

API帮助文档

版本 AF8.0.35

生成日期 2020/11/05 11:30:18



目录

目录	2
0 使用说明	14
0.1 前言	14
0.2 交互流程与报文格式	14
0.3 HTTP请求头格式	19
0.4 HTTP应答头格式	19
0.5 应答报文格式及返回码	20
0.6 注意事项	21
1 认证鉴权	22
1.1 用户登录	22
1.2 用户注销	23
1.3 token保活	24
2 对象	25
2.1 网络对象	25
2.1.1 获取网络对象	25
2.1.2 获取单个网络对象	28
2.1.3 批量获取网络对象	30
2.1.4 添加网络对象	33
2.1.5 删除网络对象	36
2.1.6 批量删除ip组信息	38
2.1.7 修改网络对象	41
2.1.8 全量修改网络对象	44
2.2 服务器识别	47
2.2.1 服务器识别高级配置	47
2.2.2 获取服务器识别高级配置	48
2.2.3 获取服务器识别结果	49
2.2.4 获取忽略服务器列表	51
2.2.5 将服务器IP添加到忽略服务器列表	52
2.2.6 将服务器IP批量添加到忽略服务器列表	53
2.2.7 删除一条忽略服务器数据	54
2.2.8 批量删除忽略服务器数据	55
2.2.9 获取指定服务器敏感数据详情	56
2.2.10 获取网络对象敏感数据详情	57

2.2.11 获取全部忽略敏感数据详情	58
2.2.12 批量忽略敏感数据详情	60
2.2.13 忽略单个敏感数据详情	62
2.2.14 删除单个敏感数据详情	63
2.2.15 批量删除忽略敏感数据详情	65
2.3 内容识别	67
2.3.1 应用识别	67
2.3.1.1 获取应用特征识别库版本日期有效期规则数量 ⁶⁷	
2.3.1.2 获取应用名称或应用分类列表	68
2.3.1.3 获取单个应用名称或应用分类	71
2.3.1.4 删除单个自定义的应用	73
2.3.1.5 批量删除自定义的应用	75
2.3.1.6 新增自定义应用或应用分类	77
2.3.1.7 修改应用或应用分类	80
2.3.1.8 获取应用规则列表	82
2.3.1.9 获取单个应用规则	86
2.3.1.10 新增自定义应用规则	88
2.3.1.11 删除单个自定义应用规则	92
2.3.1.12 批量删除自定义应用规则	95
2.3.1.13 修改应用规则	99
2.3.1.14 批量启禁用应用规则(通过规则名称启禁用) ¹⁰⁴	
2.3.1.15 批量启禁用应用规则(通过应用名称启禁用) ¹⁰⁷	
2.3.1.16 获取当前的自定义优先级	110
2.3.1.17 修改自定义优先级	111
2.4 URL分类库	112
2.4.1 获取URL分类库信息	112
2.4.2 获取指定url分类信息	113
2.4.3 添加自定义用户url数据	115
2.4.4 修改url分组信息	116
2.4.5 删除用户自定义url数据	118
2.4.6 批量删除URL信息	119
2.4.7 url查询	121
2.4.8 获取urlLib库版本信息	122
2.5 服务	123
2.5.1 获取预定义服务或自定义服务或服务组列表, 根据uri 参数servType指定	123
2.5.2 获取单条服务或服务组信息	126
2.5.3 批量获取服务或服务组信息	129

2.5.4 添加服务或服务组	133
2.5.5 修改指定服务或服务组	137
2.5.6 全量修改指定服务或服务组	141
2.5.7 删除指定服务或服务组	145
2.5.8 批量删除服务或服务组	147
2.6 IP地址库	151
2.6.1 IP归属地	151
2.6.1.1 获取IP归属地策略	151
2.6.1.2 查询一条IP归属地策略	154
2.6.1.3 添加一条IP归属地策略	155
2.6.1.4 修改IP归属地策略	157
2.6.1.5 删除IP归属地策略	159
2.6.1.6 批量删除IP归属地策略	161
2.6.1.7 全量修改IP归属地策略	163
2.6.1.8 查询IP归属地	165
2.6.1.9 查看本机地理位置	166
2.6.1.10 设置本机地理位置	167
2.6.2 国家地区列表	168
2.6.2.1 查看国家地区列表	168
2.6.3 isp地址段	170
2.6.3.1 查询IP地址对应的ISP地址段名称	170
2.6.3.2 获取所有ISP地址段	171
2.6.3.3 获取单个ISP地址段	172
2.6.3.4 添加单个ISP地址段	173
2.6.3.5 修改指定的ISP地址段	175
2.6.3.6 全量修改指定的ISP地址段	177
2.6.3.7 删除指定的ISP地址段	178
2.6.3.8 批量删除指定的ISP地址段	180
2.7 时间计划	181
2.7.1 获取所有时间计划	181
2.7.2 获取单个时间计划	184
2.7.3 添加单个时间计划	186
2.7.4 修改指定的时间计划	189
2.7.5 全量修改指定的时间计划	192
2.7.6 删除指定的时间计划	194
2.7.7 批量删除指定的时间计划	196

3.1 区域	200
3.1.1 获取区域信息列表	200
3.1.2 获取单个区域信息	203
3.1.3 添加区域信息	204
3.1.4 删除指定区域信息	206
3.1.5 批量删除区域信息	207
3.1.6 全量修改指定区域信息	208
3.1.7 增量修改指定区域信息	210
3.1.8 批量修改区域信息	211
3.2 接口	213
3.2.1 获取接口列表	213
3.2.2 获取单个接口信息	219
3.2.3 添加接口信息	224
3.2.4 删除指定接口信息	233
3.2.5 增量修改指定接口信息	238
3.2.6 全量修改指定接口信息	247
3.2.7 批量删除接口	256
3.2.8 批量修改接口信息	265
3.2.9 获取指定接口默认mac	275
3.2.10 获取所有接口默认mac	276
3.3 PPPOE接口	277
3.3.1 获取全部PPPOE接口配置信息	277
3.3.2 获取单个PPPOE接口配置信息	280
3.3.3 添加PPPOE接口配置信息	281
3.3.4 删除PPPOE单个接口配置信息	283
3.3.5 批量删除pppoe接口配置	284
3.3.6 增量修改PPPOE单个接口配置信息	286
3.3.7 全量修改PPPOE单个接口配置信息	288
3.3.8 获取PPPOE接口状态信息	290
3.3.9 PPPOE接口连接操作	291
3.4 虚拟网线组	293
3.4.1 获取全部虚拟网线组配置信息	293
3.4.2 获取单个虚拟网线组配置信息	294
3.4.3 添加虚拟网线组配置信息	295
3.4.4 删除单个虚拟网线组配置信息	296
3.4.5 批量删除虚拟网线组	297
3.4.6 增量修改单个虚拟网线组配置信息	298
3.4.7 全量修改单个虚拟网线组配置信息	300

3.5 路由	301
3.5.1 静态路由IPV4	301
3.5.1.1 获取单条IPv4静态路由信息	301
3.5.1.2 获取IPv4静态路由列表	302
3.5.1.3 添加一条IPv4静态路由信息	305
3.5.1.4 添加多条IPv4静态路由信息	307
3.5.1.5 删除指定的IPv4静态路由信息	310
3.5.1.6 通过参数删除指定的IPv4静态路由信息	311
3.5.1.7 删除多条IPv4静态路由信息	313
3.5.1.8 全量修改指定的IPv4静态路由信息	316
3.5.1.9 增量修改指定IPv4静态路由信息	318
3.5.1.10 增量修改多条IPv4静态路由信息	320
3.5.2 静态路由IPV6	323
3.5.2.1 获取单条IPv6静态路由信息	323
3.5.2.2 获取IPv6静态路由列表	324
3.5.2.3 添加一条IPv6静态路由信息	327
3.5.2.4 添加多条IPv6静态路由信息	329
3.5.2.5 删除指定的IPv6静态路由信息	331
3.5.2.6 通过参数删除指定的IPv6静态路由信息	332
3.5.2.7 删除多条IPv6静态路由信息	334
3.5.2.8 全量修改指定的IPv6静态路由信息	337
3.5.2.9 增量修改指定IPv6静态路由信息	339
3.5.2.10 增量修改多条IPv6静态路由信息	341
3.5.3 策略路由IPV4	344
3.5.3.1 获取单条IPv4策略路由信息	344
3.5.3.2 获取IPv4策略路由列表	346
3.5.3.3 添加一条IPv4策略路由信息	348
3.5.3.4 添加多条IPv4策略路由信息	352
3.5.3.5 删除指定的IPv4策略路由信息	356
3.5.3.6 删除多条指定的IPv4策略路由信息	358
3.5.3.7 全量修改指定的IPv4策略路由信息	361
3.5.3.8 增量修改指定IPv4策略路由信息	364
3.5.3.9 增量修改多条IPv4策略路由信息	367
3.5.4 策略路由IPV6	370
3.5.4.1 获取单条IPv6策略路由信息	370
3.5.4.2 获取IPv6策略路由列表	372
3.5.4.3 添加一条IPv6策略路由信息	375
3.5.4.4 添加多条IPv6策略路由信息	379

3.5.4.5 删除指定的IPv6策略路由信息	383
3.5.4.6 删除指定的多条IPv6策略路由信息	384
3.5.4.7 全量修改指定的IPv6策略路由信息	388
3.5.4.8 增量修改指定IPv6策略路由信息	390
3.5.4.9 增量修改多条IPv6策略路由信息	393
3.5.5 查询IPV4路由状态	397
3.5.5.1 获取后台IPv4路由信息列表	397
3.5.6 查询IPV6路由状态	399
3.5.6.1 获取后台IPv6路由信息列表	399
3.5.7 IPV4路由测试	401
3.5.7.1 测试路由	401
4 策略	403
4.1 地址转换	403
4.1.1 NAT策略	403
4.1.1.1 模拟匹配	403
4.1.1.2 获取nat策略的匹配数	404
4.1.1.3 批量清除nat匹配数	405
4.1.1.4 单个清除nat匹配数	406
4.1.1.5 获取符合过滤条件的全部nat	407
4.1.1.6 批量查询NAT规则	413
4.1.1.7 获取单条nat详细信息	422
4.1.1.8 添加nat	427
4.1.1.9 删除单条nat	436
4.1.1.10 增量修改一条nat配置	441
4.1.1.11 全量修改单条nat	450
4.1.1.12 批量删除nat对象	459
4.1.1.13 批量增量更新nat对象	468
4.1.2 NAT6策略	477
4.1.2.1 获取符合过滤条件的全部nat6	477
4.1.2.2 获取单条nat6详细信息	480
4.1.2.3 添加nat6	482
4.1.2.4 删除单条nat6	485
4.1.2.5 批量删除nat6策略	487
4.1.2.6 批量修改nat6	490
4.1.2.7 全量修改nat6	494
4.1.2.8 增量修改nat6	497
4.1.3 NAT64策略	500

4.1.3.1	获取nat64策略的匹配数	500
4.1.3.2	获取符合过滤条件的全部nat64	501
4.1.3.3	获取单条nat64详细信息	507
4.1.3.4	添加nat64	510
4.1.3.5	删除单条nat64策略	517
4.1.3.6	批量删除nat64	520
4.1.3.7	全量修改nat64	526
4.1.3.8	增量修改nat64	533
4.1.3.9	批量修改nat64	539
4.1.4	dnsmap	546
4.1.4.1	获取匹配上的dnsmap数据	546
4.1.4.2	获取单条Dnsmapping详细信息	547
4.1.4.3	使用整个数据删除单条Dnsmapping	548
4.1.4.4	添加Dnsmapping	549
4.1.4.5	批量删除Dnsmapping	550
4.1.4.6	全量修改Dnsmapping	552
4.1.4.7	删除单条Dnsmapping	553
4.2	访问控制	554
4.2.1	应用控制策略相关	554
4.2.1.1	应用控制策略	554
4.2.1.1.1	获取策略命中次数	554
4.2.1.1.2	批量获取策略命中次数	555
4.2.1.1.3	清除策略命中次数	556
4.2.1.1.4	批量清除策略命中次数	557
4.2.1.1.5	开始同步策略冲突检查	558
4.2.1.1.6	开始异步策略冲突检查	558
4.2.1.1.7	获取策略冲突检查状态	559
4.2.1.1.8	取消策略冲突检查	560
4.2.1.1.9	获取策略冲突列表	561
4.2.1.1.10	获取指定的策略冲突信息	563
4.2.1.1.11	批量获取指定的策略冲突信息	565
4.2.1.1.12	获取应用控制策略列表	568
4.2.1.1.13	获取指定应用控制策略	573
4.2.1.1.14	增加单项应用控制策略	576
4.2.1.1.15	删除单项应用控制策略	581
4.2.1.1.16	批量删除应用控制策略	584
4.2.1.1.17	修改单项应用控制策略	588
4.2.1.1.18	批量修改应用控制策略	593

4.2.1.1.19	获取应用控制策略全局属性	598
4.2.1.1.20	修改应用控制策略全局属性	599
4.2.1.2	应用控制策略组	600
4.2.1.2.1	获取应用控制策略组	600
4.2.1.2.2	批量获取应用控制策略组	602
4.2.1.2.3	获取指定应用控制策略组	603
4.2.1.2.4	增加单项应用控制策略组	604
4.2.1.2.5	删除单项应用控制策略组	606
4.2.1.2.6	修改单项应用控制策略组	607
4.2.1.3	应用控制策略标签	609
4.2.1.3.1	获取应用控制策略标签列表	609
4.2.1.3.2	获取指定应用控制策略标签	610
4.2.1.3.3	增加单项应用控制策略标签	611
4.2.1.3.4	删除单项应用控制策略标签	613
4.2.1.3.5	批量添加应用控制策略标签	614
4.2.1.3.6	批量删除应用控制策略标签	615
4.2.1.3.7	修改单项应用控制策略标签	617
4.2.1.4	策略生命周期管理	618
4.2.1.4.1	获取生命周期日志列表	618
4.2.1.4.2	获取生命周期管理日志对比详情	621
4.2.1.4.3	获取策略修改日志策略名或用户名列表	625
4.2.1.4.4	导出策略修改日志	627
4.2.1.4.5	修改策略修改日志导出列表	628
4.2.1.4.6	获取策略修改日志导出列表	629
4.2.2	地域访问控制	629
4.2.2.1	地域访问控制策略	630
4.2.2.1.1	获取地域访问控制策略	630
4.2.2.1.2	获取单条地域访问控制策略	632
4.2.2.1.3	新增地域访问控制策略	633
4.2.2.1.4	修改地域访问控制策略	635
4.2.2.1.5	全量修改地域访问控制策略	637
4.2.2.1.6	删除地域访问控制策略	639
4.2.2.1.7	批量删除地域访问控制策略	640
4.2.2.1.8	批量修改地域访问控制策略	643
4.2.2.2	排除列表	646
4.2.2.2.1	查看地域访问控制排除列表	646
4.2.2.2.2	修改地域访问控制排除列表	646

