allow </pre>

Note: By default we use JSON format.

Menus Privilege

Config Mode >> Service >> GSLB >> Zone

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.6.53 “gslb.zone.service.geo_location.getAll” Method

Method name

gslb.zone.service.geo_location.getAll

Description

This method offers a quick way to get all DNS address records for a service in a GSLB zone. You can achieve the same goal by calling methods ‘gslb.zone.*’ or ‘gslb.service.*’.

Parameters

parameter name	Description	Data Type	Range	Default
session_id (*)	User authenticated session id	String		
method (*)	gslb.zone.service.geo_location.getAll	String		
zone_name(*)	Name of GSLB zone	String	1 - 31	--
service_name(*)	Name of Service	String	1 - 31	

Response Fields

Table Response fields of main

Field name	Description	Data Type	Range	Default
geo_location_list (*)	List of DNS address record objects, see ‘Response fields of dns address record’	List	1 - 31	--

Table Response fields of dns address record

Parameter name	Description	Data Type	Range	Default
geo_location (*)	Geo location. The string include several sections which be separated by ‘.’. The length of each section must be no more than 15.	String (with special format)	1 - 127	
alias	It refers pre-defined DNS cname record.	String	0 - 127	
action	Action when geo location matched can be ‘allow’, ‘drop’, ‘forward query’, ‘forward response’, ‘forward both’ or ‘reject’. 0: none	Enumeration	0 – 6	0

	1: forward response 2: forward both 3: forward query 4: drop 5: reject 6: allow This parameter and parameter 'policy' are mutual exclusion.			
policy	Policy used for this geo location. The policy must be pre-defined. This parameter and parameter 'action' are mutual exclusion.	string	0 - 63	

Example

URL	https://[AX host name or IP address]:[service port]/services/rest/V2/?session_id=[your session id]&format=json&method=gslb.zone.service.geo_location.getAll
HTTP Action	POST
HTTP Post Body	{ "zone_name": "zone01", "service_name": "service01" }
Response	{ "geo_location_list": [{ "geo_location": "bbb", "alias": "test4", "action": 6, "policy": "" }] }
CLI Config Before Call	<pre> gslb zone zone01 service ftp service01 dns-cname-record test1 admin-preference 66 weight 55 as-backup dns-cname-record test3 admin-preference 50 weight 60 as-backup dns-cname-record test4 admin-preference 50 weight 60 geo-location bbb alias test4 geo-location bbb action allow </pre>
CLI Config After Call	No change

Note: By default we use JSON format.

Menus Privilege

Config Mode >> Service >> GSLB >> Zone

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.6.54 “gslb.zone.service.geo_location.deleteAll” Method

Method name

gslb.zone.service.geo_location.deleteAll

Description

This method offers a quick way to delete all DNS address records for a service in a GSLB zone. You can achieve the same goal by calling methods ‘gslb.zone.*’ or ‘gslb.service.*’.

Parameters

parameter name	Description	Data Type	Range	Default
session_id (*)	User authenticated session id	String		
method (*)	gslb.zone.service.geo_location.deleteAll	String		
zone_name(*)	Name of GSLB zone	String	1 - 31	--
service_name(*)	Name of Service	String	1 - 31	

Response Fields

General AXAPI status response

Example

URL	https://[AX host name or IP address]:[service port]/services/rest/V2/?session_id=[your session id]&format=json&method=gslb.zone.service.geo_location.deleteAll
HTTP Action	POST
HTTP Post Body	{ "zone_name": "zone01", "service_name": "service01" }
Response	{"response": {"status": "OK"}}
CLI Config Before Call	gslb zone zone01 service ftp service01 dns-cname-record test1 admin-preference 66 weight 55 as-backup dns-cname-record test3 admin-preference 50 weight 60 as-backup dns-cname-record test4 admin-preference 50 weight 60 geo-location bbb alias test4 geo-location bbb action allow
CLI Config After Call	gslb zone zone01 service ftp service01

Note: By default we use JSON format.

Menus Privilege

Config Mode >> Service >> GSLB >> Zone

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.6.55 “gslb.zone.service.geo_location.search” Method

Method name

gslb.zone.service.geo_location.search

Description

This method offers a quick way to get specific DNS address records for a service in a GSLB zone. You can achieve the same goal by calling methods ‘gslb.zone.*’ or ‘gslb.service.*’.

Parameters

parameter name	Description	Data Type	Range	Default
session_id (*)	User authenticated session id	String		
method (*)	gslb.zone.service.geo_location.search	String		
zone_name(*)	Name of GSLB zone	String	1 - 31	--
service_name(*)	Name of Service	String	1 - 31	
geo_location (*)	Geo location. The string include several	String		

	sections which be separated by '..'. The length of each section must be no more than 15.	(Special formatted)		
--	--	---------------------	--	--

Response Fields

General AXAPI status response

Example

URL	https://[AX host name or IP address]:[service port]/services/rest/V2/?session_id=[your session id]&format=json&method=gslb.zone.service.geo_location.search
HTTP Action	POST
HTTP Post Body	{ "zone_name": "zone01", "service_name": "service01", "geo_location": "bbb" }
Response	{ "geo_location": { "geo_location": "bbb", "alias": "test4", "action": 6, "policy": "" } }
CLI Config Before Call	gslb zone zone01 service ftp service01 dns-cname-record test1 admin-preference 66 weight 55 as-backup dns-cname-record test3 admin-preference 50 weight 60 as-backup dns-cname-record test4 admin-preference 50 weight 60 geo-location bbb alias test4 geo-location bbb action allow
CLI Config After Call	No change

Note: By default we use JSON format.

Menus Privilege

Config Mode >> Service >> GSLB >> Zone

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.7 GSLB Global

9.7.1 “gslb.global.get” Method

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	gslb.global.get	String		

Note: (*) parameter is required in the API.

Response Fields

```
{  
    gslb_global: {  
        run_gslb_as_site_slb_device  
        run_gslb_as_controller  
        gslb_protocol_update_interval  
        gslb_protocol_limits: {  
            active_rdt_query  
            active_rdt_response  
            active_rdt_session  
            connection_load_response  
            response  
            message  
        },  
        gslb_active_rdt: {  
            domain  
            interval  
            retry  
            sleep  
            timeout  
            track  
        }  
    }  
}
```

Example Response

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
GET

HTTP Body

session_id=xxxxxxxxxxxxxxxxxxxxxx&method=gslb.global.get

Response as the HTTP body:

```
<response status="ok">  
    <gslb_global>  
        <run_gslb_as_site_slb_device>0</run_gslb_as_site_slb_device>  
        <run_gslb_as_controller>0</run_gslb_as_controller>  
        <gslb_protocol_update_interval>0</gslb_protocol_update_interval>  
        <gslb_protocol_limits>  
            <active_rdt_query>100</active_rdt_query>  
            <active_rdt_response>100</active_rdt_response>  
            <active_rdt_session>100</active_rdt_session>  
            <connection_load_response>100</connection_load_response>  
            <response>100</response>  
            <message>100</message>  
        </gslb_protocol_limits>  
        <gslb_active_rdt>  
            <domain>xli.a10networks.com.cn</domain>  
            <interval>10</interval>  
            <retry>10</retry>  
            <sleep>10</sleep>  
            <timeout>10</timeout>  
            <track>10</track>  
        </gslb_active_rdt>  
    </gslb_global>  
</response>
```

```

</gslb_active_rdt>
</gslb_global>
</response>

```

Response as the HTTP body(json):

```
{
  "gslb_global": {
    "run_gslb_as_site_slb_device": 0,
    "run_gslb_as_controller": 0,
    "gslb_protocol_update_interval": 30,
    "gslb_protocol_limits": {
      "active_rdt_query": 100,
      "active_rdt_response": 100,
      "active_rdt_session": 100,
      "connection_load_response": 100,
      "response": 100,
      "message": 100
    },
    "gslb_active_rdt": {
      "domain": "xli.a10networks.com.cn",
      "interval": 10,
      "retry": 10,
      "sleep": 10,
      "timeout": 10,
      "track": 10
    }
  }
}
```

Menus Privilege

Config Mode >> Service >> GSLB >> Global

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

9.7.2 “gslb.global.set” Method

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	gslb.global.set	String		
gslb_global (has sub parameters)	Gslb global object	Structure		
run_gslb_as_site_slb_device	Run gslb as site slb device	Bool	0 – 1	0
run_gslb_as_controller	Run gslb as controller	Bool	0 – 1	0
gslb_protocol_update_interval	Gslb protocol update interval	Int	1 – 300	300
gslb_protocol_limits(has sub parameters)	Gslb protocol limits structure	Structure		
active_rdt_query	Query Messages of Active RTT	Int	1 – 1000000	200

active_rdt_response	Response Messages of Active RTT	Int	1 – 1000000	1000
active_rdt_session	Sessions of Active RTT	Int	1 - 1000000	32768
connection_load_response	Response Messages of Connection Load	Int	1 - 1000000	0(no limit)
response	Amount of Response Messages	Int	1 - 1000000	3600
message	Amout of Messages	Int	1 - 1000000	10000
gslb_active_rdt (has sub parameters)	Gslb active rdt	Structure		
domain	Query Domain	String	Len 1 - 31 1 - 16383	
interval	Query Interval	Int	0 - 16	1
retry	Retry Count	int	1 - 300	3
sleep	Sleep Time when query fail	int	1 - 16383	3
timeout	Query Timeout, unit: msec	int	3 - 16383	3000
track	Tracking Time, unit: second	int		60

Note: (*) parameter is required in the API.

Response Fields

Example Response

Request as:

- **URL:**
https://_AX_IP_Address_/services/rest/V2/session_id=xxxxxxxxxxxxxxxxxxxxx&method=gslb.global.set
- **HTTP Action:**
 POST

HTTP Body

```
gslb_global=run_gslb_as_site_slb_device%030%02run_gslb_as_controller%030%02gslb_protocol_update_interval%03300%02gslb_protocol_limits%02gslb_active_rdt&gslb_protocol_limits=active_rdt_query%03200%02active_rdt_response%031000%02active_rdt_session%03327%02connection_load_response%030%02response%033600%02message%0310000&gslb_active_rdt=domain%03%02interval%031%02retry%033%02sleep%033%02timeout%033000%02track%0360
```

HTTP Body (json)

```
{
  "gslb_global": {
    "run_gslb_as_site_slb_device": 0,
    "run_gslb_as_controller": 0,
    "gslb_protocol_update_interval": 300,
    "gslb_protocol_limits": {
      "active_rdt_query": 200,
      "active_rdt_response": 1000,
      "active_rdt_session": 32768,
      "connection_load_response": 0,
      "response": 3600,
      "message": 10000
    },
    "gslb_active_rdt": {
      "domain": "",
      "interval": 1
    }
}
```

```
        "retry": 3,  
        "sleep": 3,  
        "timeout": 3000,  
        "track": 60  
    }  
}  
}
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?><response status="ok"></response>
```

Response as the HTTP body(json):

```
{"response": {"status": "OK"}}
```

Menus Privilege

Config Mode >> Service >> GSLB >> Global

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

10 NAT Pool Management APIs (V2.0)

10.1 Pool

10.1.1 “nat.pool.getAll” Method

This method is used to get configuration information for all dynamic NAT pools configured on the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	nat.pool.getAll	String		

Note: (*) parameter is required in the API.