4.2.2.3 已拒绝的IP	648
4.2.2.3.1 查看已拒绝的IP	648
4.2.2.3.2 清空已拒绝IP列表	649
4.2.2.3.3 清零一个已拒绝IP的阻断次数	650
4.2.2.4 内网访问控制	650
4.2.2.4.1 查看内网访问控制策略	651
4.2.2.4.2 修改内网访问控制策略	651
4.2.3 连接数控制	653
4.2.3.1 获取连接数控制策略	653
4.2.3.2 获取一条连接数控制策略	656
4.2.3.3 添加连接数控制策略	657
4.2.3.4 修改连接数控制策略	659
4.2.3.5 全量修改连接数控制策略	662
4.2.3.6 删除一条连接数控制策略	664
4.2.3.7 批量删除连接数控制策略	665
4.2.3.8 批量修改连接数控制策略	667
4.3 认证	670
4.3.1 用户组管理	670
4.3.1.1 认证用户	670
4.3.1.1.1 查看某个组下的认证用户列表	670
4.3.1.1.2 获取一个认证用户	675
4.3.1.1.3 添加一个认证用户	677
4.3.1.1.4 删除一个认证用户	682
4.3.1.1.5 增量更新一个认证用户	684
4.3.1.1.6 全量更新一个认证用户	688
4.3.1.1.7 批量删除认证用户	693
4.3.1.1.8 批量新增认证用户	697
4.3.1.1.9 批量更新认证用户	701
4.3.1.2 认证用户组	705
4.3.1.2.1 获取指定认证用户组的信息	705
4.3.1.2.2 获取用户组列表	706
4.3.1.2.3 新增一个认证用户组	708
4.3.1.2.4 删除一个认证用户组	709
4.3.1.2.5 全量更新一个认证用户组	710
4.3.1.2.6 增量更新一个认证用户组	712
4.3.1.2.7 批量删除认证用户组	713
4.3.1.2.8 批量新增认证用户组	714
4.3.1.2.9 批量更新用户组	716

4.3.2 用户认证	717
4.3.2.1 认证策略	717
4.3.2.1.1 获取全部认证策略	717
4.3.2.1.2 获取指定认证策略	720
4.3.2.1.3 新增一条认证策略	722
4.3.2.1.4 删除一条认证策略	726
4.3.2.1.5 批量删除认证策略	728
4.3.2.1.6 全量更新一条认证策略	732
4.3.2.1.7 增量更新一条认证策略	737
4.3.2.1.8 批量更新认证策略	741
4.3.2.1.9 设置用户认证开关状态	745
4.3.2.1.10 获取用户认证状态	746
4.3.2.1.11 更新认证区域	747
4.3.2.1.12 查看认证区域	748
4.3.3 用户认证状态	749
4.3.3.1 获取在线用户列表	749
4.3.3.2 注销在线用户	751
4.3.3.3 修改在线用户状态	752
5 系统	754
5.1 管理员账户	754
5.1.1 查询所有已创建的管理员账户信息	754
5.1.2 查询一个已创建的管理员账户信息	773
5.1.3 查询管理员账户的不同角色的默认权限模板	781
5.1.4 新增一个管理员账户	788
5.1.5 修改一个管理员账户的信息	803
5.1.6 删除一个管理员账户	817
5.1.7 批量删除管理员账户	825
5.1.8 批量修改管理员账户启用、禁用状态	839
5.1.9 获取账户密码安全策略	853
5.1.10 修改账户密码安全策略	854
5.2 排障	855
5.2.1 获取系统模块列表	855
5.2.2 获取系统故障日志	856
5.3 高可用	858
5.3.1 获取所有组双机的状态	858
5.3.2 获取指定组双机的状态	859
5.4 安全能力更新	861

5.4.1 获取安全能力更新库列表	861
5.4.2 升级规则库	865
5.5 系统维护	866
5.5.1 系统升级	867
5.5.1.1 获取系统升级日志	867
5.6 导入导出	868
5.6.1 ftp文件上传下载	868
6 运营中心	870
6.1 黑白名单	870
6.1.1 获取符合过滤条件的全部黑白名单	870
6.1.2 获取单个黑白名单详细信息	872
6.1.3 添加黑白名单	873
6.1.4 批量添加黑白名单信息	875
6.1.5 删除单个黑白名单	877
6.1.6 删除自定义黑白名单	878
6.1.7 批量删除黑白名单信息	879
6.1.8 修改黑名单配置	881
6.1.9 批量修改黑白名单信息	882
6.2 封锁攻击者IP	884
6.2.1 获取封锁攻击者IP列表	884
6.2.2 批量添加封锁攻击者	887
6.2.3 批量删除封锁攻击者	888
6.2.4 清空封锁攻击者	892
6.2.5 全量修改自动封锁攻击者时间	893
6.2.6 获取自动封锁攻击者时间	894
7 状态中心	896
7.1 总览	896
7.1.1 设备信息	896
7.1.1.1 获取内存使用率	896
7.1.1.2 获取内存使用率变化曲线数据	896
7.1.1.3 获取cpu使用率	898
7.1.1.4 获取cpu使用率变化曲线数据	899
7.1.1.5 获取设备运行时间	901
7.1.1.6 获取当前时间	901
7.1.1.7 获取磁盘信息数据	902
7.1.1.8 获取硬盘使用率	903

7.1.1.9 获取系统版本信息	904
7.1.1.10 获取接口状态信息	906
7.1.2 网络活动状态	908
7.1.2.1 获取接口吞吐率折线图	908
7.2 监控	910
7.2.1 流量排行	910
7.2.1.1 获取应用流量排行	910
7.2.1.2 获取用户流量排行	912
7.2.1.3 获取IP流量趋势	915
7.2.2 会话排行	918
7.2.2.1 获取会话数量排行	918
7.2.2.2 获取监听列表IP	921
7.2.2.3 添加一条监听列表IP范围	923
7.2.2.4 删除监听列表IP范围	924
7.2.2.5 全量覆盖监听列表IP	925
7.2.2.6 获取每日新建会话信息	927
7.2.2.7 获取会话详情	928
7.2.2.8 获取会话数量趋势折线图	931
7.2.2.9 获取指定源IP的会话详情	933
8 扩展API	937
8.1 引用关系	937
8.1.1 查询指定认证用户的引用关系	937
8.1.2 查询指定认证用户组的引用关系	938
8.1.3 查询指定ipgroup的引用关系	939
8.1.4 查询指定service的引用关系	940
8.1.5 查询指定serviceGroup的引用关系	942
8.1.6 查询指定ip地址库的引用关系	943
8.1.7 查询指定循环时间计划的引用关系	944
8.1.8 查询指定单次时间计划的引用关系	946
8.1.9 查找接口被双机和sslvpn的引用情况	947
8.1.10 查询指定区域的引用关系	949
8.1.11 查询Net指定应用的引用关系	950
8.1.12 查询指定应用的引用关系	951
8.1.13 查询应规则引用状态	953
8.2 配置完整性校验	954
8.2.1 配置完整性校验	954

0 使用说明

0.1 前言

1) 读者对象

本文档针对AF RESTful API接口功能，从预备知识、通用处理流程、数据模型以及开发环境搭建等方面介绍了API接口适配过程。本文档提供了AF RESTful API接口适配举例以及数据模型文件获取路径。

本文档主要适用于以下工程师：

- 技术支持工程师
- 维护工程师
- 第三方适配开发工程师

使用本文档的读者需要熟悉HTTP/HTTPS协议、RESTful设计原则等。

2) 修订记录

修改记录累积了每次文档更新的说明。最新版本的文档包含以前所有文档版本的更新内容。

3) 文档版本

首次发布。01 (2019-12-23)

0.2 交互流程与报文格式

1) 交互流程

设备作为API交互的服务端提供服务，在认证成功之前，发生任何错误API服务端均会断开连接。首先需要有一个具备WEB API权限的账户，用户可以登录WebUI通过"系统->管理员账号"栏将默认的账户勾上WEB API选项或者新增一个账户并勾选WEB API，如图1-1所示。而后需要调用login登录的API,将用户名和密码放在http协议的body里面，用post方法提交，API连接流程如图1-2，API客户端和服务端报文交互图如图1-3，登录成功后如图1-4所示。登录成功后，设备返回该账户的token值，后续用户需要将http请求头的cookie字段赋为该值作为检查，然后再次调用业务API即可，如图1-5所示为获取密码安全策略的API的GET请求示例。若10分钟（默认，可调整）没有持续发送API请求，则认为该用户超时退出，下次请求需要重新走认证流程。

新增管理员账号

×

用户名：

test

启用状态：

启用

禁用

描述：

test

角色：

普通管理员

ⓘ

登录安全设置

页面权限设置

认证策略：

账号密码认证

ⓘ

密码：

.....

ⓘ

确定密码：

.....