Response Fields

pool_list	XML tag for the collection of the pools
pool	XML tag for the pool
name	pool name
start_ip_addr	start IP address
end_ip_addr	end IP address
netmask	network mask for start and end IP address
gateway	default gateway to use to reach other subnets
ha_group_id	HA group ID to use for session backup, 0 means not set

Example Response

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
ipv6 nat pool nat2 3001::1 3001::3 netmask 64 gateway 3001::10 ha-group-id 1
!
ip nat pool nat1 192.168.9.1 192.168.9.5 netmask /24 gateway 3.3.3.3
!
```

1. HTTP Request in URL Format:

session_id=123456789&method=nat.pool.getAll&format=url

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in JSON Format:

```
session_id=123456789&method=nat.pool.getAll&format=json
```

Response as the HTTP body (JSON Format):

```
{
    "pool_list": [
        {
            "name": "nat1",
            "start_ip_addr": "192.168.9.1",
            "end_ip_addr": "192.168.9.5",
            "netmask": "255.255.255.0",
            "gateway": "3.3.3.3",
            "ha_group_id": 0
        },
        {
            "name": "nat2",
            "start_ip_addr": "3001::1",
            "end_ip_addr": "3001::3",
            "netmask": "64",
            "gateway": "3001::10",
            "ha_group_id": 0
        }
    ]
}
```

Menus Privilege

Config Mode >> Service >> IP Source NAT >> IPv4 Pool

Config Mode >> Service >> IP Source NAT >> IPv6 Pool

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

10.1.2 “nat.pool.search” Method

This method is used to get information about a specific IP NAT pool. You can request information based on the pool name or the beginning IP address in the pool.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	nat.pool.search	String		
name (*)	nat pool name	String		

Note: (*) parameter is required in the API.

Response Fields

pool	XML tag for the pool
name	pool name
start_ip_addr	start IP address
end_ip_addr	end IP address
netmask	network mask for start and end IP address
gateway	default gateway to use to reach other subnets
ha_group_id	HA group ID to use for session backup, 0 means not set

Example Response

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
ipv6 nat pool nat2 3001::1 3001::3 netmask 64 gateway 3001::10 ha-group-id 1
!
ip nat pool nat1 192.168.9.1 192.168.9.5 netmask /24 gateway 3.3.3.3
!
```

1. HTTP Request in URL Format:

```
session_id=123456789&method=nat.pool.search&name=nat1&format=url
```

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in JSON Format:

```
session_id=123456789&method=nat.pool.search&name=nat1&format=json
```

Response as the HTTP body (JSON Format):

```
{
  "pool": [
    {
      "name": "nat1",
      "start_ip_addr": "192.168.9.1",
      "end_ip_addr": "192.168.9.5",
      "netmask": "255.255.255.0",
      "gateway": "3.3.3.3",
      "ha_group_id": 0
    }
  ]
}
```

Menus Privilege

Config Mode >> Service >> IP Source NAT >> IPv4 Pool

Config Mode >> Service >> IP Source NAT >> IPv6 Pool

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

10.1.3 “nat.pool.fetchAllStatistics” Method

This method is used to get statistics for all IP NAT pools configured on the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	nat.pool.fetchAllStatistics	String		

Note: (*) parameter is required in the API.

Response Fields

pool_stat_list	XML tag for the collection of the NAT pool statistics
pool_stat	XML tag for the NAT pool statistics
name	NAT pool name
start_ip_addr	start IP address
end_ip_addr	end IP address
port_usage	total number of sessions currently being NATted for pool clients
total_used	total number of sessions that have been NATted for the source IP address
total_freed	total number of NATted sessions that have been terminated
total_failed	total number of failed allocations of an address from the pool to a client

Example Response

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
ipv6 nat pool nat2 3001::1 3001::3 netmask 64 gateway 3001::10 ha-group-id 1
!
ip nat pool nat1 192.168.9.1 192.168.9.5 netmask /24 gateway 3.3.3.3
!
```

1. HTTP Request in URL Format:

session_id=123456789&method=nat.pool.fetchAllStatistics&format=url

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in JSON Format:

session_id=123456789&method=nat.pool.fetchAllStatistics&format=json

Response as the HTTP body:

```
{
  "pool_list": [
    {
      "name": "nat1",
      "start_ip_addr": "192.168.9.1",
      "end_ip_addr": "192.168.9.5",
      "port_usage": 0,
      "total_used": 0,
      "total_freed": 0,
```

```

        "total_failed" : 0
    },
    {
        "name" : "nat2",
        "start_ip_addr" : "3001::1",
        "end_ip_addr" : "3001::3",
        "port_usage" : 0,
        "total_used" : 0,
        "total_freed" : 0,
        "total_failed" : 0
    },
]
}

```

Menus Privilege

Config Mode >> Service >> IP Source NAT >> IPv4 Pool

Config Mode >> Service >> IP Source NAT >> IPv6 Pool

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

10.1.4 “nat.pool.fetchStatistics” Method

This method is used to get statistics for a specific IP NAT pool configured on the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	nat.pool.fetchStatistics	String		
name (*)	nat pool name	String		

Note: (*) parameter is required in the API.

Response Fields

pool_stat	XML tag for the NAT pool statistics
name	NAT pool name
start_ip_addr	start IP address
end_ip_addr	end IP address
port_usage	total number of sessions currently being NATted for pool clients
total_used	total number of sessions that have been NATted for the source IP address
total_freed	total number of NATted sessions that have been terminated
total_failed	total number of failed allocations of an address from the pool to a client

Example Response

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
ipv6 nat pool nat2 3001::1 3001::3 netmask 64 gateway 3001::10 ha-group-id 1
!
ip nat pool nat1 192.168.9.1 192.168.9.5 netmask /24 gateway 3.3.3.3
!
```

1. HTTP Request in URL Format:

```
session_id=123456789&method=nat.pool.fetchStatistics&name=nat1&format=url
```

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in JSON Format:

```
session_id=123456789&method=nat.pool.fetchStatistics&name=nat1&format=json
```

Response as the HTTP body:

```
{
  "pool": {
    "name": "nat1",
    "start_ip_addr": "192.168.9.1",
    "end_ip_addr": "192.168.9.5",
    "port_usage": 0,
    "total_used": 0,
    "total_freed": 0,
    "total_failed": 0
  }
}
```

Menus Privilege

Config Mode >> Service >> IP Source NAT >> IPv4 Pool

Config Mode >> Service >> IP Source NAT >> IPv6 Pool

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

10.1.5 “nat.pool.create” Method

This method is used to create an IP NAT pool on the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	nat.pool.create	String		
name (*)	nat pool name	String		
start_ip_addr (*)	start ip address	String		
end_ip_addr (*)	end ip address	String		
netmask (*)	network mask	String		
gateway	default gateway to use to reach other subnet	String		
ha_group_id	ha group id, 0 means not set	Int	0 .. 31	

Note: (*) parameter is required in the API.

Note: The ‘start_ip_addr’, ‘end_ip_addr’ parameter can be an IPv6 address or IPv4 address. For the type different before, the netmask should be different.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
ipv6 nat pool nat2 3001::1 3001::3 netmask 64 gateway 3001::10
!
ip nat pool nat1 192.168.9.1 192.168.9.5 netmask /24 gateway 3.3.3.3
!
```

1. HTTP Request in URL Format:

session_id=xxxxxxxxxxxxxxxxxxxxx&method=nat.pool.create&pool=name%03nat3%02start_ip_addr%03192.168.10.1%02end_ip_addr%03192.168.10.6%02netmask%03255.255.255.0%02gateway%03192.168.3.1&format=url

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in JSON Format:

session_id=xxxxxxxxxxxxxxxxxxxxx&method=nat.pool.create&pool&format=json

Request POST data (JSON Format):

```
{
  "pool": {
    "name" : "nat3",
    "start_ip_addr" : "192.168.10.1",
    "end_ip_addr" : "192.168.10.6",
    "netmask" : "255.255.255.0",
    "gateway" : "192.168.3.1",
    "ha_group_id" : 0
  }
}
```

Response as the HTTP body (JSON Format):

```
{
  "response": {
    "status" : "OK"
  }
}
```

CLI configuration after aXAPI call:

```
!
ipv6 nat pool nat2 3001::1 3001::3 netmask 64 gateway 3001::10 ha-group-id 1
!
ip nat pool nat1 192.168.9.1 192.168.9.5 netmask /24 gateway 3.3.3.3
```

```
!
ip nat pool nat3 192.168.10.1 192.168.10.6 netmask /24 gateway 192.168.3.1
!
```

Menus Privilege

Config Mode >> Service >> IP Source NAT >> IPv4 Pool

Config Mode >> Service >> IP Source NAT >> IPv6 Pool

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

10.1.6 “nat.pool.update” Method

This method is used to update one or more paramenters in an IP NAT pool.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	nat.pool.update	String		
name (*)	nat pool name	String		
start_ip_addr	start ip address	String		
end_ip_addr	end ip address	String		
netmask	network mask	String		
gateway	default gateway to use to reach other subnet	String		
ha_group_id	ha group id, 0 means not set	Int	0 .. 31	

Note: (*) parameter is required in the API.

Note: The ‘start_ip_addr’, ‘end_ip_addr’ parameter can be an IPv6 address or IPv4 address. For the type different before, the netmask should be different.

Response Fields

General XML status response

Example Response

Example

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
ipv6 nat pool nat2 3001::1 3001::3 netmask 64 gateway 3001::10
!
ip nat pool nat1 192.168.9.1 192.168.9.5 netmask /24 gateway 3.3.3.3
!
```

1. HTTP Request in URL Format:

session_id=123456789&method=nat.pool.update&pool=name%03nat1%02ha_group_id%031&format=url

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
```

```
</response>
```

2. HTTP Request in JSON Format:

```
session_id=123456789&method=nat.pool.update&pool=name%03nat1%02ha_group_id%031&format=json
```

Request POST data (JSON Format):

```
{  
    "pool":  
    {  
        "name" : "nat1",  
        "start_ip_addr" : "192.168.9.1",  
        "end_ip_addr" : "192.168.9.5",  
        "netmask" : "255.255.255.0",  
        "gateway" : "3.3.3.3",  
        "ha_group_id" : 1  
    }  
}
```

Response as the HTTP body:

```
{  
    "response":  
    {  
        "status" : "OK"  
    }  
}
```

CLI configuration after aXAPI call:

```
! ipv6 nat pool nat2 3001::1 3001::3 netmask 64 gateway 3001::10 ha-group-id 1  
!  
ip nat pool nat1 192.168.9.1 192.168.9.5 netmask /24 gateway 3.3.3.3 ha-group-id 1  
!
```

Menus Privilege

Config Mode >> Service >> IP Source NAT >> IPv4 Pool

Config Mode >> Service >> IP Source NAT >> IPv6 Pool

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

10.1.7 “nat.pool.delete” Method

This method is used to delete an IP NAT pool from the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	nat.pool.delete	String		
name (*)	nat pool name	String		

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
ipv6 nat pool nat2 3001::1 3001::3 netmask 64 gateway 3001::10
!
ip nat pool nat1 192.168.9.1 192.168.9.5 netmask /24 gateway 3.3.3.3
!
```

1. HTTP Request in URL Format:

```
session_id=123456789&method=nat.pool.delete&name=nat1&format=url
```

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in URL Format:

```
session_id=123456789&method=nat.pool.delete&name=nat1&format=json
```

Request POST data (JSON Format):

```
{
  "name" : "nat1",
}
```

Response as the HTTP body (JSON Format):

```
{
  "response": {
    {
      "status" : "OK"
    }
  }
}
```

CLI configuration after aXAPI call:

```
!
ipv6 nat pool nat2 3001::1 3001::3 netmask 64 gateway 3001::10
!
```

Menus Privilege

Config Mode >> Service >> IP Source NAT >> IPv4 Pool

Config Mode >> Service >> IP Source NAT >> IPv6 Pool

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

10.2 Pool Group

10.2.1 “nat.pool_group.getAll” Method

This method is used to get configuration information for all dynamic NAT pool groups configured on the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	nat.pool_group.getAll	String		

Note: (*) parameter is required in the API.

Response Fields

pool_group_list	XML tag for the collection of the pool groups
pool_group	XML tag for the pool group
name	pool group name
ha_group_id	HA group ID, 0 means not set
member_list	XML tag for collection for NAT pools
member	a single NAT pool name

Example Response

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
ip nat pool-group pool_group_1 nat1 nat3
!
```

1. HTTP Request in URL Format:

session_id=123456789&method=nat.pool_group.getAll&format=url

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in JSON Format:

session_id=123456789&method=nat.pool_group.getAll&format=json

Response as the HTTP body (JSON Format):

```
{
  "pool_group_list":
```

```

[  

  {  

    "name" : "pool_group_1",  

    "ha_group_id" : 0,  

    "member_list" :  

      [  

        "nat1","nat3"  

      ]  

  }  

]  

}

```

Menus Privilege

Config Mode >> Service >> IP Source NAT >> Group

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

10.2.2 “nat.pool_group.search” Method

This method is used to get information about a specific pool group configured on the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	nat.pool_group.search	String		
name (*)	nat pool group name	String		

Note: (*) parameter is required in the API.

Response Fields

pool_group	XML tag for the pool group
name	pool group name
ha_group_id	HA group ID, 0 means not set
member_list	XML tag for collection for NAT pools
member	single NAT pool name

Example Response

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!  
ip nat pool-group pool_group_1 nat1 nat3  
!
```

1. HTTP Request in URL Format:

session_id=123456789&method=nat.pool_group.search&name=pool_group_1&format=url

2. HTTP Request in JSON Format:

```
session_id=123456789&method=nat.pool_group.search&name=pool_group_1&format=json
```

Response as the HTTP body (JSON Format):

```
{  
    "pool_group":  
    {  
        "name" : "pool_group_1",  
        "ha_group_id" : 0,  
        "member_list" :  
        [  
            "nat1","nat3"  
        ]  
    }  
}
```

Menus Privilege

Config Mode >> Service >> IP Source NAT >> Group

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

10.2.3 “nat.pool_group.create” Method

This method is used to create an IP NAT pool group on the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	nat.pool_group.create	String		
name (*)	nat pool group name	String		
ha_group_id	ha group id, 0 means not set	Int	0 .. 31	
member_list (*)	nat pool group member list to be added member1^Bmember2^B ... memberN ^B: ASCII Code 0x02, URL-encode %02	String		
member<n>	nat pool name	String		

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!  
ip nat pool-group pool_group_1 nat1 nat3  
!
```

1. HTTP Request in URL Format:

```
session_id=xxxxxxxxxxxxxxxxxxxxxx&method=nat.pool_group.create&pool_group=name%03pool_group_2%02member_list&member_list=member1%02member2&member1=nat1&member2=nat2&format=url
```

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in JSON Format:

```
session_id=xxxxxxxxxxxxxxxxxxxxxx&method=nat.pool_group.create&format=json
```

Request POST data (JSON Format):

```
{
  "pool_group": {
    "name" : "pool_group_2",
    "ha_group_id" : 0,
    "member_list" :
    [
      "nat2"
    ]
  }
}
```

Response as the HTTP body (JSON Format):

```
{
  "response":
  {
    "status" : "OK"
  }
}
```

CLI configuration after aXAPI call:

```
!
ip nat pool-group pool_group_1 nat1 nat3
!
ip nat pool-group pool_group_2 nat1 nat3
!
```

Menus Privilege

Config Mode >> Service >> IP Source NAT >> Group

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

10.2.4 “nat.pool_group.update” Method

This method is used to update one or more parameters for a NAT pool group configured on the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	nat.pool_group.update	String		
name (*)	nat pool group name	String		
ha_group_id	ha group id, 0 means not set	Int	0 .. 31	
member_list	nat pool group member list to be added member1^Bmember2^B ... memberN ^B: ASCII Code 0x02, URL-encode %02	String		
member<n>	nat pool name	String		

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Example

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
ip nat pool-group pool_group_1 nat1 nat3
!
```

1. HTTP Request in URL Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=nat.pool_group.update&pool_group=name%03pool_group_1%02member_list&member_list=member1%02member2&member1=nat1&member2=nat2&format=url
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in JSON Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=nat.pool_group.update&format=json
```

Request POST data (JSON Format):

```
{
  "pool_group": {
    "name" : "pool_group_1",
    "ha_group_id" : 0,
    "member_list" :
      [
        "nat1","nat2"
      ]
  }
}
```

Response as the HTTP body (JSON Format):

```
{
  "response":
```

```

    {
        "status" : "OK"
    }
}

```

CLI configuration after aXAPI call:

```

!
ip nat pool-group pool_group_1 nat1 nat2
!

```

Menus Privilege

Config Mode >> Service >> IP Source NAT >> Group

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

10.2.5 “nat.pool_group.delete” Method

This method is used to delete a specific IP NAT pool group from the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	nat.pool_group.delete	String		
name (*)	nat pool group name	String		

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```

!
ip nat pool-group pool_group_1 nat1 nat2
!

```

1. HTTP Request in URL Format:

session_id=123456789&method=nat.pool_group.delete&name=pool_group_1&format=url

Response as the HTTP body (URL Format):

```

<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>

```

2. HTTP Request in JSON Format:

```
session_id=123456789&method=nat.pool_group.delete&format=json
```

Request POST data (JSON Format):

```
{  
    "name" : "pool_group_1"  
}
```

Response as the HTTP body (JSON Format):

```
{  
    "response":  
    {  
        "status" : "OK"  
    }  
}
```

CLI configuration after aXAPI call:

```
!  
!  
!
```

Menus Privilege

Config Mode >> Service >> IP Source NAT >> Group

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

10.3 Static NAT

10.3.1 “nat.static_translation.getAll” Method

This method is used to get configuration information for all static IP NAT translations configured on the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	nat.static_translation.getAll	String		

Note: (*) parameter is required in the API.

Response Fields

static_nat_list	XML tag for the collection of the static NAT
static_nat	XML tag for the static NAT
source_ip_addr	inside address to be translated into a global address
global_ip_addr	global address to use for the inside address
ha_group_id	HA group ID to use for session backup, 0 means not set

Example Response

Request as:

- URL:

- https://_AX_IP_Address_/services/rest/V2/
- **HTTP Action:**
GET

CLI configuration before aXAPI call:

```
!
ip nat inside source static 192.168.12.1 192.168.12.4
!
ip nat inside source static 192.168.11.1 192.168.11.5
!
```

1. HTTP Request in URL Format:

session_id=123456789&method=nat.static_translation.getAll&format=url

2. HTTP Request in JSON Format:

session_id=123456789&method=nat.static_translation.getAll&format=json

Response as the HTTP body (JSON Format):

```
{
  "static_nat_list": [
    {
      "source_ip_addr": "192.168.12.1",
      "global_ip_addr": "192.168.12.4",
      "ha_group_id": 0
    },
    {
      "source_ip_addr": "192.168.11.1",
      "global_ip_addr": "192.168.11.5",
      "ha_group_id": 0
    }
  ]
}
```

Menus Privilege

Config Mode >> Service >> IP Source NAT >> Static NAT

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

10.3.2 “nat.static_translation.search” Method

This method is used to get information about a specific static IP NAT translation. You can request information based on source IP address or global IP address.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	nat.static_translation.search	String		
source_ip_addr (*)	source ip address	String		
global_ip_addr (*)	global ip address	String		

Note: (*) parameter is required in the API.