管理方式：

WEB控制台

Web API

命令行

原厂支持

确定

取消

图1-1 WebUI界面创建账户勾选WEB API选项

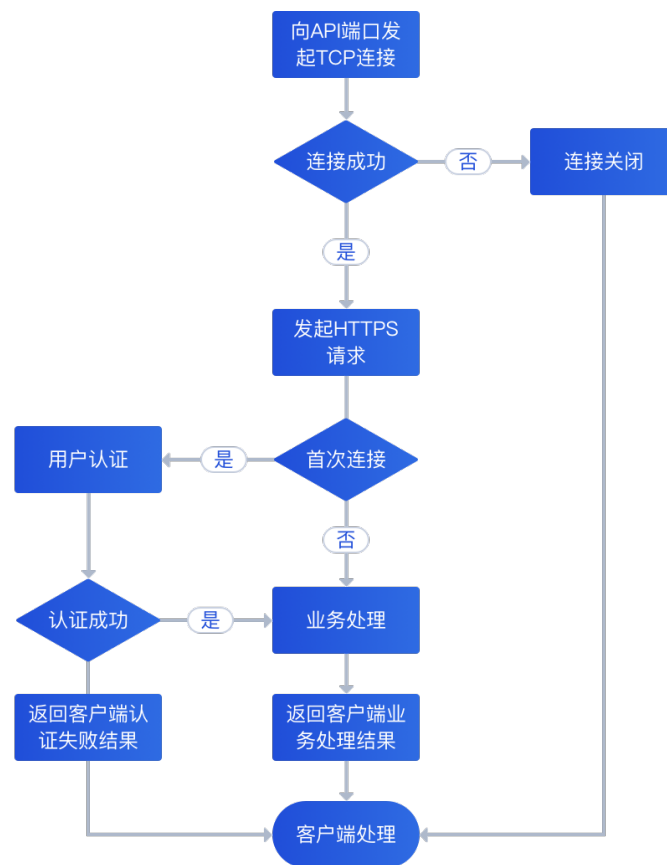


图1-2 API连接处理流程图

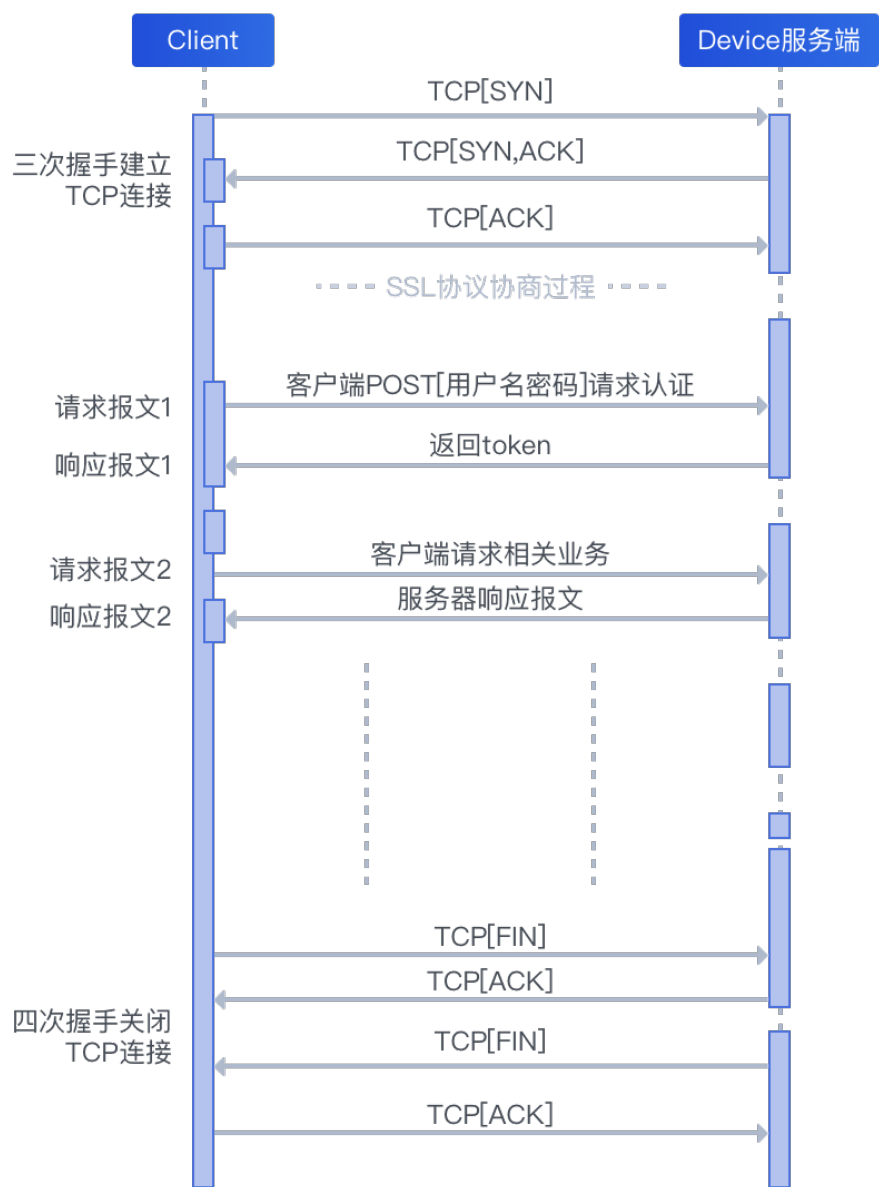


图1-3 API客户端与API服务端报文交互流程图

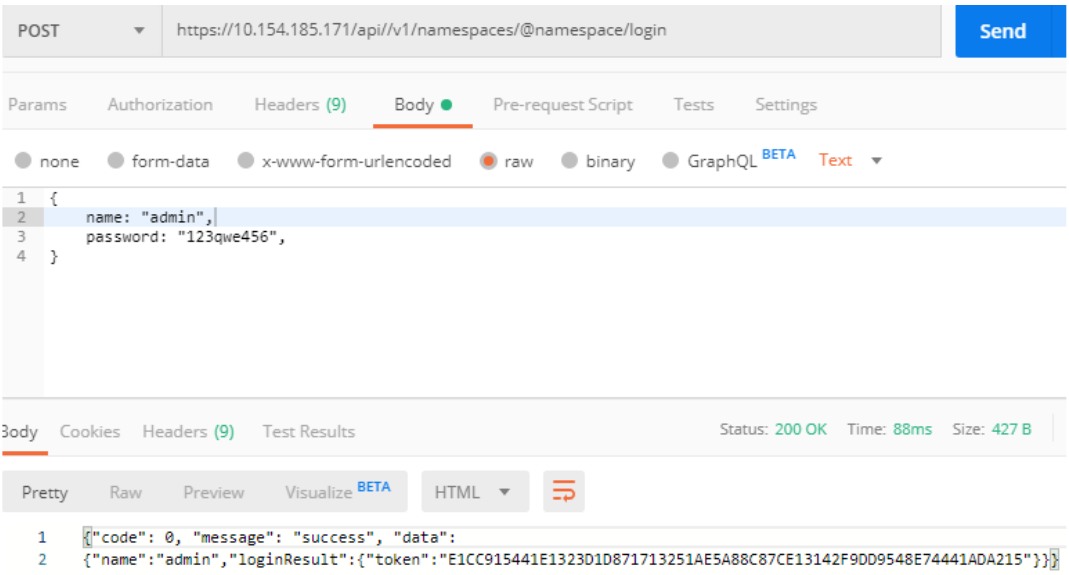


图1-4 登录API成功示例图

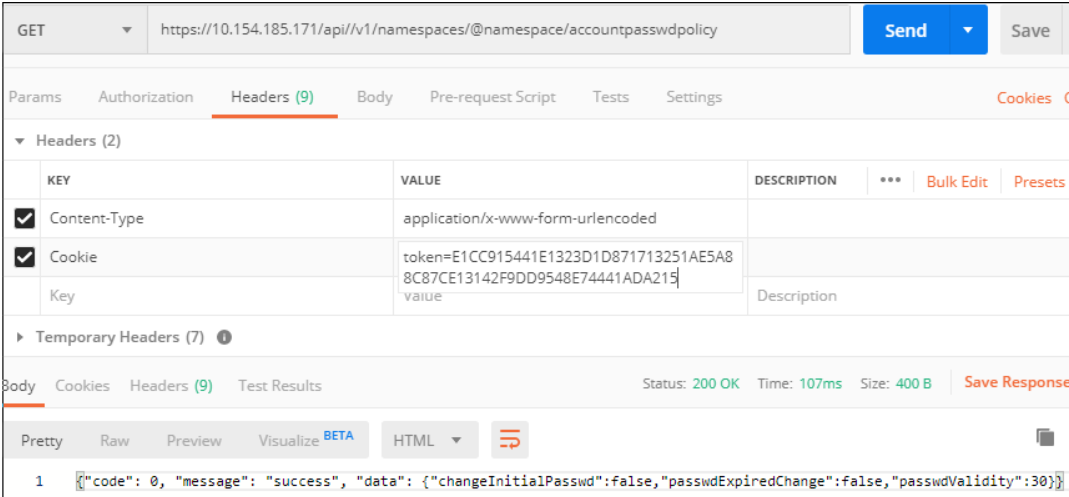


图1-5 获取密码安全策略API示例图

2) HTTP URL格式

- OP /api/version/namespaces/@namespace/resource/@name?query
- OP: 指的是HTTP方法，包括 POST、PUT、GET、PATCH、DELETE等。并不是所有的资源都支持这这5种方法。具体以各个资源说明文档为准。
- api: API 的根目录。
- version: 操作的目标资源的路径。
- namespaces: 虚拟子系统命名空间字段
- @namespace:虚拟子系统命名空间实例名字，当前只支持public即根系统
- Resource: 具体资源路径

@name: 具体资源子项名称，有些资源没有名称，需要用uuid去访问

query: ?后的参数，表示查询操作时的参数列表。

批量 URL 格式：

OP /api/bacth/version/namespaces/@namespace/resource/@name?query

该API主要场景是批量启禁用对象、策略等。

0.3 HTTP请求头格式

HTTP请求包含数据部分时，则必须包含字段Content-Length、Content-Type、Connection以及cookie。请求头字段名与字段值间必须以“:”分隔，分隔符与字段值之间最多支持一个空格或制表符。报文样例如下：

```
POST /api/v1/namespaces/@namespace/ipgroups HTTP/1.1
Connection: keep-alive
Cookie: token=E1CC915441E1323D1D871713251AE5A88C87CE13142F9D0D9548E7441ADA215
Content-Type: application/json
Content-Length: 166
{
  "name": "test_1",
  "businessType": "IP",
  "description": "aaa",
  "addressType": "IPV4",
  "important": "COMMON",
  "creator": "admin",
  "ipRanges": [
    {
      "start": "1.1.1.1",
      "end": "1.1.1.2"
    }
  ]
}
```

报文各字段含义解析如表1所示：

字段	说明
Content-Length	实际数据长度，没有数据部分时，值为0。
Content-Type	当HTTP/HTTPS Body内容为空时，该字段可不存在或取值为空。取值遵循HTTP协议标准。
Connection	遵循HTTP协议标准。
cookie	后续报文必须携带的字段，首次请求需要调用登录API获取。

0.4 HTTP应答头格式

应答报文样例如下：

```
HTTP/1.1 200 OK
Content-Length: 251
Content-Type: application/json;charset=UTF-8
Connection: close
Date: Tue, 24 Dec 2019 04:54:41 GMT
{
  "code": 0,
  "message": "success",
  "data": {
    "uuid": "D1E46218EC7744A48D1813785DA4D115",
    "name": "test_1",
    "creator": "admin",
    "businessType": "IP",
    "description": "aaa",
    "addressType": "IPV4",
    "ipRanges": [
      {
        "start": "1.1.1.1",

```

```
        "end": "1.1.1.2"
      },
    ],
    "important": "COMMON"
  }
}
```

HTTP请求应答头字段具体含义如下所示：

字段	说明
Date	发送应答的日期时间，格式为“Tue, 24 Dec 2019 04:54:41 GMT”。
Content-Length	表示实际数据长度，无数据时可以不携带或者值为0。
Content-Type	当HTTP/HTTPS Body内容为空时，该字段可不存在或取值为空。取值遵循HTTP协议标准。
Connection	遵循HTTP协议标准。

0.5 应答报文格式及返回码

本章节介绍错误应答消息报文体内容的基本格式与返回码的含义。

1) 返回码

表1描述了支持的应答消息返回码、含义以及携带报文体内容。

返回码	含义
0	成功
1	操作不允许，如认证失败
13	缺少对应接口调用权限
22	参数错误
27	文件太大
28	磁盘没有空间
110	超时
1001	接口及参数检查错误
1002	对应接口不存在
1003	未登陆
1004	对应资源不存在
1005	超出范围，如字符串过长
1006	请求被取消
1007	内部接口异常

2) 应答消息报文体格式

设备处理的应答消息除报文头携带不同返回码以区分不同类型的处理结果外，还可能在报文体中返回固定格式的信息以定位具体错误位置，固定格式如下所示：

```
{
  "code": 13,
  "message": "check: Permission denied",
  "data": ""
}
```

针对该固定格式解析如表4所述。

字段	含义
code	错误码，如上图1所示
message	错误信息，表示更为详细的错误提示，比如包含了具体业务具体对象出错的原因，并部分提供纠错引导。
data	执行成功返回关于提交的相关信息，执行失败该字段为空，没有实际意义。

3) 应答消息举例

用户在创建网络对象名称为test_1，范围为1.1.1.1–1.1.1.2，API服务端返回给客户端的完整应答消息如下：

```
HTTP/1.1 200 OK
Content-Length: 251
Content-Type: text/html; charset=UTF-8
Connection: close
Date: Tue, 24 Dec 2019 04:54:41 GMT
{
  "code": 0,
  "message": "success",
  "data": {}
}
```

其中：

```
"code": 0 是应答消息的返回码, 0表示成功。
"message": "success" 表示详细信息，执行成功默认返回success。
"data":
{
  "uuid": "D1E46218EC7744A48D1813785DA4D115",
  "name": "test_1",
  "creator": "admin",
  "businessType": "IP",
  "description": "aaa",
  "addressType": "IPv4",
  "ipRanges": [
    {
      "start": "1.1.1.1",
      "end": "1.1.1.2"
    }
  ],
  "important": "COMMON"
}
```

data表示执行成功后，设备内部对数据的处理，比如插入网络对象这里加了uuid字段等。上文举例描述的即为通用的应答消息内容，API接口支持的应答消息的格式与举例相同，具体值根据不同的资源、不同的请求及不同的错误类型填充。

0.6 注意事项

暂无。

1 认证鉴权

1.1 用户登录

功能说明：管理员账户可以通过本接口登录

1) API请求数据

POST https://{host}/api/v1/namespaces/@namespace/login

2) 输入参数

- Body参数

参数	类型	是否必选	描述
obj	object	是	调用login接口需要输入的信息
obj.name	string	是	账户名 最小长度:1 最大长度:60
obj.password	string	是	密码 最小长度:1 最大长度:512

请求样例：

登录

```
POST https://192.168.1.1/api/v1/namespaces/public/login
{
  "name": "admin",
  "password": "admin"
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	api执行后返回的数据
data.name	string	执行操作的管理员账户名 最小长度:1 最大长度:60
data.loginResult	object	登录后得到的数据
data.loginResult.token	string	登录后得到的会话令牌
data.passwdStatus	bool	密码状态，是否过期或需要修改

data.role	string	账户角色 枚举值: ADMINISTRATOR:超级管理员 COMMON:普通管理员 SAFE:三权分立账户, 安全管理员 LOG:三权分立账户, 审计员 SYSTEM:三权分立账户, 系统管理员 REMOTE:远程认证账户]
-----------	--------	--

请求成功样例:

```
{
  "data": {
    "loginResult": {
      "token": "3869A36E56525592B3AD88DADD6E87C0F67A3EAA4DFB3C223541EA10E611761"
    },
    "passwdStatus": true,
    "name": "admin",
    "role": "ADMINISTRATOR"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

1.2 用户注销

功能说明: 管理员账户可以通过本接口注销登录状态

1) API请求数据

POST https://{host}/api/v1/namespaces/@namespace/logout

2) 输入参数

请求样例:

注销

POST https://192.168.1.1/api/v1/namespaces/public/logout

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	api执行后返回的数据
data.name	string	执行操作的管理员账户名 最小长度:1 最大长度:60

请求成功样例:

```
{
  "data": {
    "name": "admin"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

1.3 token保活

功能说明: 已登录的账户可以通过本接口刷新token的时间戳, 从而保持token不超时

1) API请求数据

```
GET https://{host}/api/v1/namespaces/@namespace/keepalive
```

2) 输入参数

请求样例:

token保活

GET https://192.168.1.1/api/v1/namespaces/public/keepalive

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	int32	

请求成功样例:

```
{
  "data": 0,
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```


2 对象

2.1 网络对象

2.1.1 获取网络对象

功能说明：通过此接口，可以获得符合条件的全部网络对象

1) API请求数据

GET <https://{host}/api/v1/namespaces/@namespace/ipgroups>

2) 输入参数

• URL参数

参数	类型	是否必选	描述
_start	int32	否	起始位置, 从 0 开始 最小值:0 最大值:2000000
businessType	string	否	过滤参数：业务类型过滤 枚举值:[ADDRGROUP:业务类型为地址组 IP:业务类型是IP地址 USER:业务类型是用户地址 BUSINESS:业务类型是业务地址 ALL:不过滤业务类型]
__nameprefix	string	否	网络对象组名字前缀过滤 最小长度:1 最大长度:95
important	string	否	过滤参数：重要级别 枚举值:[COMMON:重要级别为普通 CORE:重要级别为核心 ALL:不过滤重要级别]
hasSensitiveData	string	否	过滤参数：是否过滤敏感数据业务 枚举值:[TRUE:过滤敏感数据业务 FALSE:过滤非敏感数据业务 ALL:不过滤敏感数据业务]
getRefBy	string	否	过滤参数：传入一个网络对象的uuid，查找指定网络对象被哪些地址组引用 最小长度:32 最大长度:32
_length	int32	否	获取最大长度，限制最大值，如果数据不够可能返回小于 length 最小值:1 最大值:200 默认值:true

hasref	string	否	过滤参数：是否过滤被引用的IP组 枚举值:[TRUE:过滤被引用的网络对象 FALSE:过滤没有被引用的网络对象]
_search	string	否	模糊搜索关键字 最大长度:95
_select	string	否	选择字段, 字段用逗号分隔 最大长度:1000
_order	string	否	指定排序方式（正序/倒序） 默认值:true 枚举值:[asc:按升序排列 desc:按降序排列]
_sortby	string	否	指定排序字段 最大长度:100
addressType	string	否	过滤参数：ip协议版本 枚举值:[IPV4:ip协议版本为ipv4 IPV6:ip协议版本为ipv6 ALL:不过滤ip协议版本]
excludeAll	string	否	过滤参数：是否过滤名字为全部的IP组 枚举值:[TRUE:过滤名字为全部的IP组 FALSE:不过滤名字为全部的IP组]

请求样例：

获取网络对象列表,默认查询条件：起始位置为0，最多展示200条

GET <https://192.168.1.1/api/v1/namespaces/public/ipgroups>

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	分页数据
data.totalItems	int32	总共多少项目
data.totalPages	int32	总共多少页
data.pageNumber	int32	当前页码，从 1 开始
data.pageSize	int32	每页多大
data.itemsOffset	int32	当前条目偏移, 从 0 开始
data.itemLength	int32	数据列表长度
data.items	object[]	有效数据列表

data.items.uuid	string	资源唯一id
data.items.name	string	网络对象组名字 最小长度:1 最大长度:95
data.items.businessType	string	网络对象组类型 枚举值:[IP:ip组类型 BUSINESS:业务组类型 USER:用户组类型 ADDRGROUP:地址组类型]
data.items.description	string	描述字段 最小长度:0 最大长度:95
data.items.addressType	string	地址类型 枚举值:[IPV4:IPv4类型 IPV6:IPv6类型]
data.items.ipRanges	object[]	ip表项组
data.items.ipRanges.start	string	ip范围的起始位置 格式:ip-address
data.items.ipRanges.end	string	ip范围的终止位置 格式:ip-address
data.items.ipRanges.bits	uint8	ip掩码简写
data.items.ipRanges.mask	string	ip掩码 格式:ip-address
data.items.important	string	标识核心业务/普通业务或者标识普通用户/核心用户 枚举值:[COMMON:普通用户或普通业务 CORE:核心用户或核心业务]
data.items.refIpGroup	string[]	自定义识别范围引用网络对象列表
data.items.dataStatus	string	标识敏感数据识别方式是自动识别或者存在或者不存在 枚举值:[AUTO:系统自动识别 EXIST:存在 NOT-EXIST:不存在]
data.items.hasref	bool	标识是否被引用
data.items.isdefault	bool	标识是否是默认配置
data.items.showname	string	标识默认配置的名字
data.items.pageNum	int32	敏感数据页面数

请求成功样例:

```
{
  "data": {
    "itemsOffset": 0,
```

```

    "items": [
      {
        "hasref": true,
        "businessType": "IP",
        "important": "COMMON",
        "uuid": "5FE8FDA10D8343CB9310E8B252E94C74",
        "ipRanges": [
          {
            "start": "0.0.0.0",
            "end": "255.255.255.255"
          }
        ],
        "isdefault": true,
        "name": "全部",
        "addressType": "IPv4",
        "description": "所有IP地址"
      },
      {
        "hasref": false,
        "businessType": "IP",
        "important": "COMMON",
        "uuid": "22A180730D98436DB4FD54BBB1674137",
        "ipRanges": [
          {
            "start": "10.0.0.0",
            "end": "10.255.255.255"
          },
          {
            "start": "172.16.0.0",
            "end": "172.31.255.255"
          },
          {
            "start": "192.168.0.0",
            "end": "192.168.255.255"
          }
        ],
        "isdefault": true,
        "name": "私有网段",
        "addressType": "IPv4",
        "description": "私有IP地址"
      }
    ],
    "totalItems": 2,
    "pageNumber": 1,
    "itemLength": 2,
    "totalPages": 1,
    "pageSize": 100
  },
  "code": 0,
  "message": "成功"
}

```

请求失败样例:

```

{
  "data": "",
  "code": 22,
  "message": "无效参数"
}

```

2.1.2 获取单个网络对象

功能说明: 通过此接口, 可以通过名字查询到单个网络对象

1) API请求数据

GET <https://{host}/api/v1/namespaces/@namespace/ipgroups/@name>

2) 输入参数

- URL参数

参数	类型	是否必选	描述
----	----	------	----

_search	string	否	模糊搜索关键字 最大长度:95
---------	--------	---	--------------------

请求样例:

获取指定网络对象的配置, 查询条件: name为全部

GET https://192.168.1.1/api/v1/namespaces/public/ipgroups/全部

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	网络对象
data.uuid	string	资源唯一id
data.name	string	网络对象组名字 最小长度:1 最大长度:95
data.businessType	string	网络对象组类型 枚举值:[IP:ip组类型 BUSINESS:业务组类型 USER:用户组类型 ADDRGROUP:地址组类型]
data.description	string	描述字段 最小长度:0 最大长度:95
data.addressType	string	地址类型 枚举值:[IPV4:IPv4类型 IPV6:IPv6类型]
data.ipRanges	object[]	ip表项组
data.ipRanges.start	string	ip范围的起始位置 格式:ip-address
data.ipRanges.end	string	ip范围的终止位置 格式:ip-address
data.ipRanges.bits	uint8	ip掩码简写
data.ipRanges.mask	string	ip掩码 格式:ip-address
data.important	string	标识核心业务/普通业务或者标识普通用户/核心用户 枚举值:[COMMON:普通用户或普通业务 CORE:核心用户或核心业务]
data.refIpGroup	string[]	自定义识别范围引用网络对象列表

data.dataStatus	string	标识敏感数据识别方式是自动识别或者存在或者不存在 枚举值: AUTO:系统自动识别 EXIST:存在 NOT-EXIST:不存在]
data.hasref	bool	标识是否被引用
data.isdefault	bool	标识是否是默认配置
data.showname	string	标识默认配置的名字
data.pageNum	int32	敏感数据页面数

请求成功样例:

```
{
  "data": {
    "hasref": true,
    "businessType": "IP",
    "important": "COMMON",
    "uuid": "5FE8FDA10D8343CB9310E8B252E94C74",
    "ipRanges": [
      {
        "start": "0.0.0.0",
        "end": "255.255.255.255"
      }
    ],
    "isdefault": true,
    "name": "全部",
    "addressType": "IPv4",
    "description": "所有IP地址"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

2.1.3 批量获取网络对象

功能说明: 通过此接口, 可以通过名字获取一批网络对象

1) API请求数据

POST	https://{host}/api/batch/v1/namespaces/@namespace/ipgroups?_method=get
------	--

2) 输入参数

• Body参数

参数	类型	是否必选	描述
netobjs	object[]	是	网络对象组列表
netobjs.name	string	是	网络对象组名字 最小长度:1 最大长度:95

netobjs.businessType	string	否	网络对象组类型 枚举值:[IP:ip组类型 BUSINESS:业务组类型 USER:用户组类型 ADDRGROUP:地址组类型]
netobjs.description	string	否	描述字段 最小长度:0 最大长度:95
netobjs.addressType	string	否	地址类型 默认值:true 枚举值:[IPV4:IPv4类型 IPV6:IPv6类型]
netobjs.ipRanges	object[]	否	ip表项组
netobjs.ipRanges.start	string	是	ip范围的起始位置 格式:ip-address
netobjs.ipRanges.end	string	否	ip范围的终止位置 格式:ip-address
netobjs.ipRanges.bits	uint8	否	ip掩码简写
netobjs.ipRanges.mask	string	否	ip掩码 格式:ip-address
netobjs.important	string	否	标识核心业务/普通业务或者标识普通用户/ 核心用户 默认值:true 枚举值:[COMMON:普通用户或普通业务 CORE:核心用户或核心业务]
netobjs.refIpGroup	string[]	否	自定义识别范围引用网络对象列表
netobjs.dataStatus	string	否	标识敏感数据识别方式是自动识别或者存在 或者不存在 枚举值:[AUTO:系统自动识别 EXIST:存在 NOT-EXIST:不存在]

请求样例:

获取指定名字的网络对象列表, 查询条件: name为testA和testB

```
POST https://192.168.1.1/api/batch/v1/namespaces/public/ipgroups?_method=GET
[
  {
    "name": "testA"
  },
  {
    "name": "testB"
  }
]
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	网络对象组列表
data.uuid	string	资源唯一id
data.name	string	网络对象组名字 最小长度:1 最大长度:95
data.businessType	string	网络对象组类型 枚举值:[IP:ip组类型 BUSINESS:业务组类型 USER:用户组类型 ADDRGROUP:地址组类型]
data.description	string	描述字段 最小长度:0 最大长度:95
data.addressType	string	地址类型 枚举值:[IPV4:IPv4类型 IPV6:IPv6类型]
data.ipRanges	object[]	ip表项组
data.ipRanges.start	string	ip范围的起始位置 格式:ip-address
data.ipRanges.end	string	ip范围的终止位置 格式:ip-address
data.ipRanges.bits	uint8	ip掩码简写
data.ipRanges.mask	string	ip掩码 格式:ip-address
data.important	string	标识核心业务/普通业务或者标识普通用户/核心用户 枚举值:[COMMON:普通用户或普通业务 CORE:核心用户或核心业务]
data.refIpGroup	string[]	自定义识别范围引用网络对象列表
data.dataStatus	string	标识敏感数据识别方式是自动识别或者存在或者不存在 枚举值:[AUTO:系统自动识别 EXIST:存在 NOT-EXIST:不存在]
data.hasref	bool	标识是否被引用

data.isdefault	bool	标识是否是默认配置
data.showname	string	标识默认配置的名字
data.pageNum	int32	敏感数据页面数

请求成功样例：

```
{
  "data": [
    {
      "hasref": false,
      "ipRanges": [
        {
          "start": "1.1.1.1"
        }
      ],
      "uuid": "613F7B9F57F54021940187173C46C664",
      "businessType": "IP",
      "addressType": "IPV4",
      "important": "COMMON",
      "name": "testA",
      "description": ""
    },
    {
      "hasref": false,
      "ipRanges": [
        {
          "start": "1.1.1.1"
        }
      ],
      "uuid": "918B2DD9DDFC48A9906159A90EE56ECB",
      "businessType": "IP",
      "addressType": "IPV4",
      "important": "COMMON",
      "name": "testB",
      "description": ""
    }
  ],
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

2.1.4 添加网络对象

功能说明：通过此接口，可以添加一个网络对象

1) API请求数据

POST

https://{host}/api/v1/namespaces/@namespace/ipgroups

2) 输入参数

• Body参数

参数	类型	是否必选	描述
netobj	object	是	网络对象
netobj.name	string	是	网络对象组名字 最小长度:1 最大长度:95

netobj.businessType	string	否	网络对象组类型 枚举值:[IP:ip组类型 BUSINESS:业务组类型 USER:用户组类型 ADDRGROUP:地址组类型]
netobj.description	string	否	描述字段 最小长度:0 最大长度:95
netobj.addressType	string	否	地址类型 默认值:true 枚举值:[IPV4:IPv4类型 IPV6:IPv6类型]
netobj.ipRanges	object[]	否	ip表项组
netobj.ipRanges.start	string	是	ip范围的起始位置 格式:ip-address
netobj.ipRanges.end	string	否	ip范围的终止位置 格式:ip-address
netobj.ipRanges.bits	uint8	否	ip掩码简写
netobj.ipRanges.mask	string	否	ip掩码 格式:ip-address
netobj.important	string	否	标识核心业务/普通业务或者标识普通用户/ 核心用户 默认值:true 枚举值:[COMMON:普通用户或普通业务 CORE:核心用户或核心业务]
netobj.refIpGroup	string[]	否	自定义识别范围引用网络对象列表
netobj.dataStatus	string	否	标识敏感数据识别方式是自动识别或者存在 或者不存在 枚举值:[AUTO:系统自动识别 EXIST:存在 NOT-EXIST:不存在]

请求样例:

新建一个网络对象, 类型为IP地址, 名字为testB, ip地址为1.1.1.1

```
POST https://192.168.1.1/api/v1/namespaces/public/ipgroups
{
  "ipRanges": [
    {
      "start": "1.1.1.1"
    }
  ],
  "name": "testB",
  "businessType": "IP"
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	网络对象
data.uuid	string	资源唯一id
data.name	string	网络对象组名字 最小长度:1 最大长度:95
data.businessType	string	网络对象组类型 枚举值:[IP:ip组类型 BUSINESS:业务组类型 USER:用户组类型 ADDRGROUP:地址组类型]
data.description	string	描述字段 最小长度:0 最大长度:95
data.addressType	string	地址类型 枚举值:[IPV4:IPv4类型 IPV6:IPv6类型]
data.ipRanges	object[]	ip表项组
data.ipRanges.start	string	ip范围的起始位置 格式:ip-address
data.ipRanges.end	string	ip范围的终止位置 格式:ip-address
data.ipRanges.bits	uint8	ip掩码简写
data.ipRanges.mask	string	ip掩码 格式:ip-address
data.important	string	标识核心业务/普通业务或者标识普通用户/核心用户 枚举值:[COMMON:普通用户或普通业务 CORE:核心用户或核心业务]
data.refIpGroup	string[]	自定义识别范围引用网络对象列表
data.dataStatus	string	标识敏感数据识别方式是自动识别或者存在或者不存在 枚举值:[AUTO:系统自动识别 EXIST:存在 NOT-EXIST:不存在]
data.hasref	bool	标识是否被引用

data.isdefault	bool	标识是否是默认配置
data.showname	string	标识默认配置的名字
data.pageNum	int32	敏感数据页面数

请求成功样例：

```
{
  "data": {
    "ipRanges": [
      {
        "start": "1.1.1.1"
      }
    ],
    "uuid": "918B2DD9DDFC48A9906159A90EE56ECB",
    "businessType": "IP",
    "addressType": "IPv4",
    "important": "COMMON",
    "name": "testB",
    "description": ""
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

2.1.5 删除网络对象

功能说明：通过此接口，可以删除一个网络对象

1) API请求数据

DELETE https://{host}/api/v1/namespaces/@namespace/ipgroups/@name

2) 输入参数

请求样例：

删除指定名字为testB的网络对象
DELETE https://192.168.1.1/api/v1/namespaces/public/ipgroups/testB

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	网络对象
data.uuid	string	资源唯一id
data.name	string	网络对象组名字 最小长度:1 最大长度:95

data.businessType	string	网络对象组类型 枚举值:[IP:ip组类型 BUSINESS:业务组类型 USER:用户组类型 ADDRGROUP:地址组类型]
data.description	string	描述字段 最小长度:0 最大长度:95
data.addressType	string	地址类型 枚举值:[IPV4:IPv4类型 IPV6:IPv6类型]
data.ipRanges	object[]	ip表项组
data.ipRanges.start	string	ip范围的起始位置 格式:ip-address
data.ipRanges.end	string	ip范围的终止位置 格式:ip-address
data.ipRanges.bits	uint8	ip掩码简写
data.ipRanges.mask	string	ip掩码 格式:ip-address
data.important	string	标识核心业务/普通业务或者标识普通用户/核心用户 枚举值:[COMMON:普通用户或普通业务 CORE:核心用户或核心业务]
data.refIpGroup	string[]	自定义识别范围引用网络对象列表
data.dataStatus	string	标识敏感数据识别方式是自动识别或者存在或者不存在 枚举值:[AUTO:系统自动识别 EXIST:存在 NOT-EXIST:不存在]
data.hasref	bool	标识是否被引用
data.isdefault	bool	标识是否是默认配置
data.showname	string	标识默认配置的名字
data.pageNum	int32	敏感数据页面数

请求成功样例:

```
{
  "data": {
    "ipRanges": [
      {
        "start": "2.2.2.2"
      }
    ],
    "uuid": "918B2DD9DDFC48A9906159A90EE56ECB",
    "businessType": "IP",
    "addressType": "IPV4",
    "important": "COMMON",
  }
}
```

```

    "name": "testB",
    "description": ""
  },
  "code": 0,
  "message": "成功"
}

```

请求失败样例:

```

{
  "data": "",
  "code": 22,
  "message": "无效参数"
}

```

2.1.6 批量删除ip组信息

功能说明: 通过此接口, 可以删除多个网络对象

1) API请求数据

POST

https://{host}/api/batch/v1/namespaces/@namespace/ipgroups?_method=delete

2) 输入参数

• Body参数

参数	类型	是否必选	描述
obj	object[]	是	网络对象组列表
obj.name	string	是	网络对象组名字 最小长度:1 最大长度:95
obj.businessType	string	否	网络对象组类型 枚举值:[IP:ip组类型 BUSINESS:业务组类型 USER:用户组类型 ADDRGROUP:地址组类型]
obj.description	string	否	描述字段 最小长度:0 最大长度:95
obj.addressType	string	否	地址类型 默认值:true 枚举值:[IPV4:IPv4类型 IPV6:IPv6类型]
obj.ipRanges	object[]	否	ip表项组
obj.ipRanges.start	string	是	ip范围的起始位置 格式:ip-address
obj.ipRanges.end	string	否	ip范围的终止位置 格式:ip-address
obj.ipRanges.bits	uint8	否	ip掩码简写

obj.ipRanges.mask	string	否	ip掩码 格式:ip-address
obj.important	string	否	标识核心业务/普通业务或者标识普通用户/ 核心用户 默认值:true 枚举值:[COMMON:普通用户或普通业务 CORE:核心用户或核心业务]
obj.refIpGroup	string[]	否	自定义识别范围引用网络对象列表
obj.dataStatus	string	否	标识敏感数据识别方式是自动识别或者存在 或者不存在 枚举值:[AUTO:系统自动识别 EXIST:存在 NOT-EXIST:不存在]

请求样例:

批量删除指定为testA和testB名字的网络对象

```
POST https://192.168.1.1/api/batch/v1/namespaces/public/ipgroups?_method=DELETE
[
  {
    "name": "testA"
  },
  {
    "name": "testB"
  }
]
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	网络对象组列表
data.uuid	string	资源唯一id
data.name	string	网络对象组名字 最小长度:1 最大长度:95
data.businessType	string	网络对象组类型 枚举值:[IP:ip组类型 BUSINESS:业务组类型 USER:用户组类型 ADDRGROUP:地址组类型]
data.description	string	描述字段 最小长度:0 最大长度:95

data.addressType	string	地址类型 枚举值:[IPV4:IPv4类型 IPV6:IPv6类型]
data.ipRanges	object[]	ip表项组
data.ipRanges.start	string	ip范围的起始位置 格式:ip-address
data.ipRanges.end	string	ip范围的终止位置 格式:ip-address
data.ipRanges.bits	uint8	ip掩码简写
data.ipRanges.mask	string	ip掩码 格式:ip-address
data.important	string	标识核心业务/普通业务或者标识普通用户/核心用户 枚举值:[COMMON:普通用户或普通业务 CORE:核心用户或核心业务]
data.refIpGroup	string[]	自定义识别范围引用网络对象列表
data.dataStatus	string	标识敏感数据识别方式是自动识别或者存在或者不存在 枚举值:[AUTO:系统自动识别 EXIST:存在 NOT-EXIST:不存在]
data.hasref	bool	标识是否被引用
data.isdefault	bool	标识是否是默认配置
data.showname	string	标识默认配置的名字
data.pageNum	int32	敏感数据页面数