Response Fields

static_nat	XML tag for the static NAT
source_ip_addr	inside address to be translated into a global address
global_ip_addr	global address to use for the inside address
ha_group_id	HA group ID to use for session backup, 0 means not set

Example Response

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
ip nat inside source static 192.168.12.1 192.168.12.4
!
ip nat inside source static 192.168.11.1 192.168.11.5
!
```

1. HTTP Request in URL Format:

session_id=123456789&method=nat.static_translation.search&source_ip_addr=192.168.11.1&global_ip_addr=192.168.11.5&format=url

2. HTTP Request in JSON Format:

session_id=123456789&method=nat.static_translation.search&source_ip_addr=192.168.11.1&global_ip_addr=192.168.11.5&format=json

Response as the HTTP body (JSON Format):

```
{
  "static_nat": {
    "source_ip_addr": "192.168.11.1",
    "global_ip_addr": "192.168.11.5",
    "ha_group_id": 0
  }
}
```

Menus Privilege

Config Mode >> Service >> IP Source NAT >> Static NAT

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

10.3.3 “nat.static_translation.fetchAllStatistics” Method

This method is used to get statistics for all static IP NAT translations.

Parameters

Parameter Name	Description	Data Type	Range	Default
----------------	-------------	-----------	-------	---------

session_id (*)	user authenticated session id	String		
method (*)	nat.static_translation.fetchAllStatistics	String		

Note: (*) parameter is required in the API.

Response Fields

static_nat_list	XML tag for the collection of the static NAT
static_nat	XML tag for the static NAT
source_ip_addr	inside address to be translated into a global address
port_usage	total number of session currently being NATted for the source IP address
total_used	total number of sessions that have been NATted for the source IP address
total_freed	total number of NATted sessions that have been terminated

Example Response

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
ip nat inside source static 192.168.12.1 192.168.12.4
!
ip nat inside source static 192.168.11.1 192.168.11.5
!
```

1. HTTP Request in URL Format:

session_id=123456789&method=nat.static_translation.fetchAllStatistics&format=url

2. HTTP Request in JSON Format:

session_id=123456789&method=nat.static_translation.fetchAllStatistics&format=json

Response as the HTTP body:

```
{
  "static_nat_list": [
    [
      {
        "source_ip_addr": "192.168.12.1",
        "port_usage": 0,
        "total_used": 0,
        "total_freed": 0
      },
      {
        "source_ip_addr": "192.168.11.1",
        "port_usage": 0,
        "total_used": 0,
        "total_freed": 0
      }
    ]
  ]
}
```

Menus Privilege

Config Mode >> Service >> IP Source NAT >> Static NAT

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

10.3.4 “nat.static_translation.fetchStatistics” Method

This method is used to get statistics for a specific static IP NAT translation. You can request information based on source IP address or global IP address.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	nat.static_translation.fetchStatistics	String		
source_ip_addr (*)	source ip address	String		

Note: (*) parameter is required in the API.

Response Fields

static_nat	XML tag for the static NAT
source_ip_addr	inside address to be translated into a global address
port_usage	total number of session currently being NATted for the source IP address
total_used	total number of sessions that have been NATted for the source IP address
total_freed	total number of NATted sessions that have been terminated

Example Response

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
ip nat inside source static 192.168.12.1 192.168.12.4
!
ip nat inside source static 192.168.11.1 192.168.11.5
!
```

1. HTTP Request in URL Format:

```
session_id=123456789&method=nat.static_translation.fetchStatistics&source_ip_addr=192.168.12.1&format=url
```

2. HTTP Request in JSON Format:

```
session_id=123456789&method=nat.static_translation.fetchStatistics&source_ip_addr=192.168.12.1&format=json
```

Response as the HTTP body (JSON Format):

```
{
  "static_nat": {
    "
```

```

        "source_ip_addr" : "192.168.12.1",
        "port_usage" : 0,
        "total_used" : 0,
        "total_freed" : 0
    }
}

```

Menus Privilege

Config Mode >> Service >> IP Source NAT >> Static NAT

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

10.3.5 “nat.static_translation.create” Method

This method is used to create a static IP NAT translation on the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	nat.static_translation.create	String		
source_ip_addr (*)	source ip address	String		
global_ip_addr (*)	global ip address	String		
ha_group_id	ha group id, 0 means not set	Int	0 .. 31	

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
ha id 2
!
ha group 1 priority 100
!
ip nat inside source static 192.168.12.1 192.168.12.4
!
ip nat inside source static 192.168.11.1 192.168.11.5
!
```

1. HTTP Request in URL Format:

```
session_id=xxxxxxxxxxxxxxxxxxxxx&method=nat.static_translation.create&static_nat=source_ip
_addr%03192.168.13.1%02global_ip_addr%03192.168.13.6&format=url
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in JSON Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=nat.static_translation.create&format=json
```

CLI configuration after aXAPI call:

```
!
ip nat inside source static 192.168.13.1 192.168.13.6
!
ip nat inside source static 192.168.12.1 192.168.12.4 ha-group-id 1
!
ip nat inside source static 192.168.11.1 192.168.11.5
!
```

Menus Privilege

Config Mode >> Service >> IP Source NAT >> Static NAT

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

10.3.6 “slb.static_translation.update” Method

This method is used to update one or more parameters for a given static IP NAT translation. You can identify the translation by source IP address or global IP address.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	nat.static_translation.update	String		
source_ip_addr (*)	source ip address	String		
global_ip_addr (*)	global ip address	String		
ha_group_id	ha group id, 0 means not set	Int	0 .. 31	

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Example

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
ip nat inside source static 192.168.12.1 192.168.12.4
!
ip nat inside source static 192.168.11.1 192.168.11.5
!
```

1. HTTP Request in URL Format:

```
session_id=123456789&method=nat.static_translation.update&static_nat=source_ip_addr%03192
.168.11.1%02global_ip_addr%03192.168.11.5%02ha_group%031&format=url
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in JSON Format:

```
session_id=123456789&method=nat.static_translation.update&format=json
```

Response as the HTTP body:

```
{
  "response": {
    "status" : "OK"
  }
}
```

CLI configuration after aXAPI call:

```
!
ip nat inside source static 192.168.12.1 192.168.12.4
!
ip nat inside source static 192.168.11.1 192.168.11.5
!
```

Menus Privilege

Config Mode >> Service >> IP Source NAT >> Static NAT

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

10.3.7 “nat.static_translation.delete” Method

This method is used to delete a specific static IP NAT translation.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	nat.static_translation.delete	String		
source_ip_addr (*)	source ip address	String		
global_ip_addr (*)	global ip address	String		

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:

- https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
ip nat inside source static 192.168.12.1 192.168.12.4
!
ip nat inside source static 192.168.11.1 192.168.11.5
!
```

1. HTTP Request in URL Format:

```
session_id=123456789&method=nat.static_translation.delete&source_ip_addr=192.168.11.1&global_ip_addr=192.168.11.5&format=url
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in JSON Format:

```
session_id=123456789&method=nat.static_translation.delete&format=json
```

Response as the HTTP body (JSON Format):

```
{
  "response": [
    {
      "status": "OK"
    }
  ]
}
```

CLI configuration after aXAPI call:

```
!
ip nat inside source static 192.168.12.1 192.168.12.4
!
```

Menus Privilege

Config Mode >> Service >> IP Source NAT >> Static NAT

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

10.4 NAT Range

10.4.1 “nat.range.getAll” Method

This method is used to get configuration information for all static IP NAT range lists configured on the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	nat.range.getAll	String		

Note: (*) parameter is required in the API.

Response Fields

range_list	XML tag for the collection of the NAT ranges
range	XML tag for the NAT range
name	pool group name
local_ip_addr	local IP address
local_ip_mask	network mask of local IP address
global_ip_addr	global IP address
global_ip_mask	network mask of global IP address
count	number of address to be translated
ha_group_id	HA group ID, 0 means not set

Example Response

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
ip nat range-list nat_range_1 192.168.9.0 255.255.255.0 192.168.10.0 255.255.255.0 count 255
!
```

1. HTTP Request in URL Format:

```
session_id=123456789&method=nat.range.getAll&format=url
```

2. HTTP Request in JSON Format:

```
session_id=123456789&method=nat.range.getAll&format=json
```

Response as the HTTP body (JSON Format):

```
{
  "range_list": [
    {
      "name": "nat_range_1",
      "local_ip_addr": "192.168.9.0",
      "local_ip_mask": "255.255.255.0",
      "global_ip_addr": "192.168.10.0",
      "global_ip_mask": "255.255.255.0",
      "count": 255,
      "ha_group_id": 0
    }
  ]
}
```

Menus Privilege

Config Mode >> Service >> IP Source NAT >> NAT Range

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

10.4.2 “nat.range.search” Method

This method is used to get configuration information about a specific static IP NAT range list. You can request information based on the range list name, or by a local IP address or global IP address in the range list.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	nat.range.search	String		
name (*)	nat range name	String		

Note: (*) parameter is required in the API.

Response Fields

range	XML tag for the NAT range
name	pool group name
local_ip_addr	local IP address
local_ip_mask	network mask of local IP address
global_ip_addr	global IP address
global_ip_mask	network mask of global IP address
count	number of address to be translated
ha_group_id	HA group ID, 0 means not set

Example Response

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
ip nat range-list nat_range_1 192.168.9.0 255.255.255.0 192.168.10.0 255.255.255.0 count 255
!
```

1. HTTP Request in URL Format:

session_id=123456789&method=nat.range.search&name=nat_range_1&format=url

2. HTTP Request in JSON Format:

session_id=123456789&method=nat.range.search&name=nat_range_1&format=json

Response as the HTTP body (JSON Format):

```
{
  "range": {
    "name": "nat_range_1",
    "local_ip_addr": "192.168.9.0",
    "local_ip_mask": "255.255.255.0",
    "global_ip_addr": "192.168.10.0",
    "global_ip_mask": "255.255.255.0",
  }
}
```

```

        "count" : 255,
        "ha_group_id" : 0
    }
}

```

Menus Privilege

Config Mode >> Service >> IP Source NAT >> NAT Range

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

10.4.3 “nat.range.create” Method

This method is used to create a static IP NAT range list.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	nat.range.create	String		
name (*)	nat range name	String		
local_ip_addr (*)	local ip address	String		
local_ip_mask (*)	network mask of local ip address	String		
global_ip_addr (*)	global ip address	String		
global_ip_mask (*)	network mask of global ip address	String		
count (*)	number of address to be translated	Int	1 .. 200000	1
ha_group_id	ha group id, 0 means not set	Int	0 .. 31	

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
!
```

1. HTTP Request in URL Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=nat.range.create&range=name%03nat_range_1%02local_
ip_addr%03192.168.9.0%02local_ip_mask%03255.255.255.0%02global_ip_addr%03192.168.10.0%02g
lobal_ip_mask%03255.255.255.0%02count%03255&format=url
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in JSON Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=nat.range.create&format=json
```

Request POST data (JSON Format):

```
{  
    "range":  
    {  
        "name": "nat_range_1",  
        "local_ip_addr": "192.168.9.0",  
        "local_ip_mask": "255.255.255.0",  
        "global_ip_addr": "192.168.10.0",  
        "global_ip_mask": "255.255.255.0",  
        "count": 255,  
        "ha_group_id": 0  
    }  
}
```

Response as the HTTP body (JSON Format):

```
{  
    "response":  
    {  
        "status": "OK"  
    }  
}
```

CLI configuration after aXAPI call:

```
!  
ip nat range-list nat_range_1 192.168.9.0 255.255.255.0 192.168.10.0 255.255.255.0 count 255  
!
```

Menus Privilege

Config Mode >> Service >> IP Source NAT >> NAT Range

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

10.4.4 “nat.range.update” Method

This method is used to update a static IP NAT range list.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	nat.range.update	String		
name (*)	nat range name	String		
local_ip_addr	local ip address	String		
local_ip_mask	network mask of local ip address	String		
global_ip_addr	global ip address	String		
global_ip_mask	network mask of global ip address	String		
count	number of address to be translated	Int	1 .. 200000	1
ha_group_id	ha group id, 0 means not set	Int	0 .. 31	

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Example

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
ip nat range-list nat_range_1 192.168.9.0 255.255.255.0 192.168.10.0 255.255.255.0 count 255
!
```

1. HTTP Request in URL Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=nat.range.update&range=name%03nat_range_1%02local_ip_addr%03192.168.9.0%02local_ip_mask%03255.255.255.0%02global_ip_addr%03192.168.10.0%02global_ip_mask%03255.255.255.0%02count%03255%02ha_group_id%031&format=url
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in JSON Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=nat.range.update&format=json
```

Request POST data (JSON Format):

```
{
  "range": {
    "name": "nat_range_1",
    "local_ip_addr": "192.168.9.0",
    "local_ip_mask": "255.255.255.0",
    "global_ip_addr": "192.168.10.0",
    "global_ip_mask": "255.255.255.0",
    "count": 255,
    "ha_group_id": 1
  }
}
```

Response as the HTTP body (JSON Format):

```
{
  "response": {
    "status": "OK"
  }
}
```

CLI configuration after aXAPI call:

```
!
ip nat range-list nat_range_1 192.168.9.0 255.255.255.0 192.168.10.0 255.255.255.0 count 255
  ha-group-id 1
!
```

Menus Privilege

Config Mode >> Service >> IP Source NAT >> NAT Range

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

10.4.5 “nat.range.delete” Method

This method is used to delete a specific static IP NAT range list.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	nat.range.delete	String		
name (*)	nat range name	String		

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- **URL:**
https://_AX_IP_Address_/services/rest/V2/
- **HTTP Action:**
POST

CLI configuration before aXAPI call:

```
!
ip nat range-list nat_range_1 192.168.9.0 255.255.255.0 192.168.10.0 255.255.255.0 count 255
!
```

1. HTTP Request in URL Format:

session_id=123456789&method=nat.range.delete&name=nat_range_1&format=url

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in JSON Format:

session_id=123456789&method=nat.range.delete&name=nat_range_1&format=json

Request POST data (JSON Format):

```
{
    "name" : "nat_range_1",
}
```

Response as the HTTP body (JSON Format):

```
{
```

```

    "response":
    {
        "status" : "OK"
    }
}

```

CLI configuration after aXAPI call:

```
!
!
```

Menus Privilege

Config Mode >> Service >> IP Source NAT >> NAT Range

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

10.5 ACL Bind

10.5.1 “nat.acl_bind.getAll” Method

Description TBD.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	nat.acl_bind.getAll	String		

Note: (*) parameter is required in the API.

Response Fields

acl_bind_list	XML tag for the collection of the ACL binds
acl_bind	XML tag for the ACL bind
acl	IPv4 ACL ID
nat_pool	IP source NAT pool name

Example Response

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
! ip nat inside source list 1 pool pool_group_2
!
```

1. HTTP Request in URL Format:

session_id=123456789&method=nat.acl_bind.getAll&format=url

2. HTTP Request in JSON Format:

```
session_id=123456789&method=nat.acl_bind.getAll&format=json
```

Response as the HTTP body (JSON Format):

```
{
    "acl_bind_list": [
        {
            "acl": 1,
            "nat_pool": "pool_group_1"
        }
    ]
}
```

Menus Privilege

Config Mode >> Service >> IP Source NAT >> ACL Bind

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

10.5.2 “nat.acl_bind.create” Method

Description TBD.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	nat.acl_bind.create	String		
acl (*)	IPv4 acl id	Int	1 .. 199	
nat_pool (*)	ip source nat pool name	String		

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
access-list 3 permit any
!
ip nat pool nat3 192.168.10.1 192.168.10.6 netmask /24 gateway 192.168.3.1
!
```

1. HTTP Request in URL Format:

```
session_id=xxxxxxxxxxxxxxxxxxxxx&method=nat.acl_bind.create&acl_bind=acl%03%02nat_pool%03
nat3&format=url
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in JSON Format:

session_id=xxxxxxxxxxxxxxxxxxxx&method=nat.acl_bind.create&format=json

Request POST data (JSON Format):

```
{
  "acl_bind": {
    "acl": 1,
    "nat_pool": "nat3"
  }
}
```

Response as the HTTP body (JSON Format):

```
{
  "response": {
    "status": "OK"
  }
}
```

CLI configuration after aXAPI call:

```
!
ip nat inside source list 3 pool nat3
!
```

Menus Privilege

Config Mode >> Service >> IP Source NAT >> ACL Bind

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

10.5.3 “nat.acl_bind.delete” Method

Description TBD.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	nat.acl_bind.delete	String		
acl (*)	IPv4 acl id	Int	1 .. 199	
nat_pool (*)	ip source nat pool name	String		

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- **URL:**
https://_AX_IP_Address_/services/rest/V2/
- **HTTP Action:**
POST

CLI configuration before aXAPI call:

```
!
ip nat inside source list 3 pool nat3
!
```

1. HTTP Request in URL Format:

```
session_id=123456789&method=nat.acl_bind.delete&acl_bind=acl%033%02nat_pool%03nat3&format=url
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in JSON Format:

```
session_id=123456789&method=nat.acl_bind.delete&format=json
```

Request POST data (JSON Format):

```
{
  "acl_bind": {
    "acl": 1,
    "nat_pool": "nat3"
  }
}
```

Response as the HTTP body (JSON Format):

```
{
  "response": {
    "status": "OK"
  }
}
```

CLI configuration after aXAPI call:

```
!
```

Menus Privilege

Config Mode >> Service >> IP Source NAT >> ACL Bind

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

10.6 Interface

10.6.1 “nat.interface.getAll” Method

Description TBD.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	nat.interface.getAll	String		

Note: (*) parameter is required in the API.

Response Fields

interface_list	XML tag for the collection of the interfaces
interface	XML tag for the interface
name	interface name, include ethernet and virtual ethernet
direction	inside(0), outside(1) or both(2)

Example Response

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
interface ethernet 1
 ip address 192.168.78.5 255.255.255.0
 ip nat inside
!
```

1. HTTP Request in URL Format:

session_id=123456789&method=nat.interface.getAll&format=url

2. HTTP Request in JSON Format:

session_id=123456789&method=nat.interface.getAll&format=json

Response as the HTTP body (JSON Format):

```
{
  "interface_list": [
    {
      "name": "ethernet1",
      "direction": 0
    }
  ]
}
```

Menus Privilege

Config Mode >> Service >> IP Source NAT >> Interface

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

10.6.2 “nat.interface.create” Method

Description TBD.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	nat.interface.create	String		
name (*)	Interface or ve name	String		
direction	inside(0), outside(1) or both(2)	Int	0, 1 or 2	0

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
```

1. HTTP Request in URL Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=nat.interface.create&interface=name%03ethernet1%02
direction%030&format=url
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in JSON Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=nat.interface.create&format=json
```

Request POST data:

```
{
  "interface": {
    "name" : "ethernet1",
    "direction" : 0
  }
}
```

Response as the HTTP body:

```
{
  "response": {
    "status" : "OK"
  }
}
```

CLI configuration after aXAPI call:

```
!
interface ethernet 1
 ip address 192.168.78.5 255.255.255.0
 ip nat inside
!
```

Menus Privilege

Config Mode >> Service >> IP Source NAT >> Interface

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

10.6.3 “nat.interface.delete” Method

Description TBD.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	nat.interface.delete	String		
name (*)	Interface or ve name	String		
direction	inside(0), outside(1) or both(2)	Int	0, 1 or 2	0

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
interface ethernet 1
 ip address 192.168.78.5 255.255.255.0
 ip nat inside
!
```

1. HTTP Request in URL Format:

```
session_id=123456789&method=nat.interface.delete&interface=name%03ethernet1%02direction%0
30&format=url
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in JSON Format:

session_id=123456789&method=nat.interface.delete&format=json

Request POST data (JSON Format):

```
{
  "interface": {
    "name" : "ethernet1",
    "direction" : 0
  }
}
```

Response as the HTTP body (JSON Format):

```
{
  "response": {
    "status" : "OK"
  }
}
```

CLI configuration after aXAPI call:

!

Menus Privilege

Config Mode >> Service >> IP Source NAT >> Interface

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

10.7 Global

10.7.1 “nat.global.get” Method

Description TBD.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	nat.global.get	String		

Note: (*) parameter is required in the API.

Response Fields

global XML tag for the collection of the NAT global

pptp_nat_alg pptp NAT alg, enabled(1) or disabled(0)

nat_allow_static IP source NAT allow static host, enabled(1) or disabled(0)

nat_gw_l3	source NAT gateway for l3, enabled(1) or disabled(0)
syn_timeout	syn timeout
tcp_timeout	TCP timeout
udp_timeout	UDP timeout
icmp_timeout	ICMP timeout, 60 – 15000, 0 means fast type
service_timeout_list	XML tag for the collection of the services timeout
service_timeout	XML tag for the service timeout
protocol	TCP(0) or UDP(1)
port	
timeout	60 – 15000, 0 means fast type

Example Response

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
!
```

1. HTTP Request in URL Format:

```
session_id=123456789&method=nat.global.get&format=url
```

2. HTTP Request in JSON Format:

```
session_id=123456789&method=nat.global.get&format=json
```

Response as the HTTP body (JSON Format):

```
{
    "global": {
        "pptp_nat_alg" : 1,
        "nat_allow_static" : 0,
        "nat_gw_l3" : 0,
        "syn_timeout" : 60,
        "tcp_timeout" : 300,
        "udp_timeout" : 300,
        "icmp_timeout" : 0,
        "service_timeout_list" :
            [
                {
                    "protocol" : 1,
                    "port" : 53,
                    "timeout" : 0
                }
            ]
    }
}
```

Menus Privilege

Config Mode >> Service >> IP Source NAT >> Global

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

10.7.2 “nat.global.set” Method

Description TBD.