请求成功样例:

```
{
  "data": [
    {
      "ipRanges": [
        {
          "start": "1.1.1.2"
        }
      ],
      "uuid": "613F7B9F57F54021940187173C46C664",
      "businessType": "IP",
      "addressType": "IPV4",
      "important": "COMMON",
      "name": "testA",
      "description": ""
    },
    {
      "ipRanges": [
        {
          "start": "1.1.1.2"
        }
      ],
      "uuid": "EC17574C884248EFB68CABAB05FA0270",
      "businessType": "IP",
      "addressType": "IPV4",
      "important": "COMMON",
      "name": "testB",
      "description": ""
    }
  ]
}
```



```

        "description": ""
    },
    "code": 0,
    "message": "成功"
}

```

请求失败样例:

```

{
  "data": "",
  "code": 22,
  "message": "无效参数"
}

```

2.1.7 修改网络对象

功能说明: 通过此接口, 可以通过名字修改对应的网络对象的属性

1) API请求数据

PATCH <https://{host}/api/v1/namespaces/@namespace/ipgroups/@name>

2) 输入参数

• URL参数

参数	类型	是否必选	描述
_arrayop	string	否	对象属性是数组的操作控制 枚举值: add:添加对应的数组元素 remove:删除对应的数组元素]

• Body参数

参数	类型	是否必选	描述
netobj	object	是	网络对象
netobj.name	string	是	网络对象组名字 最小长度:1 最大长度:95
netobj.businessType	string	否	网络对象组类型 枚举值: IP:ip组类型 BUSINESS:业务组类型 USER:用户组类型 ADDRGROUP:地址组类型]
netobj.description	string	否	描述字段 最小长度:0 最大长度:95
netobj.addressType	string	否	地址类型 默认值:true 枚举值: IPV4:IPv4类型 IPV6:IPv6类型]
netobj.ipRanges	object[]	否	ip表项组

netobj.ipRanges.start	string	是	ip范围的起始位置 格式:ip-address
netobj.ipRanges.end	string	否	ip范围的终止位置 格式:ip-address
netobj.ipRanges.bits	uint8	否	ip掩码简写
netobj.ipRanges.mask	string	否	ip掩码 格式:ip-address
netobj.important	string	否	标识核心业务/普通业务或者标识普通用户/ 核心用户 默认值:true 枚举值:[COMMON:普通用户或普通业务 CORE:核心用户或核心业务]
netobj.refIpGroup	string[]	否	自定义识别范围引用网络对象列表
netobj.dataStatus	string	否	标识敏感数据识别方式是自动识别或者存在 或者不存在 枚举值:[AUTO:系统自动识别 EXIST:存在 NOT-EXIST:不存在]

请求样例:

增量修改指定名字为testB的网络对象, 修改ip地址为: 2.2.2.2, 其他属性值不变

```
PATCH https://192.168.1.1/api/v1/namespaces/public/ipgroups/testB
{
  "ipRanges": [
    {
      "start": "2.2.2.2"
    }
  ]
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	网络对象
data.uuid	string	资源唯一id
data.name	string	网络对象组名字 最小长度:1 最大长度:95

data.businessType	string	网络对象组类型 枚举值:[IP:ip组类型 BUSINESS:业务组类型 USER:用户组类型 ADDRGROUP:地址组类型]
data.description	string	描述字段 最小长度:0 最大长度:95
data.addressType	string	地址类型 枚举值:[IPV4:IPv4类型 IPV6:IPv6类型]
data.ipRanges	object[]	ip表项组
data.ipRanges.start	string	ip范围的起始位置 格式:ip-address
data.ipRanges.end	string	ip范围的终止位置 格式:ip-address
data.ipRanges.bits	uint8	ip掩码简写
data.ipRanges.mask	string	ip掩码 格式:ip-address
data.important	string	标识核心业务/普通业务或者标识普通用户/核心用户 枚举值:[COMMON:普通用户或普通业务 CORE:核心用户或核心业务]
data.refIpGroup	string[]	自定义识别范围引用网络对象列表
data.dataStatus	string	标识敏感数据识别方式是自动识别或者存在或者不存在 枚举值:[AUTO:系统自动识别 EXIST:存在 NOT-EXIST:不存在]
data.hasref	bool	标识是否被引用
data.isdefault	bool	标识是否是默认配置
data.showname	string	标识默认配置的名字
data.pageNum	int32	敏感数据页面数

请求成功样例:

```
{
  "data": {
    "ipRanges": [
      {
        "start": "2.2.2.2"
      }
    ],
    "uuid": "918B2DD9DDFC48A9906159A90EE56ECB",
    "businessType": "IP",
    "addressType": "IPV4",
    "important": "COMMON",
  }
}
```

```

    "name": "testB",
    "description": ""
  },
  "code": 0,
  "message": "成功"
}

```

请求失败样例:

```

{
  "data": "",
  "code": 22,
  "message": "无效参数"
}

```

2.1.8 全量修改网络对象

功能说明: 通过此接口, 可以通过名字全量修改一个网络对象

1) API请求数据

PUT <https://{host}/api/v1/namespaces/@namespace/ipgroups/@name>

2) 输入参数

- Body参数

参数	类型	是否必选	描述
obj	object	是	网络对象
obj.name	string	是	网络对象组名字 最小长度:1 最大长度:95
obj.businessType	string	否	网络对象组类型 枚举值:[IP:ip组类型 BUSINESS:业务组类型 USER:用户组类型 ADDRGROUP:地址组类型]
obj.description	string	否	描述字段 最小长度:0 最大长度:95
obj.addressType	string	否	地址类型 默认值:true 枚举值:[IPV4:IPv4类型 IPV6:IPv6类型]
obj.ipRanges	object[]	否	ip表项组
obj.ipRanges.start	string	是	ip范围的起始位置 格式:ip-address
obj.ipRanges.end	string	否	ip范围的终止位置 格式:ip-address
obj.ipRanges.bits	uint8	否	ip掩码简写
obj.ipRanges.mask	string	否	ip掩码 格式:ip-address

obj.important	string	否	标识核心业务/普通业务或者标识普通用户/核心用户 默认值:true 枚举值:[COMMON:普通用户或普通业务 CORE:核心用户或核心业务]
obj.refIpGroup	string[]	否	自定义识别范围引用网络对象列表
obj.dataStatus	string	否	标识敏感数据识别方式是自动识别或者存在或者不存在 枚举值:[AUTO:系统自动识别 EXIST:存在 NOT-EXIST:不存在]

请求样例:

全量修改指定名字的网络对象, name为testA, 类型为IP地址, ip地址为1.1.1.2, 其他属性值使用默认值

```
PUT https://192.168.1.1/api/v1/namespaces/public/ipgroups/testA
{
  "ipRanges": [
    {
      "start": "1.1.1.2"
    }
  ],
  "name": "testA",
  "businessType": "IP"
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	网络对象
data.uuid	string	资源唯一id
data.name	string	网络对象组名字 最小长度:1 最大长度:95
data.businessType	string	网络对象组类型 枚举值:[IP:ip组类型 BUSINESS:业务组类型 USER:用户组类型 ADDRGROUP:地址组类型]
data.description	string	描述字段 最小长度:0 最大长度:95

data.addressType	string	地址类型 枚举值:[IPV4:IPv4类型 IPV6:IPv6类型]
data.ipRanges	object[]	ip表项组
data.ipRanges.start	string	ip范围的起始位置 格式:ip-address
data.ipRanges.end	string	ip范围的终止位置 格式:ip-address
data.ipRanges.bits	uint8	ip掩码简写
data.ipRanges.mask	string	ip掩码 格式:ip-address
data.important	string	标识核心业务/普通业务或者标识普通用户/核心用户 枚举值:[COMMON:普通用户或普通业务 CORE:核心用户或核心业务]
data.refIpGroup	string[]	自定义识别范围引用网络对象列表
data.dataStatus	string	标识敏感数据识别方式是自动识别或者存在或者不存在 枚举值:[AUTO:系统自动识别 EXIST:存在 NOT-EXIST:不存在]
data.hasref	bool	标识是否被引用
data.isdefault	bool	标识是否是默认配置
data.showname	string	标识默认配置的名字
data.pageNum	int32	敏感数据页面数

请求成功样例:

```
{
  "data": {
    "ipRanges": [
      {
        "start": "1.1.1.2"
      }
    ],
    "uuid": "613F7B9F57F54021940187173C46C664",
    "businessType": "IP",
    "addressType": "IPV4",
    "important": "COMMON",
    "name": "testA",
    "description": ""
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

2.2 服务器识别

2.2.1 服务器识别高级配置

功能说明：通过此接口，进行服务器识别高级配置

1) API请求数据

PATCH `https://{host}/api/v1/namespaces/@namespace/netidentify/advancedconfig`

2) 输入参数

- Body参数

参数	类型	是否必选	描述
obj	object	是	服务器识别高级配置
obj.autoidentifyEnable	bool	是	自动识别开关
obj.sensitiveDataEnable	bool	是	识别敏感数据业务开关
obj.customEnable	bool	是	自定义识别范围开关
obj.ipGroups	string[]	是	自定义识别范围引用网络对象列表

请求样例：

服务器识别高级配置

```
PATCH https://192.168.1.1/api/v1/namespaces/public/netidentify/advancedconfig
{
  "ipGroups": [],
  "customEnable": false,
  "sensitiveDataEnable": true,
  "autoidentifyEnable": true
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	服务器识别高级配置
data.autoidentifyEnable	bool	自动识别开关
data.sensitiveDataEnable	bool	识别敏感数据业务开关
data.customEnable	bool	自定义识别范围开关
data.ipGroups	string[]	自定义识别范围引用网络对象列表

请求成功样例：

```
{
  "data": {
    "ipGroups": [],
    "customEnable": false,
    "sensitiveDataEnable": true,
    "autoidentifyEnable": true
  },
  "code": 0,
}
```

```
    "message": "成功"
  }
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

2.2.2 获取服务器识别高级配置

功能说明：通过此接口，获取服务器识别高级配置

1) API请求数据

GET https://{host}/api/v1/namespaces/@namespace/netidentify/advancedconfig

2) 输入参数

请求样例：

查看服务器识别高级配置

GET https://192.168.1.1/api/v1/namespaces/public/netidentify/advancedconfig

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	服务器识别高级配置
data.autoidentifyEnable	bool	自动识别开关
data.sensitiveDataEnable	bool	识别敏感数据业务开关
data.customEnable	bool	自定义识别范围开关
data.ipGroups	string[]	自定义识别范围引用网络对象列表

请求成功样例：

```
{
  "data": {
    "ipGroups": [],
    "customEnable": false,
    "sensitiveDataEnable": true,
    "autoidentifyEnable": true
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```


2.2.3 获取服务器识别结果

功能说明：通过此接口，获取服务器识别结果

1) API请求数据

GET <https://{host}/api/v1/namespaces/@namespace/netidentify/identifyservices>

2) 输入参数

- URL参数

参数	类型	是否必选	描述
_start	int32	否	起始位置, 从 0 开始 最小值:0 最大值:2000000
_sortby	string	否	指定排序字段 最大长度:100
_search	string	否	模糊搜索关键字 最大长度:95
_select	string	否	选择字段, 字段用逗号分隔 最大长度:1000
businessFlag	uint32	否	服务器所属业务类型 枚举值:[0:全部业务 1:未配置业务 2:已配置业务]
_order	string	否	指定排序方式（正序/倒序） 默认值:true 枚举值:[asc:按升序排列 desc:按降序排列]
_length	int32	否	获取最大长度，限制最大值，如果数据不够可能返回小于 length 最小值:1 最大值:200 默认值:true
filterSecuDataPage	uint32	否	敏感数据业务类型 枚举值:[0:全部业务 1:敏感数据业务 2:非敏感数据业务]

请求样例：

查看服务器识别结果列表

GET <https://192.168.1.1/api/v1/namespaces/public/netidentify/identifyservices>

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	分页数据
data.totalItems	int32	总共多少项目
data.totalPages	int32	总共多少页
data.pageNumber	int32	当前页码, 从 1 开始
data.pageSize	int32	每页多大
data.itemsOffset	int32	当前条目偏移, 从 0 开始
data.itemLength	int32	数据列表长度
data.items	object[]	有效数据列表
data.items.ip	string	服务器ip 格式:ip-address
data.items.business	string	所属业务
data.items.important	string	标识核心业务/普通业务或者标识普通用户/核心用户 枚举值:[COMMON:普通用户或普通业务 CORE:核心用户或核心业务]
data.items.identifyMethod	string	发现方式 枚举值:[USERDEFINED:自定义服务器 AUTOFIND:自动识别服务器]
data.items.serviceType	uint32	服务类型
data.items.pageNum	uint32	敏感数据个数
data.items.updateTime	string	更新时间
data.items.svrTypeInfo	string	服务类型

请求成功样例:

```
{
  "data": [
    {
      "updateTime": "-",
      "business": "测试业务",
      "ip": "1.1.1.1",
      "svrTypeInfo": "",
      "identifyMethod": "USERDEFINED",
      "important": "COMMON",
      "pageNum": 0,
      "serviceType": 0
    }
  ],
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

2.2.4 获取忽略服务器列表

功能说明：通过此接口，获取忽略服务器列表

1) API请求数据

GET <https://{host}/api/v1/namespaces/@namespace/netidentify/ignoredserver>

2) 输入参数

- URL 参数

参数	类型	是否必选	描述
_start	int32	否	起始位置, 从 0 开始 最小值:0 最大值:2000000
_search	string	否	模糊搜索关键字 最大长度:95
_length	int32	否	获取最大长度, 限制最大值, 如果数据不够 可能返回小于 length 最小值:1 最大值:200 默认值:true

请求样例：

获取忽略服务器列表

GET <https://192.168.1.1/api/v1/namespaces/public/netidentify/ignoredserver>

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	分页数据
data.totalItems	int32	总共多少项目
data.totalPages	int32	总共多少页
data.pageNumber	int32	当前页码, 从 1 开始
data.pageSize	int32	每页多大
data.itemsOffset	int32	当前条目偏移, 从 0 开始
data.itemLength	int32	数据列表长度
data.items	object[]	有效数据列表
data.items.ip	string	服务器ip

请求成功样例：

```
{
  "data": {
    "itemsOffset": 0,
    "items": [
      {
        "ip": "192.168.110.1"
      },
      {
        "ip": "192.168.110.11"
      },
      {
        "ip": "192.168.110.111"
      }
    ],
    "totalItems": 3,
    "pageNumber": 1,
    "itemLength": 3,
    "totalPages": 1,
    "pageSize": 100
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

2.2.5 将服务器IP添加到忽略服务器列表

功能说明: 通过此接口, 将服务器IP添加到忽略服务器列表

1) API请求数据

POST https://{host}/api/v1/namespaces/@namespace/netidentify/ignoredserver

2) 输入参数

• Body参数

参数	类型	是否必选	描述
obj	object	是	忽略服务器
obj.ip	string	是	服务器ip

请求样例:

将服务器IP: 192.168.110.1添加到忽略服务器列表

```
POST https://192.168.1.1/api/v1/namespaces/public/netidentify/ignoredserver
{
  "ip": "192.168.110.1"
}
```

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	忽略服务器

data.ip	string	服务器ip
---------	--------	-------

请求成功样例:

```
{
  "data": {
    "ip": "192.168.110.1"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

2.2.6 将服务器IP批量添加到忽略服务器列表

功能说明: 通过此接口, 将服务器IP批量添加到忽略服务器列表

1) API请求数据

POST	https://{host}/api/batch/v1/namespaces/@namespace/netidentify/ignoredserver
------	---

2) 输入参数

- Body参数

参数	类型	是否必选	描述
obj	object[]	是	忽略服务器列表
obj.ip	string	是	服务器ip

请求样例:

将服务器IP: 192.168.110.11和192.168.110.111批量添加到忽略服务器列表