Parameters

Parameter Name	Description	Data Type	Range	Default
<code>session_id (*)</code>	user authenticated session id	String		
<code>method (*)</code>	<code>nat.global.set</code>	String		
<code>pptp_nat_alg</code>	pptv nat alg, enabled(1) or disabled(0)	Int	0 or 1	1
<code>nat_allow_static</code>	nat allow static host, enabled(1) or disabled(0)	Int	0 or 1	0
<code>nat_gw_l3</code>	nat gateway for l3, enabled(1) or disabled(0)	Int	0 or 1	0
<code>syn_timeout</code>	syn timeout	Int	60 .. 300	60
<code>tcp_timeout</code>	tcp timeout	Int	60 .. 15000	300
<code>udp_timeout</code>	udp timeout	Int	60 .. 15000	300
<code>icmp_timeout</code>	icmp timeout, 0 means fast type	Int	60 .. 15000 or 0	0
<code>service_timeout_list</code>	service timeout list to be added: <code>service_timeout1^Bservice_timeout2^B ...</code> ^B: ASCII Code 0x02, URL-encode %02	String	1 to 100	1
<code>service_timeout<n></code> <code>protocol:</code> <code>port: (*)</code> <code>timeout:</code>	service timeout at element <n> protocol type, either TCP(0) or UDP(1) port number 0 means fast type Note: In the n-th parameter, use ^B to separate different option, and use ^C to separate value for the option. ^B: ASCII Code 0x02, URL-encode %02 ^C: ASCII Code 0x03, URL-encode %03	Int Int Int	0 or 1 1 .. 65535 60 .. 15000 or 0	0 60

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!  
!
```

1. HTTP Request in URL Format:

```
session_id=xxxxxxxxxxxxxxxxxxxx&method=nat.global.set&global=pptp_nat_alg%031%02syn_timeo
ut%0360%02service_timeout_list&service_timeout_list=service_timeout1%02service_timeout2&s
ervice_timeout1=protocol%030%02port%038080&service_timeout2=protocol%031%02port%03808%02t
imeout%0360&format=url
```

Response as the HTTP body (URL Format):

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

2. HTTP Request in JSON Format:

```
session_id=xxxxxxxxxxxxxxxxxxxxxx&method=nat.global.set&format=json
```

Request POST data (JSON Format):

```
{
  "global": {
    "pptp_nat_alg" : 1,
    "nat_allow_static" : 0,
    "nat_gw_l3" : 1,
    "syn_timeout" : 120,
    "tcp_timeout" : 240,
    "udp_timeout" : 240,
    "icmp_timeout" : 60,
    "service_timeout_list" :
    [
      {
        "protocol" : 1,
        "port" : 8001,
        "timeout" : 60
      }
    ]
  }
}
```

Response as the HTTP body (JSON Format):

```
{
  "response": {
    "status" : "OK"
  }
}
```

CLI configuration after aXAPI call:

```
!
ip nat translation syn-timeout 120
ip nat translation tcp-timeout 240
ip nat translation udp-timeout 240
ip nat translation icmp-timeout 60
slb snat-gwy-for-l3
ip nat translation service-timeout udp 8001 60
!
```

Menus Privilege

Config Mode >> Service >> IP Source NAT >> Global

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

11 RBA/L3V Partition Management APIs

11.1 “system.partition.getAll” Method

This method is used to get configuration information for all partitions configured on the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	system.partition.getAll	String		

Note: (*) parameter is required in the API.

Response Fields

The response XML is the collection of partitions.

partition_list	XML tag for the collection of the partitions
partition	XML tag for the partition
name	partition name
max_aflex_file	aFlex script files allowed in the partition.
network_partition	network partition status, either enabled(1) or disabled(0) (network_partition is only available in 64-bit platform)

Example Response

Request as:

- **URL:**
https://_AX_IP_Address_/services/rest/V2/
- **HTTP Action:**
GET

CLI configuration before aXAPI call:

```
!
partition partition1 max-aflex-file 8
partition partition2 max-aflex-file 10 network-partition
!
```

HTTP Body

```
session_id=123456789&method=system.partition.getAll
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
  <partition_list>
    <partition>
      <name>partition1</name>
      <max_aflex_file>8</max_aflex_file>
      <network_partition>0</network_partition >
    </partition>
    <partition>
      <name>partition2</name>
      <max_aflex_file>10</max_aflex_file>
      <network_partition>1</network_partition >
    </partition>
  </partition_list>
</response>
```

Menus Privilege

Config Mode >> System >> Admin >> Partition

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

11.2 “system.partition.search” Method

This method is used to get configuration information for a specific partition, based on partition name.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	system.partition.search	String		
name (*)	partition name	String		

Note: (*) parameter is required in the API.

Response Fields

The response XML is the collection of partitions.

partition	XML tag for the partition
name	partition name.
max_aflex_file	aFlex script files allowed in the partition.
network_partition	network partition status, either enabled(1) or disabled(0)

Example Response

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
GET

CLI configuration before aXAPI call:

```
!
partition partition1 max-aflex-file 8
partition partition2 max-aflex-file 10 network-partition
!
```

HTTP Body

```
session_id=123456789&method=system.partition.search&name=partition1
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
  <partition>
    <name>partition1</name>
    <max_aflex_file>8</max_aflex_file>
    <network_partition>0</network_partition >
  </partition>
</response>
```

Menus Privilege

Config Mode >> System >> Admin >> Partition

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

11.3 “system.partition.create” Method

This method is used to create a new system partition on the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	system.partition.create	String		
name (*)	partition name			
max_aflex_file	partition aflex script files allow in the partition	Int	1 - 128	32
network_partition	network partition status, enabled(1) or disabled(0)	Int	1 or 0	0

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
!
```

Create a partition, partition1 using max aflex script allowed in this partition number as 32, network partition status as 1.

HTTP Body

```
session_id=123456789&method=system.partition.create&name=partition1&max_aflex_file=32&net
work_partition=1
```

CLI configuration after aXAPI call:

```
!
partition partition1 network-partition
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> System >> Admin >> Partition

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

11.4 “system.partition.update” Method

This method is used to update one or more parameters within a given system partition.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	system.partition.update	String		
name (*)	partition name			
max_aflex_file	partition aflex script files allow in the partition	Int	1 - 128	32

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Example 1.

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
POST

Update an existing partition, partition1 using max aflex script allowed in this partition number as 32, network partition status as 1.

CLI configuration before aXAPI call:

```
!
partition partition1 max-aflex-file 8
!
```

HTTP Body

```
session_id=123456789&method=system.partition.update&name=partition1&max_aflex_file=32
```

CLI configuration after aXAPI call:

```
!
partition partition1
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> System >> Admin >> Partition

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

11.5 “system.partition.active” Method

This method is used to create an active partition. Consecutive API calls will be associated with the current partition for the partition-aware APIs.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	system.partition.active	String		
name (*)	partition name			

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Example 1.

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
POST
- CLI configuration before aXAPI call:

```
!
partition partition1
!
```
- HTTP Body
session_id=123456789&method=system.partition.active&name=partition1

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

N/A

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

11.6 “system.partition.delete” Method

This method is used to delete a partition from the AX device.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id	String		
method (*)	system.partition.delete	String		
name (*)	partition name	String		

Note: (*) parameter is required in the API.

Response Fields

General XML status response

Example Response

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
POST

CLI configuration before aXAPI call:

```
!
```

```
partition partition1 max-aflex-file 8
partition partition2 max-aflex-file 10 network-partition
!
```

HTTP Body

```
session_id=123456789&method=system.partition.delete&name=partition1
```

CLI configuration after axAPI call:

```
!
partition partition2 max-aflex-file 10 network-partition
!
```

Response as the HTTP body:

```
<?xml version="1.0" encoding="utf-8" ?>
<response status="ok">
</response>
```

Menus Privilege

Config Mode >> System >> Admin >> Partition

Please see “Admin Roles” on page 29 for the privileges associated with this method.

Error Codes

TBD

12 CLI Deploying APIs

“CLI Deploying APIs” allow you to process CLI commands through the aXAPI interface, for both ‘read’ and ‘set’ operations. These APIs are supplemental to the standard set of aXAPI methods, and they can be used if none of the standard methods are able to meet your requirements.

To process these APIs, place your CLI commands in the HTTP post buffer, and then place the method name, session ID, and other parameters in the URL. Then, send them to the AX device’s HTTPS management interface.

The process should generate the CLI output. However, if an error occurs, an XML-based document is produced, which contains the error information.

Note: If you see ‘OK’ this means the aXAPI logic was successfully executed. We do not validate the command. If you input a command that does not exist, the aXAPI will return the CLI output to you without any packaging.

“CLI Deploying APIs” allow you to process multiple CLI commands in one call. However, since aXAPI is HTTP-based web service, the I/O length is limited, as indicated in the table below:

Buffer	Limit
Post buffer (Input of commands)	1M
HTTP payload buffer (output of single command)	1M
HTTP payload buffer (output of multiple command)	4M

Entering input that is too long will cause the call to fail. Similarly, if the output is too long, then the output will be cut off.

The commands listed below cannot be processed:

```
exit
ping
ssh
telnet
traceroute
axdebug
debug
reboot
reload
restart
repeat
shutdown
terminal
do ...
show tech
?
```

(Including all sub-commands following the “do” command prefix)

(This symbol is not allowed in any command)

12.1 “cli.show_info” Method

This method processes commands under Privileged EXEC mode, so you cannot execute any ‘set’ operations.