```
POST https://192.168.1.1/api/batch/v1/namespaces/public/netidentify/ignoredserver
[
  {
    "ip": "192.168.110.11"
  },
  {
    "ip": "192.168.110.111"
  }
]
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	忽略服务器列表
data.ip	string	服务器ip

请求成功样例:

```
{
  "data": [
    {
      "ip": "192.168.110.11"
    },
    {
      "ip": "192.168.110.111"
    }
  ],
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

2.2.7 删除一条忽略服务器数据

功能说明：通过此接口，删除一条忽略服务器数据

1) API请求数据

DELETE `https://{host}/api/v1/namespaces/@namespace/netidentify/ignoredserver/@ip`

2) 输入参数

请求样例：

删除一条忽略服务器数据，IP为192.168.110.1

DELETE `https://192.168.1.1/api/v1/namespaces/public/netidentify/ignoredserver/192.168.110.1`

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	忽略服务器
data.ip	string	服务器ip

请求成功样例：

```
{
  "data": {
    "ip": "192.168.110.1"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

2.2.8 批量删除忽略服务器数据

功能说明：通过此接口，批量删除忽略服务器数据

1) API请求数据

POST

https://{host}/api/batch/v1/namespaces/@namespace/netidentify/ignoredserver?_method=delete

2) 输入参数

- Body参数

参数	类型	是否必选	描述
obj	object[]	是	忽略服务器列表
obj.ip	string	是	服务器ip

请求样例：

批量删除忽略服务器数据，IP为192.168.110.11和192.168.110.111

```
[
  {
    "ip": "192.168.110.11"
  },
  {
    "ip": "192.168.110.111"
  }
]
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	忽略服务器列表
data.ip	string	服务器ip

请求成功样例：

```
{
  "data": [
    {
      "ip": "192.168.110.11"
    },
    {
      "ip": "192.168.110.111"
    }
  ],
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

2.2.9 获取指定服务器敏感数据详情

功能说明：通过此接口，获取指定服务器敏感数据详情

1) API请求数据

GET

```
https://{host}/api/v1/namespaces/@namespace/netidentify/sensitivepages/  
@ip
```

2) 输入参数

请求样例:

获取敏感数据详情，要获取的服务器IP为192.168.102.102

GET https://192.168.1.1/api/v1/namespaces/public/netidentify/sensitivepages/192.168.102.102

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	忽略页面列表
data.url	string	页面uri 最小长度:1 最大长度:4096
data.pageTitle	string	页面标题 最大长度:128
data.operation	int32	页面状态 0为正常 1为忽略 2为删除
data.tableHtml	string	页面详情
data.autold	int32	页面id

请求成功样例:

[illegible]

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

功能说明：通过此接口，获取网络对象敏感数据详情

```
POST https://{host}/api/v1/namespaces/@namespace/ipgroups/sensitivepages?_method=get
```

- Body参数

参数	类型	是否必选	描述
obj	object	是	IPv4地址数组
obj.iplist	string[]	是	IPv4列表

请求样例:

```
POST https://192.168.1.1/api/v1/namespaces/public/ipgroups/sensitivepages?_method=get
{
  "iplist": [
    "192.168.1.1"
  ]
}
```

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	忽略页面列表
data.url	string	页面uri 最小长度:1 最大长度:4096

data.pageTitle	string	页面标题 最大长度:128
data.operation	int32	页面状态 0为正常 1为忽略 2为删除
data.tableHtml	string	页面详情
data.autold	int32	页面id

请求成功样例:

```
{
  "message": "",
  "data": [
    {
      "pageTitle": "test",
      "operation": 0,
      "url": "www.sangfor.com",
      "tableHtml": "test",
      "autold": 0
    }
  ],
  "code": 0
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

2.2.11 获取全部忽略敏感数据详情

功能说明: 通过此接口, 获取全部忽略敏感数据详情

1) API请求数据

GET

https://{host}/api/v1/namespaces/@namespace/netidentify/ignoredpages

2) 输入参数

- URL 参数

参数	类型	是否必选	描述
_start	int32	否	起始位置, 从 0 开始 最小值:0 最大值:2000000
_length	int32	否	获取最大长度, 限制最大值, 如果数据不够可能返回小于 length 最小值:1 最大值:200 默认值:true

请求样例:

获取全部忽略敏感数据详情

GET https://192.168.1.1/api/v1/namespaces/public/netidentify/ignoredpages

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	分页数据
data.totalItems	int32	总共多少项目
data.totalPages	int32	总共多少页
data.pageNumber	int32	当前页码, 从 1 开始
data.pageSize	int32	每页多大
data.itemsOffset	int32	当前条目偏移, 从 0 开始
data.itemLength	int32	数据列表长度
data.items	object[]	有效数据列表
data.items.uri	string	页面uri 最小长度:1 最大长度:4096
data.items.pageTitle	string	页面标题 最大长度:128
data.items.operation	int32	页面状态 0为正常 1为忽略 2为删除
data.items.tableHtml	string	页面详情
data.items.autoid	int32	页面id

请求成功样例:

[illegible]

[illegible]

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

2.2.12 批量忽略敏感数据详情

功能说明：通过此接口，批量忽略敏感数据详情

1) API请求数据

PATCH

https://{host}/api/batch/v1/namespaces/@namespace/netidentify/ignoredpa
ges

2) 输入参数

- Body参数

参数	类型	是否必选	描述
obj	object[]	是	忽略页面列表
obj.url	string	否	页面uri 最小长度:1 最大长度:4096
obj.pageTitle	string	否	页面标题 最大长度:128
obj.operation	int32	否	页面状态 0为正常 1为忽略 2为删除
obj.tableHtml	string	否	页面详情
obj.autold	int32	否	页面id

请求样例:

批量忽略敏感数据详情，对应的敏感数据autold为3和4，autold可以使用获取敏感数据详情API去查看

```
PATCH https://192.168.1.1/api/batch/v1/namespaces/public/netidentify/ignoredpages
[
  {
    "autold": 3
  },
  {
    "autold": 4
  }
]
```

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	忽略页面列表
data.url	string	页面uri 最小长度:1 最大长度:4096
data.pageTitle	string	页面标题 最大长度:128
data.operation	int32	页面状态 0为正常 1为忽略 2为删除
data.tableHtml	string	页面详情
data.autold	int32	页面id

请求成功样例:

[illegible]

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

功能说明：通过此接口，忽略敏感数据详情

DELETE https://{host}/api/v1/namespaces/@namespace/netidentify/ignoredpages/@autold

请求样例:

忽略单个敏感数据详情，对应的敏感数据autold为2，autold可以使用获取敏感数据详情API去查看

DELETE https://192.168.1.1/api/v1/namespaces/public/netidentify/ignoredpages/2

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	忽略页面

DELETE

```
https://{host}/api/v1/namespaces/@namespace/netidentify/deletedpages/@autold? method=delete
```

请求样例:

删除单个敏感数据详情，对应的敏感数据autold为2，autold可以使用获取敏感数据详情API去查看

DELETE <https://192.168.1.1/api/v1/namespaces/public/netidentify/deletedpages/2>

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	忽略页面
data.url	string	页面uri 最小长度:1 最大长度:4096
data.pageTitle	string	页面标题 最大长度:128
data.operation	int32	页面状态 0为正常 1为忽略 2为删除
data.tableHtml	string	页面详情
data.autold	int32	页面id

请求成功样例:

[illegible]

[illegible]

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

2.2.15 批量删除忽略敏感数据详情

功能说明：通过此接口，批量删除忽略敏感数据详情

1) API请求数据

POST

```
https://{host}/api/batch/v1/namespaces/@namespace/netidentify/ignoredpa  
ges? method=delete
```

2) 输入参数

- Body参数

参数	类型	是否必选	描述
obj	object[]	是	忽略页面列表
obj.url	string	否	页面uri 最小长度:1 最大长度:4096
obj.pageTitle	string	否	页面标题 最大长度:128
obj.operation	int32	否	页面状态 0为正常 1为忽略 2为删除
obj.tableHtml	string	否	页面详情
obj.autold	int32	否	页面id

请求样例:

批量删除被忽略敏感数据详情，对应的敏感数据autold为3和4，autold可以使用获取敏感数据详情API去查看

```
[
  {
    "autold": 3
  },
  {
    "autold": 4
  }
]
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码

请求成功样例：

第 66 页

```
}  
},  
"code": 0,  
"message": "成功"  
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

2.3 内容识别

2.3.1 应用识别

2.3.1.1 获取应用特征识别库版本日期有效期规则数量

功能说明：获取应用特征识别库版本日期有效期规则数量

1) API请求数据

```
GET https://{host}/api/v1/namespaces/@namespace/appidentifylibdate
```

2) 输入参数

请求样例:

获取应用特征识别库版本日期,有效期,规则数量

GET <https://192.168.1.1/api/v1/namespaces/public/appidentifvlibdate>

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	前端显示应用特征识别库的有效日期
data.date	string	应用特征识别库版本日期

data.expire	string	应用特征识别库版本过期日期 最小长度:2 最大长度:128
data.rules	string	应用规则总数

请求成功样例:

```
{
  "data": {
    "date": "2020-05-09 11:55:55",
    "expire": "2020-12-31",
    "rules": "10386"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

2.3.1.2 获取应用名称或应用分类列表

功能说明: 获取应用名称或应用分类列表

1) API请求数据

GET <https://{host}/api/v1/namespaces/@namespace/appidentifys>

2) 输入参数

• URL 参数

参数	类型	是否必选	描述
page	bool	否	是否分页
apppath	string	否	应用路径名,如应用自定义_a1,所属类型是自定义_t1,appPath就是自定义_t1/自定义_a1 最小长度:1 最大长度:287
_start	int32	否	起始位置, 从 0 开始 最小值:0 最大值:2000000

libtype	string	否	应用所属库类型, 枚举型。可选参数[FEATURE,INTELL,CUSTOM,PREDEF,ALL], FEATURE:特征库,INTELL:智能库,CUSTOM:自定义库,PREDEF:特征库与智能库,ALL:所有应用,包涵‘全部’与‘其他’两个大类,不设置该参数表示所有,不包涵‘全部’与‘其他’两个大类 枚举值:[FEATURE:特征库 INTELL:智能库 CUSTOM:自定义库 PREDEF:预定义应用 ALL:全部应用]
_order	string	否	指定排序方式(正序/倒序) 默认值:true 枚举值:[asc:按升序排列 desc:按降序排列]
_sortby	string	否	指定排序字段 最大长度:100
filter	string	否	应用名称或者描述的过滤条件,模糊搜索 最小长度:0 最大长度:95
_length	int32	否	获取最大长度, 限制最大值, 如果数据不够可能返回小于 length 最小值:1 最大值:200 默认值:true
__nameprefix	string	否	应用路径名前缀匹配 最小长度:1 最大长度:287
_select	string	否	选择字段, 字段用逗号分隔 最大长度:1000
isenable	int32	否	启禁用状态,整形。可选值[0,1,2,3],0:禁用,1:启用,2:所有,3:部分启用,不设置该参数表示所有状态
_search	string	否	模糊搜索关键字 最大长度:95
apptype	string	否	过滤应用类型,枚举型。可选值[APP,BIGGROUP,GROUP,MIDGROUP],APP:应用,BIGGROUP:大类,GROUP:大类和中间类,MIDGROUP:中间类,不设置该参数表示所有 枚举值:[APP:获取应用 BIGGROUP:获取大类分组 GROUP:获取所有中间类分组 MIDGROUP:获取所有分组]

filterincomplete	bool	否	过滤不完整的自定义应用
------------------	------	---	-------------

请求样例:

获取应用名称或应用分类列表,libtype=FEATURE 表示获取特征库,不填写表示获取所有库类型的数据;apptype=APP 表示获取应用,不填写表示获取所有应用类型的数据,apppath=访问网站 表示获取访问网站类别下面所有的应用,不填写表示不做类型过滤,起始位置为0,默认获取200条,page=true 表示进行分页,不填写表示返回的数据不进行分页

GET https://192.168.1.1/api/v1/namespaces/public/appidentifys?libtype=FEATURE&apptype=APP&apppath=访问网站&_start=0&_length=1&page=true

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	分页数据
data.totalItems	uint32	总条数
data.totalPages	uint32	总页数
data.pageNumber	uint32	当前第多少页
data.pageSize	uint32	每一页的条数
data.itemsOffset	uint32	当前偏移量
data.itemLength	uint32	当前返回条数
data.items	object[]	返回的数据列表
data.items.uuid	string	唯一标识id
data.items.name	string	应用名称 最小长度:1 最大长度:95
data.items.description	string	应用描述 最小长度:0 最大长度:1024
data.items.objType	string	应用类型, 枚举型。 枚举值: TYPE_OBJ:应用大类对象 SUB_TYPE_OBJ:应用中间类对象 APP_OBJ:应用大类对象]
data.items.libType	string	应用所属库类型, 枚举型。可选参数[FEATURE,INTELL,CUSTOM,FEATURE-INTELL,RESERVE], FEATURE:特征库,INTELL:智能库,CUSTOM:自定义库, FEATURE-INTELL:同时属于特征库与智能库,RESERVE:系统预留 枚举值: FEATURE:特征库 INTELL:智能库 CUSTOM:自定义库 FEATURE-INTELL:既属于特征库,又属于智能库 RESERVE:系统保留(全部,其他应用)]

data.items.appPath	string	应用路径名,如应用自定义_a1,所属类型是自定义_t1,appPath就是自定义_t1/自定义_a1 最小长度:1 最大长度:287
data.items.userdefCategory	string	自定义应用所属分类
data.items.appNum	int32	一个应用分类下面具体的应用个数
data.items.appStatus	object	应用名称所需参数
data.items.appStatus.ruleNum	uint16	所包含的应用规则数量
data.items.appStatus.enableNum	uint16	已经启用的应用规则数量
data.items.hasref	bool	标识是否被引用

请求成功样例:

```
{
  "data": {
    "itemsOffset": 0,
    "items": [
      {
        "libType": "FEATURE",
        "appPath": "访问网站/HTTP_GET",
        "appStatus": {
          "enableNum": 66,
          "ruleNum": 66
        },
        "uuid": "00000000000000000000106000643473918",
        "objType": "APP_OBJ",
        "name": "HTTP_GET",
        "description": "HTTP协议的GET方法。"
      }
    ],
    "totalItems": 1,
    "pageNumber": 1,
    "itemLength": 1,
    "totalPages": 1,
    "pageSize": 1
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

2.3.1.3 获取单个应用名称或应用分类

功能说明: 获取单个应用名称或应用分类

1) API请求数据

GET

https://{host}/api/v1/namespaces/@namespace/appidentifys/@appPath

2) 输入参数

- URL 参数

参数	类型	是否必选	描述
----	----	------	----

appname	string	否	应用名称 最小长度:1 最大长度:95
---------	--------	---	---------------------------

请求样例:

获取DNS/DNS协议应用是详细信息, uri中名称如果有/, 用%2f表示/

GET https://192.168.1.1/api/v1/namespaces/public/appidentifys/DNS%2fDNS协议

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	应用对象
data.uuid	string	唯一标识id
data.name	string	应用名称 最小长度:1 最大长度:95
data.description	string	应用描述 最小长度:0 最大长度:1024
data.objType	string	应用类型, 枚举型。 枚举值: TYPE_OBJ:应用大类对象 SUB_TYPE_OBJ:应用中间类对象 APP_OBJ:应用大类对象]
data.libType	string	应用所属库类型, 枚举型。可选参数[FEATURE, INTELL, CUSTOM, FEATURE-INTELL, RESERVE], FEATURE:特征库, INTELL:智能库, CUSTOM:自定义库, FEATURE-INTELL:同时属于特征库与智能库, RESERVE:系统预留 枚举值: FEATURE:特征库 INTELL:智能库 CUSTOM:自定义库 FEATURE-INTELL:既属于特征库,又属于智能库 RESERVE:系统保留(全部,其他应用)]
data.appPath	string	应用路径名,如应用自定义_a1,所属类型是自定义_t1, appPath就是自定义_t1/自定义_a1 最小长度:1 最大长度:287
data.userdefCategory	string	自定义应用所属分类
data.appNum	int32	一个应用分类下面具体的应用个数
data.appStatus	object	应用名称所需参数

data.appStatus.ruleNum	uint16	所包含的应用规则数量
data.appStatus.enableNum	uint16	已经启用的应用规则数量
data.hasref	bool	标识是否被引用

请求成功样例：