Parameters

Parameter Name	Description	Data Type	Range	Default
username	The username to login to the AX CLI. If you omit this parameter, the username of this aXAPI session will be used.	String	1 - 64	Username of this aXAPI session.
password	The password to login to AX CLI. If you omit this parameter, the password of this aXAPI session will be used.	String	1 - 64	Password of this aXAPI session.
enable_password	The password to enter enable mode (privilege EXEC mode). If enable-password is set, you must fill this parameter. Otherwise, leave it empty.	String	1 - 64	Empty string

Example Response

Request as:

- **URL:**
https://192.168.100.44:443/services/rest/V2/?session_id=ff52ffff07ffff58ffff46ff4ffff&method=cli.show_info
- **HTTP Action:**
POST
- **Response Fields:**
Standard plain text CLI output

Example:

Process ‘show version’ and ‘help’ in aXAPI:

HTTP POST Body:

```
show ver  
help
```

OUTPUT:

```
show ver  
AX Series Advanced Traffic Manager AX2100  
Copyright 2007-2011 by A10 Networks, Inc. All A10 Networks products are  
protected by one or more of the following US patents and patents pending:  
7716378, 7675854, 7647635, 7552126, 20090049537, 20080229418, 20080040789,  
20070283429, 20070271598, 20070180101  
  
Advanced Core OS (ACOS) version 2.6.1-P3, build 49 (Nov-10-2011,15:13)  
Booted from Hard Disk primary image  
Serial Number: AX21021107420008  
aFleX version: 2.0.0  
aXAPI version: 2.0  
Hard Disk primary image (default) version 2.6.1-P3, build 49  
Hard Disk secondary image version 2.4.4-P1, build 40  
Compact Flash primary image version 2.6.0-p3, build 1  
Compact Flash secondary image (default) version 2.4.2-p2, build 5  
Last configuration saved at Nov-11-2011, 01:31  
Hardware: 4 CPUs(Stepping 6), Single 149G Hard disk  
Memory 2073 Mbyte, Free Memory 777 Mbyte  
Current time is Nov-11-2011, 03:02  
The system has been up 0 day, 1 hour, 30 minutes  
AX2100#help
```

CLI Quick Reference
=====

1. Online Help

Enter "?" at a command prompt to list the commands available at that CLI level.
Enter "?" at any point within a command to list the available options.

Two types of help are provided:

- 1) When you are ready to enter a command option, type "?" to display each possible option and its description. For example: show ?
- 2) If you enter part of an option followed by "?", each command or option that matches the input is listed. For example: show us?

2. Word Completion

The CLI supports command completion, so you do not need to enter the entire name of a command or option. As long as you enter enough characters of the command or option name to avoid ambiguity with other commands or options, the CLI can complete the command or option.

After entering enough characters to avoid ambiguity, press "tab" to auto-complete the command or option.

AX2100#

The output is the same as what would appear in the AX CLI.

Menus Privilege

No

Error Codes

- 2003: Access denied. Invalid enable password.
- 2004: Cannot enter config mode since another user is in it.
- 2005: Internal I/O error.
- 2006: This command is not allowed to be processed. line %d.
- 2007: Invalid username/password.
- 2008: Permission denied: Insufficient privileges.

12.2 “cli.deploy” Method

This method processes CLI commands while in config mode, so you can process both ‘read’ and ‘set’ operations.

Parameters

Parameter Name	Description	Data Type	Range	Default
Username	The username to login to AX CLI. If you omit this parameter, the username of this aXAPI session will be used.	String	1 - 64	Username of this aXAPI session.
Password	The password to login to AX CLI. If you omit this parameter, the password of this aXAPI session will be used.	String	1 - 64	Password of this aXAPI session.
enable_password	The password to enter enable mode (privilege EXEC mode). If enable-password is set, you must fill this parameter.	String	1 - 64	Empty string

	Otherwise, leave it empty.			
grab_config	The AX CLI allows only one user in config mode at a time. Thus, if you are the root user, you can kick others out by setting this value equal to 1. 1: kick out other user 0: do not kick out others, you will receive a error message indicating that the other user is in config mode.	Bool	0 – 1	0

Example Response

Request as:

- **URL:**
`https://192.168.100.44:443/services/rest/V2/?session_id=ff52ffff07ffff58ffff46ff4fffff&method=cli.deploy&username=admin&password=a10&enable_password=123&grab_config=1`
 - **HTTP Action:**
POST
 - **Response Fields:**
Standard plain text CLI output

Example:

In this example, we will configure a real server.

HTTP POST Body:

```
slb server apiTest01 122.21.251.144
    template server servert
    health-check ping
    port 90  udp
    port 80  udp
        template port portt
    port 20  tcp
        conn-limit 8000000 no-logging
```

OUTPUT:

```
slb server apiTest01 122.21.251.144
AX2100(config-real server)#      template server servert
AX2100(config-real server)#      health-check ping
AX2100(config-real server)#      port 90    udp
AX2100(config-real server-node port)#   port 80    udp
AX2100(config-real server-node port)#   template port portt
AX2100(config-real server-node port)#   port 20    tcp
AX2100(config-real server-node port)#   conn-limit 8000000 no-logging
AX2100(config-real server-node port)#

```

The output is the same as what would appear in the AX CLI.

Menus Privilege

No

Error Codes

- 2003: Access denied. Invalid enable password.
 - 2004: Cannot enter config mode since another user is in it.
 - 2005: Internal I/O error.
 - 2006: This command is not allowed to process. line %d.
 - 2007: Invalid username/password.
 - 2008: Permission denied: Insufficient privileges.

13 AXAPI Method Management APIs

13.1 “axapi.method.getAll” Method

This method is used to get all aXAPI methods for one version.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	axapi.method.getAll	String		
Version	Version of axapi methods	String	V1.0/1.1/1.2/1.3/2.0	V2

Note: (*) parameter is required in the API.

Response Fields

The response XML is a collection of aXAPI methods. Each aXAPI method will have a collection of the method members.

axapi_method_list	XML tag for the collection of the aXAPI methods
axapi_method	XML tag for the aXAPI method
name	the name of the aXAPI method
privilege	the privilege of the aXAPI method.
json_enabled	shows whether the aXAPI method is enabled for json, 0(false), 1(true)

Example Response

Request as:

- URL:
https://_AX_IP_Address_/services/rest/V2/
- HTTP Action:
GET

1 HTTP Request in URL Format:

session_id=123456789&method=axapi.method.getAll&format=url&version=v1

Response as the HTTP body (URL Format):

```
<response status="ok">
    <axapi_method_list>
        <axapi_method>
            <name>authenticate</name>
            <privilege>0</privilege>
            <json_enabled>1</json_enabled>
        </axapi_method>
        <axapi_method>
            <name>slb.server.delete</name>
            <privilege>1</privilege>
            <json_enabled>1</json_enabled>
        </axapi_method>
        <axapi_method>
            <name>slb.server.deleteAll</name>
            <privilege>1</privilege>
            <json_enabled>1</json_enabled>
        </axapi_method>
    </axapi_method_list>
</response>
```

```

</axapi_method>
<axapi_method>
  <name>slb.server.deletePort</name>
  <privilege>1</privilege>
  <json_enabled>1</json_enabled>
</axapi_method>
<axapi_method>
  <name>slb.server.deleteAllPorts</name>
  <privilege>1</privilege>
  <json_enabled>1</json_enabled>
</axapi_method>
<axapi_method>
  <name>slb.server.fetchStatistics</name>
  <privilege>0</privilege>
  <json_enabled>1</json_enabled>
</axapi_method>
<axapi_method>
  <name>slb.server.fetchAllStatistics</name>
  <privilege>0</privilege>
  <json_enabled>1</json_enabled>
</axapi_method>
</axapi_method_list>
</response>

```

2 HTTP Request in JSON Format:

session_id=123456789&method=axapi.method.getAll&format=json&version=v1

Response as the HTTP body (JSON Format):

```
{
  "axapi_method_list": [
    {
      "name": "authenticate",
      "privilege": 0,
      "json_enabled": 1
    },
    {
      "name": "session.close",
      "privilege": 0,
      "json_enabled": 1
    },
    {
      "name": "system.config-file.upload",
      "privilege": 0,
      "json_enabled": 1
    },
    {
      "name": "system.config-file.restore",
      "privilege": 1,
      "json_enabled": 1
    },
    {
      "name": "slb.service-group.getAll",
      "privilege": 0,
      "json_enabled": 1
    },
    {
      "name": "slb.service-group.getByName",
      "privilege": 0
    }
  ]
}
```

```

        "json_enabled": 1
    },
    {
        "name": "slb.service-group.create",
        "privilege": 1,
        "json_enabled": 1
    },
    {
        "name": "slb.service-group.update",
        "privilege": 1,
        "json_enabled": 1
    },
]
}

```

Error Codes

TBD

13.2 “axapi.method.search” Method

This method searches an aXAPI method by name and version.

Parameters

Parameter Name	Description	Data Type	Range	Default
session_id (*)	user authenticated session id.	String		
method (*)	axapi.method.search	String		
Version	Version of axapi methods	String	V1.0/1.1/1.2/1.3/2.0	V2
Name(*)	The name of axapi method Or the substring of axapi method name	String		

Note: (*) parameter is required in the API.

Response Fields

The response XML is a collection of the aXAPI methods that have been searched. Each aXAPI method will have a collection of method members.

axapi_method_list

axapi_method

name

privilege

json_enabled

XML tag for the collection of the aXAPI methods

XML tag for the aXAPI method

the name of the aXAPI method

the privilege of the aXAPI method

shows whether the aXAPI method is enabled for json ,0(false), 1(true)

Example Response

Request as:

- URL:

https://_AX_IP_Address_/services/rest/V2/

- HTTP Action:

GET

1 HTTP Request in URL Format:

```
session_id=123456789&method=axapi.method.search&format=url&version=v1&name=slb
```

Response as the HTTP body (URL Format):

```
<response status="ok">
  <axapi_method_list>
    <axapi_method>
      <name>slb.service-group.getAll</name>
      <privilege>0</privilege>
      <json_enabled>1</json_enabled>
    </axapi_method>
    <axapi_method>
      <name>slb.service-group.getByName</name>
      <privilege>0</privilege>
      <json_enabled>1</json_enabled>
    </axapi_method>
    <axapi_method>
      <name>slb.service-group.create</name>
      <privilege>1</privilege>
      <json_enabled>1</json_enabled>
    </axapi_method>
    <axapi_method>
      <name>slb.service-group.update</name>
      <privilege>1</privilege>
      <json_enabled>1</json_enabled>
    </axapi_method>
    <axapi_method>
      <name>slb.service-group.delete</name>
      <privilege>1</privilege>
      <json_enabled>1</json_enabled>
    </axapi_method>
    <axapi_method>
      <name>slb.service-group.deleteAll</name>
      <privilege>1</privilege>
      <json_enabled>1</json_enabled>
    </axapi_method>
    <axapi_method>
      <name>slb.service-group.deleteMember</name>
      <privilege>1</privilege>
      <json_enabled>1</json_enabled>
    </axapi_method>
  </axapi_method_list>
</response>
```

2 HTTP Request in JSON Format:

```
session_id=123456789&method=axapi.method.search&format=json&version=v1&name=slb
```

Response as the HTTP body (JSON Format):

```
{
  "axapi_method_list": [
    {
      "name": "slb.service-group.getAll",
      "privilege": 0,
      "json_enabled": 1
    },
    {
      "name": "slb.service-group.getByName",
      "privilege": 0,
      "json_enabled": 1
    }
  ]
}
```

```
},
{
  "name": "slb.service-group.create",
  "privilege": 1,
  "json_enabled": 1
},
{
  "name": "slb.service-group.update",
  "privilege": 1,
  "json_enabled": 1
},
{
  "name": "slb.service-group.delete",
  "privilege": 1,
  "json_enabled": 1
},
{
  "name": "slb.service-group.deleteAll",
  "privilege": 1,
  "json_enabled": 1
},
{
  "name": "slb.service-group.deleteMember",
  "privilege": 1,
  "json_enabled": 1
},
]
}
```

Error Codes

TBD

14 Error Messages

The table below lists the error messages that can be returned by the aXAPI.

Note: A10 Networks recommends that applications use the error codes rather than the error text to detect and handle errors, because the message text may change over time while the error codes should be more stable.

TABLE - aXAPI Error Messages

Message Number	Message Code	Message Text
1001	WS_RESULT_CODE_HTTPS	The web service only accepts the HTTPS protocol.
1002	WS_RESULT_CODE_MEM	Memory fault
1003	WS_RESULT_CODE_INPARAM	Invalid parameters
1004	WS_RESULT_CODE_INVER	Invalid version parameter
1005	WS_RESULT_CODE_NOT_WRITABLE	Access denied: no write privilege
1006	WS_RESULT_CODE_NOT_PRIVILEGE	Access denied: no privilege for this method
1007	WS_RESULT_CODE_AUTH_FAIL	Authentication failed
1008	WS_RESULT_CODE_AUTH_MATCH	Invalid web service method name
1009	WS_RESULT_CODE_AUTH_SID	Invalid session ID
1010	WS_RESULT_CODE_SESS_CLOSE	Failed to close aXAPI session
1011	WS_RESULT_CODE_SRVGRP_NAMETAG	Missing name parameter tag
1012	WS_RESULT_CODE_SRVGRP_NAME	Can not find the service group name
1013	WS_RESULT_CODE_SRVGRP_TYPE	Invalid service group type. Must be 2 or 3.
1014	WS_RESULT_CODE_SRVGRP_MTD	Invalid load-balancing method

1015	WS_RESULT_CODE_SRVGRP_IP	Invalid server IP address
1016	WS_RESULT_CODE_SRVGRP_IPTYPE	Only one type of service group IP address is supported (IPv4 or IPv6)
1017	WS_RESULT_CODE_SRVGRP_PORT	Invalid service group port number. Valid range is 0-65535.
1018	WS_RESULT_CODE_SRVGRP_CONNLIMIT	Invalid service group conn_limit. Valid range is 1-1000000.
1019	WS_RESULT_CODE_SRVGRP_CONNRESUME	Invalid service group conn_resume. Valid range is 0-1000000.
1020	WS_RESULT_CODE_SRVGRP_WEIGHT	Invalid service group weight. Valid range is 1-100.
1021	WS_RESULT_CODE_SRVGRP_PRI	Invalid service group priority. Valid range is 1-16.
1022	WS_RESULT_CODE_SRVGRP_STATUS	Invalid service group status. Must be 0 or 1.
1023	WS_RESULT_CODE_SRVGRP_MATCHMEM	Can not find the service group member
1024	WS_RESULT_CODE_SRVGRP_MEMLIST	Invalid service group member list parameter
1025	WS_RESULT_CODE_VIRTUAL_DUP	The virtual server already exists
1026	WS_RESULT_CODE_VIRTUAL_NAMETAG	Missing name parameter tag
1027	WS_RESULT_CODE_VIRTUAL_NAME	Can not find the virtual server name
1028	WS_RESULT_CODE_VIRTUAL_IPTAG	Missing address parameter tag
1029	WS_RESULT_CODE_VIRTUAL_IP	Invalid virtual server IP address
1030	WS_RESULT_CODE_VIRTUAL_STATUS	Invalid virtual server status. Must be 0 or 1.
1031	WS_RESULT_CODE_VIRTUAL_HA	Invalid virtual server HA group. Must be 1 or 2.
1032	WS_RESULT_CODE_VIRTUAL_ARP	Invalid virtual server ARP status. Must be 0 or 1.
1033	WS_RESULT_CODE_VIRTUAL_TYPE	Invalid virtual server port type. Must be 2, 3, 8, 9, 10, 11, 12, 14, 15, 16 or 17.

1034	WS_RESULT_CODE_VIRTUAL_PORT	Invalid virtual server port. Valid range is 0-65534.
1035	WS_RESULT_CODE_VIRTUAL_SRVGRP	Invalid virtual server port service group parameter
1036	WS_RESULT_CODE_VIRTUAL_VPORT	Invalid virtual server service list parameter
1037	WS_RESULT_CODE_VIRTUAL_CONNLIMIT	Invalid virtual server port conn_limit. Valid range is 1-1000000.
1038	WS_RESULT_CODE_VIRTUAL_ACL	Invalid virtual server port ACL number. Valid range is 1-199.
1039	WS_RESULT_CODE_VIRTUAL_VPORT_HA	Invalid virtual server port ha_conn_mirror state. Must be 0 or 1.
1040	WS_RESULT_CODE_VIRTUAL_DSR	Invalid virtual server port DSR state. Must be 0 or 1.
1041	WS_RESULT_CODE_VIRTUAL_SYN	Invalid virtual server port syn_cookie state. Must be 0 or 1.
1042	WS_RESULT_CODE_VIRTUAL_INMATCH	Can not find the virtual server
1043	WS_RESULT_CODE_VIRTUAL_VPORTINMATCH	Can not find the virtual server port
1044	WS_RESULT_CODE_VIRTUAL_AFLEX	aFlex syntax or compiler error
1045	WS_RESULT_CODE_SRVNODE_DUP	The server already exists
1046	WS_RESULT_CODE_SRVNODE_NAMETAG	Missing name parameter tag
1047	WS_RESULT_CODE_SRVNODE_NAME	Can not find the server name
1048	WS_RESULT_CODE_SRVNODE_IPTAG	Missing address parameter tag
1049	WS_RESULT_CODE_SRVNODE_IP	Invalid server IP address
1050	WS_RESULT_CODE_SRVNODE_STATUS	Invalid server status. Must be 0 or 1.
1051	WS_RESULT_CODE_SRVNODE_CONNLIMIT	Invalid server connection limit. Valid range is 1-1000000.
1052	WS_RESULT_CODE_SRVNODE_CONNRESUME	Invalid server connection resume value. Valid range is 0-1000000.

1053	WS_RESULT_CODE_SRVNODE_LIMIT_LT_RESERVED	Connection resume value cannot be larger than connection limit value
1054	WS_RESULT_CODE_SRVNODE_WEIGHT	Invalid server weight. Valid range is 1-100.
1055	WS_RESULT_CODE_SRVNODE_PORT	Invalid server port number. Valid range is 0-65534.
1056	WS_RESULT_CODE_SRVNODE_TYPE	Invalid server port protocol. Must be 2 or 3.
1057	WS_RESULT_CODE_SRVNODE_PORT_LIST	Invalid server port list parameter
1058	WS_RESULT_CODE_SRVNODE_INMATCH	Can not find the server
1059	WS_RESULT_CODE_SRVNODE_MATCHPORT	Can not find the server port
1060	WS_RESULT_CODE_IF_PORT	Can not find the port parameter
1061	WS_RESULT_CODE_IF_INMATCH	Can not find the interface
1062	WS_RESULT_CODE_CONFIG_FILETYPETAG	Missing file-type parameter tag
1063	WS_RESULT_CODE_CONFIG_FILETYPE	Can not find the file-type
1064	WS_RESULT_CODE_CONFIG_FILENAME_NO_EXIST	Can not find the filename in AX
1065	WS_RESULT_CODE_CONFIG_PROTOCOLTAG	Missing protocol parameter tag
1066	WS_RESULT_CODE_CONFIG_PROTOCOL	Invalid protocol. Must be ftp, tftp, rcp or scp.
1067	WS_RESULT_CODE_CONFIG_HOSTTAG	Missing host parameter tag
1068	WS_RESULT_CODE_CONFIG_HOST	Can not find the host
1069	WS_RESULT_CODE_CONFIG_SAVEFILETAG	Missing save-filename parameter tag
1070	WS_RESULT_CODE_CONFIG_SAVEFILE	Can not find the save-filename
1071	WS_RESULT_CODE_CONFIG_FILEFMT	Invalid file-format. Must be text or tgz.

1072	WS_RESULT_CODE_CONFIG_PORTTAG	Missing port parameter tag
1073	WS_RESULT_CODE_CONFIG_PORT	Invalid port number. Valid range is 1-65534.
1074	WS_RESULT_CODE_EXIST_HEALTH_MONITOR	Health Monitor does not exist
1075	WS_RESULT_CODE_EXIST_SERVICEGROUP	Service group does not exist
1076	WS_RESULT_CODE_EXIST_AFLEX	aFlex does not exist
1077	WS_RESULT_CODE_SYS_BUSY	aXAPI is still processing a previous request. Please try again later.
1078	WS_RESULT_CODE_EXIST_HTTP_TEMPL	Http template does not exist
1079	WS_RESULT_CODE_EXIST_CONNREUSE_TEMPLATE	Connection reuse template does not exist
1080	WS_RESULT_CODE_EXIST_TCPPROXY_TEMPLATE	TCP proxy template does not exist
1081	WS_RESULT_CODE_EXIST_PERSISCOOKIE_TEMPLATE	Persistent cookie template does not exist
1082	WS_RESULT_CODE_EXIST_PERSISSRCIP_TEMPLATE	Persistent source IP template does not exist
1083	WS_RESULT_CODE_EXIST_CLISSL_TEMPLATE	Client SSL template does not exist
1084	WS_RESULT_CODE_EXIST_SRVSSL_TEMPLATE	Server SSL template does not exist
1085	WS_RESULT_CODE_EXIST_TCP_TEMPLATE	TCP template does not exist
1086	WS_RESULT_CODE_EXIST_UDP_TEMPLATE	UDP template does not exist
1087	WS_RESULT_CODE_EXIST_RTSP_TEMPLATE	RTSP template does not exist
1088	WS_RESULT_CODE_EXIST_SMTP_TEMPLATE	SMTP template does not exist
1089	WS_RESULT_CODE_EXIST_SIP_TEMPLATE	SIP template does not exist
1090	WS_RESULT_CODE_EXIST_PBSLB_POLICY	PBSLB policy template does not exist