```
{
  "data": {
    "libType": "FEATURE",
    "appPath": "DNS/DNS协议",
    "appStatus": {
      "enableNum": 1,
      "ruleNum": 1
    },
    "uuid": "00000000000000000106001468244927",
    "objType": "APP_OBJ",
    "name": "DNS协议",
    "description": "域名系统（Domain Name System，DNS）是因特网的一项服务，它作为将域名和IP地址相互映射的一个分布式数据库，能够使人更方便的访问互联网。DNS 使用TCP和UDP端口53。"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

2.3.1.4 删除单个自定义的应用

功能说明：删除单个自定义的应用

1) API请求数据

DELETE https://{host}/api/v1/namespaces/@namespace/appidentifys/@appPath

2) 输入参数

请求样例：

删除应用自定义_tt1/自定义_aa1，uri中名称如果有/，用%2f表示/

DELETE https://192.168.1.1/api/v1/namespaces/public/appidentifys/自定义_tt1%2f自定义_aa1

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	应用对象
data.uuid	string	唯一标识id
data.name	string	应用名称 最小长度:1 最大长度:95

data.description	string	应用描述 最小长度:0 最大长度:1024
data.objType	string	应用类型, 枚举型。 枚举值:[TYPE_OBJ:应用大类对象 SUB_TYPE_OBJ:应用中间类对象 APP_OBJ:应用大类对象]
data.libType	string	应用所属库类型, 枚举型。可选参数[FEATURE,INTELL,CUSTOM,FEATURE-INTELL,RESERVE], FEATURE:特征库,INTELL:智能库,CUSTOM:自定义库, FEATURE-INTELL:同时属于特征库与智能库,RESERVE:系统预留 枚举值:[FEATURE:特征库 INTELL:智能库 CUSTOM:自定义库 FEATURE-INTELL:既属于特征库,又属于智能库 RESERVE:系统保留(全部,其他应用)]
data.appPath	string	应用路径名,如应用自定义_a1,所属类型是自定义_t1,appPath就是自定义_t1/自定义_a1 最小长度:1 最大长度:287
data.userdefCategory	string	自定义应用所属分类
data.appNum	int32	一个应用分类下面具体的应用个数
data.appStatus	object	应用名称所需参数
data.appStatus.ruleNum	uint16	所包含的应用规则数量
data.appStatus.enableNum	uint16	已经启用的应用规则数量
data.hasref	bool	标识是否被引用

请求成功样例:

```
{
  "data": {
    "libType": "CUSTOM",
    "appPath": "自定义_tt1/自定义_aa1",
    "userdefCategory": "自定义_tt1",
    "appStatus": {
      "enableNum": 0,
      "ruleNum": 0
    },
    "uuid": "78997B6472B242FF9BFF397505D7F2C4",
    "objType": "APP_OBJ",
    "name": "自定义_aa1",
    "description": "新增自定义应用"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
```

```

    "code": 22,
    "message": "无效参数"
  }

```

2.3.1.5 批量删除自定义的应用

功能说明：批量删除自定义的应用,通过appPath字段删除

1) API请求数据

POST

https://{host}/api/batch/v1/namespaces/@namespace/appidentifys?_method=delete

2) 输入参数

• Body参数

参数	类型	是否必选	描述
objs	object[]	是	应用名称/分类对象数组
objs.name	string	是	应用名称 最小长度:1 最大长度:95
objs.description	string	否	应用描述 最小长度:0 最大长度:1024
objs.objType	string	是	应用类型, 枚举型。 枚举值:[TYPE_OBJ:应用大类对象 SUB_TYPE_OBJ:应用中间类对象 APP_OBJ:应用大类对象]
objs.libType	string	否	应用所属库类型, 枚举型。可选参数[FEATURE,INTELL,CUSTOM,FEATURE-INTELL,RESERVE], FEATURE:特征库,INTELL:智能库,CUSTOM:自定义库,FEATURE-INTELL:同时属于特征库与智能库,RESERVE:系统预留 枚举值:[FEATURE:特征库 INTELL:智能库 CUSTOM:自定义库 FEATURE-INTELL:既属于特征库,又属于智能库 RESERVE:系统保留(全部,其他应用)]
objs.appPath	string	是	应用路径名,如应用自定义_a1,所属类型是自定义_t1,appPath就是自定义_t1/自定义_a1 最小长度:1 最大长度:287
objs.userdefCategory	string	否	自定义应用所属分类
objs.appNum	int32	否	一个应用分类下面具体的应用个数
objs.appStatus	object	是	应用名称所需参数
objs.appStatus.ruleNum	uint16	否	所包含的应用规则数量

objs.appStatus.enableNum	uint16	否	已经启用的应用规则数量
objs.hasref	bool	否	标识是否被引用

请求样例:

批量删除自定义的应用自定义_tt1/自定义_aa1和自定义_tt1, body中的/不需要进行转义

```
POST https://192.168.1.1/api/batch/v1/namespaces/public/appidentifys?_method=DELETE
[
  {
    "appPath": "自定义_tt1/自定义_aa1"
  },
  {
    "appPath": "自定义_tt1"
  }
]
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	应用名称/分类对象数组
data.uuid	string	唯一标识id
data.name	string	应用名称 最小长度:1 最大长度:95
data.description	string	应用描述 最小长度:0 最大长度:1024
data.objType	string	应用类型, 枚举型。 枚举值: TYPE_OBJ:应用大类对象 SUB_TYPE_OBJ:应用中间类对象 APP_OBJ:应用大类对象]
data.libType	string	应用所属库类型, 枚举型。可选参数[FEATURE, INTELL, CUSTOM, FEATURE-INTELL, RESERVE], FEATURE:特征库, INTELL:智能库, CUSTOM:自定义库, FEATURE-INTELL:同时属于特征库与智能库, RESERVE:系统预留 枚举值: FEATURE:特征库 INTELL:智能库 CUSTOM:自定义库 FEATURE-INTELL:既属于特征库,又属于智能库 RESERVE:系统保留(全部,其他应用)]
data.appPath	string	应用路径名,如应用自定义_aa1,所属类型是自定义_t1, appPath就是自定义_t1/自定义_aa1 最小长度:1 最大长度:287

data.userdefCategory	string	自定义应用所属分类
data.appNum	int32	一个应用分类下面具体的应用个数
data.appStatus	object	应用名称所需参数
data.appStatus.ruleNum	uint16	所包含的应用规则数量
data.appStatus.enableNum	uint16	已经启用的应用规则数量
data.hasref	bool	标识是否被引用

请求成功样例：

```
{
  "data": [
    {
      "libType": "CUSTOM",
      "appPath": "自定义_tt1/自定义_aa1",
      "userdefCategory": "自定义_tt1",
      "appStatus": {
        "enableNum": 0,
        "ruleNum": 0
      },
      "uuid": "5D54F9BDEFC44C81BD662F6CAF4222CF",
      "objType": "APP_OBJ",
      "name": "自定义_aa1",
      "description": "新增自定义应用"
    },
    {
      "appPath": "自定义_tt1",
      "libType": "CUSTOM",
      "uuid": "46E30B45E69C4AE4ADCCA065CA892616",
      "objType": "TYPE_OBJ",
      "name": "自定义_tt1",
      "description": "新增自定义大类"
    }
  ],
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

2.3.1.6 新增自定义应用或应用分类

功能说明：新增自定义应用或应用分类

1) API请求数据

POST

https://{host}/api/v1/namespaces/@namespace/appidentifys

2) 输入参数

- Body参数

参数	类型	是否必选	描述
obj	object	是	应用对象
obj.name	string	是	应用名称 最小长度:1 最大长度:95

obj.description	string	否	应用描述 最小长度:0 最大长度:1024
obj.objType	string	是	应用类型, 枚举型。 枚举值: TYPE_OBJ:应用大类对象 SUB_TYPE_OBJ:应用中间类对象 APP_OBJ:应用大类对象]
obj.libType	string	否	应用所属库类型, 枚举型。可选参数[FEATURE,INTELL,CUSTOM,FEATURE-INTELL,RESERVE], FEATURE:特征库,INTELL:智能库,CUSTOM:自定义库,FEATURE-INTELL:同时属于特征库与智能库,RESERVE:系统预留 枚举值: FEATURE:特征库 INTELL:智能库 CUSTOM:自定义库 FEATURE-INTELL:既属于特征库,又属于智能库 RESERVE:系统保留(全部,其他应用)]
obj.appPath	string	是	应用路径名,如应用自定义_a1,所属类型是自定义_t1,appPath就是自定义_t1/自定义_a1 最小长度:1 最大长度:287
obj.userdefCategory	string	否	自定义应用所属分类
obj.appNum	int32	否	一个应用分类下面具体的应用个数
obj.appStatus	object	是	应用名称所需参数
obj.appStatus.ruleNum	uint16	否	所包含的应用规则数量
obj.appStatus.enableNum	uint16	否	已经启用的应用规则数量
obj.hasref	bool	否	标识是否被引用

请求样例:

新增自定义应用或应用分类, 可以不指定name字段, 但必须指定appPath字段, objType字段表示新增的是应用还是大类, TYPE_OBJ: 大类, APP_OBJ: 应用, body中的/不需要进行转义

```
POST https://192.168.1.1/api/v1/namespaces/public/appidentifys
{
  "objType": "APP_OBJ",
  "appPath": "自定义_t1/自定义_aa1",
  "description": "新增自定义应用"
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码

message	string	错误信息
data	object	应用对象
data.uuid	string	唯一标识id
data.name	string	应用名称 最小长度:1 最大长度:95
data.description	string	应用描述 最小长度:0 最大长度:1024
data.objType	string	应用类型, 枚举型。 枚举值:[TYPE_OBJ:应用大类对象 SUB_TYPE_OBJ:应用中间类对象 APP_OBJ:应用大类对象]
data.libType	string	应用所属库类型, 枚举型。可选参数[FEATURE,INTELL,CUSTOM,FEATURE-INTELL,RESERVE], FEATURE:特征库,INTELL:智能库,CUSTOM:自定义库, FEATURE-INTELL:同时属于特征库与智能库,RESERVE:系统预留 枚举值:[FEATURE:特征库 INTELL:智能库 CUSTOM:自定义库 FEATURE-INTELL:既属于特征库,又属于智能库 RESERVE:系统保留(全部,其他应用)]
data.appPath	string	应用路径名,如应用自定义_a1,所属类型是自定义_t1,appPath就是自定义_t1/自定义_a1 最小长度:1 最大长度:287
data.userdefCategory	string	自定义应用所属分类
data.appNum	int32	一个应用分类下面具体的应用个数
data.appStatus	object	应用名称所需参数
data.appStatus.ruleNum	uint16	所包含的应用规则数量
data.appStatus.enableNum	uint16	已经启用的应用规则数量
data.hasref	bool	标识是否被引用

请求成功样例:

```
{
  "data": {
    "appPath": "自定义_t1/自定义_aa1",
    "libType": "CUSTOM",
    "uuid": "46E30B45E69C4AE4ADCCA065CA892616",
    "objType": "APP_OBJ",
    "name": "自定义_aa1",
    "description": "新增自定义应用"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

2.3.1.7 修改应用或应用分类

功能说明：修改应用或应用分类

1) API请求数据

PATCH <https://{host}/api/v1/namespaces/@namespace/appidentifys/@appPath>

2) 输入参数

- Body参数

参数	类型	是否必选	描述
obj	object	是	应用对象
obj.name	string	是	应用名称 最小长度:1 最大长度:95
obj.description	string	否	应用描述 最小长度:0 最大长度:1024
obj.objType	string	是	应用类型, 枚举型。 枚举值:[TYPE_OBJ:应用大类对象 SUB_TYPE_OBJ:应用中间类对象 APP_OBJ:应用大类对象]
obj.libType	string	否	应用所属库类型, 枚举型。可选参数[FEATURE,INTELL,CUSTOM,FEATURE-INTELL,RESERVE], FEATURE:特征库,INTELL:智能库,CUSTOM:自定义库,FEATURE-INTELL:同时属于特征库与智能库,RESERVE:系统预留 枚举值:[FEATURE:特征库 INTELL:智能库 CUSTOM:自定义库 FEATURE-INTELL:既属于特征库,又属于智能库 RESERVE:系统保留(全部,其他应用)]
obj.appPath	string	是	应用路径名,如应用自定义_a1,所属类型是自定义_t1,appPath就是自定义_t1/自定义_a1 最小长度:1 最大长度:287
obj.userdefCategory	string	否	自定义应用所属分类
obj.appNum	int32	否	一个应用分类下面具体的应用个数

obj.appStatus	object	是	应用名称所需参数
obj.appStatus.ruleNum	uint16	否	所包含的应用规则数量
obj.appStatus.enableNum	uint16	否	已经启用的应用规则数量
obj.hasref	bool	否	标识是否被引用

请求样例:

修改应用自定义_tt1/自定义_aa1的描述字段, uri中名称如果有/, 用%2f表示/

```
PATCH https://192.168.1.1/api/v1/namespaces/public/appidentifys/自定义_tt1%2f自定义_aa1
{
  "description": "应用和应用分类只能修改描述修 其他字段不允许修改"
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	应用对象
data.uuid	string	唯一标识id
data.name	string	应用名称 最小长度:1 最大长度:95
data.description	string	应用描述 最小长度:0 最大长度:1024
data.objType	string	应用类型, 枚举型。 枚举值: TYPE_OBJ:应用大类对象 SUB_TYPE_OBJ:应用中间类对象 APP_OBJ:应用大类对象]
data.libType	string	应用所属库类型, 枚举型。可选参数[FEATURE, INTELL, CUSTOM, FEATURE-INTELL, RESERVE], FEATURE:特征库, INTELL:智能库, CUSTOM:自定义库, FEATURE-INTELL:同时属于特征库与智能库, RESERVE:系统预留 枚举值: FEATURE:特征库 INTELL:智能库 CUSTOM:自定义库 FEATURE-INTELL:既属于特征库,又属于智能库 RESERVE:系统保留(全部,其他应用)]
data.appPath	string	应用路径名,如应用自定义_a1,所属类型是自定义_t1, appPath就是自定义_t1/自定义_a1 最小长度:1 最大长度:287

data.userdefCategory	string	自定义应用所属分类
data.appNum	int32	一个应用分类下面具体的应用个数
data.appStatus	object	应用名称所需参数
data.appStatus.ruleNum	uint16	所包含的应用规则数量
data.appStatus.enableNum	uint16	已经启用的应用规则数量
data.hasref	bool	标识是否被引用

请求成功样例:

```
{
  "data": {
    "libType": "CUSTOM",
    "appPath": "自定义_t1/自定义_aa1",
    "userdefCategory": "自定义_tt1",
    "appStatus": {
      "enableNum": 0,
      "ruleNum": 0
    },
    "uuid": "AD2FA7C43DF94E6D8CA2E63B12C9904C",
    "objType": "APP_OBJ",
    "name": "自定义_aa1",
    "description": "应用和应用分类只能修改描述修 其他字段不允许修改"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

2.3.1.8 获取应用规则列表

功能说明: 获取应用规则列表

1) API请求数据

GET <https://{host}/api/v1/namespaces/@namespace/appidentifyrules>

2) 输入参数

• URL参数

参数	类型	是否必选	描述
_start	int32	否	起始位置, 从 0 开始 最小值:0 最大值:2000000
apppath	string	否	应用路径名,如应用自定义_a1,所属类型是自定义_t1,appPath就是自定义_t1/自定义_a1 最小长度:1 最大长度:287
_search	string	否	模糊搜索关键字 最大长度:95

_select	string	否	选择字段, 字段用逗号分隔 最大长度:1000
_order	string	否	指定排序方式 (正序/倒序) 默认值:true 枚举值:[asc:按升序排列 desc:按降序排列]
libtype	string	否	应用所属库类型, 枚举型。可选参数[FEATURE,INTELL,CUSTOM,PREDEF,ALL] , FEATURE:特征库,INTELL:智能库,CUSTOM:自定义库,PREDEF:特征库与智能库,ALL:所有应用,包涵‘全部’与‘其他’两个大类,不设置该参数表示所有,不包涵‘全部’与‘其他’两个大类 枚举值:[FEATURE:特征库 INTELL:智能库 CUSTOM:自定义库 PREDEF:预定义应用 ALL:全部应用]
_length	int32	否	获取最大长度, 限制最大值, 如果数据不够可能返回小于 length 最小值:1 最大值:200 默认值:true
_sortby	string	否	指定排序字段 最大长度:100

请求样例:

获取应用规则列表

GET https://192.168.1.1/api/v1/namespaces/public/appidentifyrules?libtype=FEATURE&apppath=DNS/mDNS协议

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	分页数据
data.totalItems	int32	总共多少项目
data.totalPages	int32	总共多少页
data.pageNumber	int32	当前页码, 从 1 开始
data.pageSize	int32	每页多大
data.itemsOffset	int32	当前条目偏移, 从 0 开始
data.itemLength	int32	数据列表长度

data.items	object[]	有效数据列表
data.items.uuid	string	唯一标识id
data.items.name	string	规则名称 最小长度:1 最大长度:95
data.items.description	string	规则描述 最小长度:0 最大长度:1024
data.items.enable	bool	启禁用
data.items.ruleOwner	string	规则库属性,枚举型。可选参数[FEATURE,INTELL,CUSTOM], FEATURE:特征应用规则,INTELL:智能应用规则,CUSTOM:自定义应用规则 枚举值: FEATURE:特征应用规则 INTELL:智能应用规则 CUSTOM:自定义应用规则]
data.items.intellRuleParam	object	应用智能识别规则的配置参数
data.items.intellRuleParam.sensitivity	string	灵敏度,枚举型。可选参数[LOWER,LOW,MID,HIGH], LOWER:极低, LOW:低, MID:中, HIGH:高 枚举值: LOWER:极低 LOW:低 MID:中 HIGH:高]
data.items.intellRuleParam.excludePort	object[]	排除扫描端口列表,格式:[{"start": 1}, {"start": 2}]
data.items.intellRuleParam.excludePort.start	int32	网络端口起始值 最小值:0 最大值:65535
data.items.intellRuleParam.excludePort.end	int32	网络端口结束值 最小值:0 最大值:65535
data.items.customRuleParam	object	自定义规则配置参数
data.items.customRuleParam.direction	string	数据包方向,枚举型。可选参数[BOTH,LAN-WAN,WAN-LAN], BOTH:双向, LAN-WAN:LAN到WAN, WAN-LAN:WAN到LAN 枚举值: BOTH:LAN<->WAN LAN-WAN:LAN->WAN WAN-LAN:WAN->LAN]
data.items.customRuleParam.protocolNum	uint8	协议号, 范围:1-255 最小值:1 最大值:255

data.items.customRuleParam.portRange	object[]	指定端口或范围
data.items.customRuleParam.portRange.start	int32	网络端口起始值 最小值:0 最大值:65535
data.items.customRuleParam.portRange.end	int32	网络端口结束值 最小值:0 最大值:65535
data.items.customRuleParam.ipRange	object[]	指定IP地址或范围
data.items.customRuleParam.ipRange.start	string	ip范围的起始位置
data.items.customRuleParam.ipRange.end	string	ip范围的终止位置
data.items.customRuleParam.ipRange.bits	uint8	ip掩码简写
data.items.customRuleParam.ipRange.mask	string	ip掩码
data.items.customRuleParam.domains	string[]	匹配目标域名
data.items.appPath	string	应用规则所属的具体应用路径,如应用a1,所属类型是t1, appPath 就是t1/a1 最小长度:1 最大长度:287
data.items.userdefApplication	string	自定义应用规则所属的具体应用名称

请求成功样例:

```
{
  "data": {
    "itemsOffset": 0,
    "items": [
      {
        "enable": true,
        "ruleOwner": "FEATURE",
        "appPath": "DNS/mdns协议",
        "uuid": "00000000000000000306000050575131",
        "name": "mDNS",
        "description": "组播DNS"
      }
    ],
    "totalItems": 1,
    "pageNumber": 1,
    "itemLength": 1,
    "totalPages": 1,
    "pageSize": 100
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

2.3.1.9 获取单个应用规则

功能说明：获取单个应用规则

1) API请求数据

GET `https://{host}/api/v1/namespaces/@namespace/appidentifyrules/@name`

2) 输入参数

请求样例：

获取单个应用规则

GET `https://192.168.1.1/api/v1/namespaces/public/appidentifyrules/mDNS`

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	应用识别规则对象
data.uuid	string	唯一标识id
data.name	string	规则名称 最小长度:1 最大长度:95
data.description	string	规则描述 最小长度:0 最大长度:1024
data.enable	bool	启禁用
data.ruleOwner	string	规则库属性,枚举型。可选参数[FEATURE,INTELL,CUSTOM], FEATURE:特征应用规则,INTELL:智能应用规则,CUSTOM:自定义应用规则 枚举值: FEATURE:特征应用规则 INTELL:智能应用规则 CUSTOM:自定义应用规则]
data.intellRuleParam	object	应用智能识别规则的配置参数
data.intellRuleParam.sensitivity	string	灵敏度,枚举型。可选参数[LOWER,LOW,MID,HIGH], LOWER:极低, LOW:低, MID:中, HIGH:高 枚举值: LOWER:极低 LOW:低 MID:中 HIGH:高]
data.intellRuleParam.excludePort	object[]	排除扫描端口列表,格式:[{"start": 1}, {"start": 2}]

data.intellRuleParam.excludePort.start	int32	网络端口起始值 最小值:0 最大值:65535
data.intellRuleParam.excludePort.end	int32	网络端口结束值 最小值:0 最大值:65535
data.customRuleParam	object	自定义规则配置参数
data.customRuleParam.direction	string	数据包方向,枚举型。可选参数[BOTH,LAN-WAN,WAN-LAN], BOTH:双向, LAN-WAN:LAN到WAN, WAN-LAN:WAN到LAN 枚举值: BOTH:LAN<->WAN LAN-WAN:LAN->WAN WAN-LAN:WAN->LAN]
data.customRuleParam.protocolNum	uint8	协议号, 范围:1-255 最小值:1 最大值:255
data.customRuleParam.portRange	object[]	指定端口或范围
data.customRuleParam.portRange.start	int32	网络端口起始值 最小值:0 最大值:65535
data.customRuleParam.portRange.end	int32	网络端口结束值 最小值:0 最大值:65535
data.customRuleParam.ipRange	object[]	指定IP地址或范围
data.customRuleParam.ipRange.start	string	ip范围的起始位置
data.customRuleParam.ipRange.end	string	ip范围的终止位置
data.customRuleParam.ipRange.bits	uint8	ip掩码简写
data.customRuleParam.ipRange.mask	string	ip掩码
data.customRuleParam.domains	string[]	匹配目标域名
data.appPath	string	应用规则所属的具体应用路径,如应用a1,所属类型是t1, appPath 就是t1/a1 最小长度:1 最大长度:287
data.userdefApplication	string	自定义应用规则所属的具体应用名称

请求成功样例:

```
{
  "data": {
    "enable": true,
    "ruleOwner": "FEATURE",
    "appPath": "DNS/mDNS协议",
  }
}
```


obj.intellRuleParam.excludePort	object[]	否	排除扫描端口列表,格式:[{"start": 1}, {"start": 2}]
obj.intellRuleParam.excludePort.start	int32	是	网络端口起始值 最小值:0 最大值:65535
obj.intellRuleParam.excludePort.end	int32	否	网络端口结束值 最小值:0 最大值:65535
obj.customRuleParam	object	是	自定义规则配置参数
obj.customRuleParam.direction	string	否	数据包方向,枚举型。可选参数[BOTH, LAN-WAN, WAN-LAN], BOTH:双向, LAN-WAN:LAN到WAN, WAN-LAN:WAN到LAN 默认值:true 枚举值:[BOTH:LAN<->WAN LAN-WAN:LAN->WAN WAN-LAN:WAN->LAN]
obj.customRuleParam.protocolNum	uint8	是	协议号, 范围:1-255 最小值:1 最大值:255
obj.customRuleParam.portRange	object[]	否	指定端口或范围
obj.customRuleParam.portRange.start	int32	是	网络端口起始值 最小值:0 最大值:65535
obj.customRuleParam.portRange.end	int32	否	网络端口结束值 最小值:0 最大值:65535
obj.customRuleParam.ipRange	object[]	否	指定IP地址或范围
obj.customRuleParam.ipRange.start	string	是	ip范围的起始位置
obj.customRuleParam.ipRange.end	string	否	ip范围的终止位置
obj.customRuleParam.ipRange.bits	uint8	否	ip掩码简写
obj.customRuleParam.ipRange.mask	string	否	ip掩码
obj.customRuleParam.domains	string[]	否	匹配目标域名
obj.appPath	string	是	应用规则所属的具体应用路径,如应用a1,所属类型是t1, appPath 就是t1/a1 最小长度:1 最大长度:287
obj.userdefApplication	string	是	自定义应用规则所属的具体应用名称

请求样例:

新增自定义应用规则ruletest，状态为启用，portRange（端口范围）可以不指定，不指定表示所有端口，ipRange（ip范围）可以不指定，不指定表示所有IP，domains（域名匹配）可以不指定，不指定表示不做域名匹配

POST https://192.168.1.1/api/v1/namespaces/public/appidentifyrules

```
{
  "enable": true,
  "customRuleParam": {
    "domains": [
      "www.aaa.com",
      "www.11111111.com"
    ],
    "portRange": [
      {
        "start": 1,
        "end": 5
      },
      {
        "start": 10
      }
    ],
    "protocolNum": 6,
    "direction": "BOTH",
    "ipRange": [
      {
        "start": "0.0.0.1",
        "end": "0.0.0.2"
      }
    ]
  },
  "name": "ruletest",
  "appPath": "自定义_tt1/自定义_aa1",
  "description": "新增自定义规则"
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	应用识别规则对象
data.uuid	string	唯一标识id
data.name	string	规则名称 最小长度:1 最大长度:95
data.description	string	规则描述 最小长度:0 最大长度:1024
data.enable	bool	启禁用
data.ruleOwner	string	规则库属性,枚举型。可选参数[FEATURE,INTELL,CUSTOM], FEATURE:特征应用规则,INTELL:智能应用规则,CUSTOM:自定义应用规则 枚举值: FEATURE:特征应用规则 INTELL:智能应用规则 CUSTOM:自定义应用规则]
data.intellRuleParam	object	应用智能识别规则的配置参数

data.intellRuleParam.sensitivity	string	灵敏度,枚举型。可选参数[LOWER,LOW,MID,HIGH], LOWER:极低, LOW:低, MID:中, HIGH:高 枚举值: LOWER:极低 LOW:低 MID:中 HIGH:高]
data.intellRuleParam.excludePort	object[]	排除扫描端口列表,格式:[{"start": 1}, {"start": 2}]
data.intellRuleParam.excludePort.start	int32	网络端口起始值 最小值:0 最大值:65535
data.intellRuleParam.excludePort.end	int32	网络端口结束值 最小值:0 最大值:65535
data.customRuleParam	object	自定义规则配置参数
data.customRuleParam.direction	string	数据包方向,枚举型。可选参数[BOTH,LAN-WAN,WAN-LAN], BOTH:双向, LAN-WAN:LAN到WAN, WAN-LAN:WAN到LAN 枚举值: BOTH:LAN<->WAN LAN-WAN:LAN->WAN WAN-LAN:WAN->LAN]
data.customRuleParam.protocolNum	uint8	协议号, 范围:1-255 最小值:1 最大值:255
data.customRuleParam.portRange	object[]	指定端口或范围
data.customRuleParam.portRange.start	int32	网络端口起始值 最小值:0 最大值:65535
data.customRuleParam.portRange.end	int32	网络端口结束值 最小值:0 最大值:65535
data.customRuleParam.ipRange	object[]	指定IP地址或范围
data.customRuleParam.ipRange.start	string	ip范围的起始位置
data.customRuleParam.ipRange.end	string	ip范围的终止位置
data.customRuleParam.ipRange.bits	uint8	ip掩码简写
data.customRuleParam.ipRange.mask	string	ip掩码
data.customRuleParam.domains	string[]	匹配目标域名

data.appPath	string	应用规则所属的具体应用路径,如应用a1,所属类型是t1, appPath 就是t1/a1 最小长度:1 最大长度:287
data.userdefApplication	string	自定义应用规则所属的具体应用名称

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

2.3.1.11 删除单个自定义应用规则

功能说明: 删除单个自定义应用规则

1) API请求数据

DELETE https://{host}/api/v1/namespaces/@namespace/appidentifyrules/@name

2) 输入参数

- URL 参数

参数	类型	是否必选	描述
deleteapp	bool	否	是否删除空应用

请求样例:

```
DELETE https://192.168.1.1/api/v1/namespaces/public/appidentifyrules/admin
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	应用识别规则对象
data.uuid	string	唯一标识id
data.name	string	规则名称 最小长度:1 最大长度:95
data.description	string	规则描述 最小长度:0 最大长度:1024
data.enable	bool	启禁用

data.ruleOwner	string	规则库属性,枚举型。可选参数[FEATURE,INTELL,CUSTOM], FEATURE:特征应用规则,INTELL:智能应用规则,CUSTOM:自定义应用规则 枚举值: FEATURE:特征应用规则 INTELL:智能应用规则 CUSTOM:自定义应用规则]
data.intellRuleParam	object	应用智能识别规则的配置参数
data.intellRuleParam.sensitivity	string	灵敏度,枚举型。可选参数[LOWER,LOW,MID,HIGH], LOWER:极低, LOW:低, MID:中, HIGH:高 枚举值: LOWER:极低 LOW:低 MID:中 HIGH:高]
data.intellRuleParam.excludePort	object[]	排除扫描端口列表,格式:[{"start": 1}, {"start": 2}]
data.intellRuleParam.excludePort.start	int32	网络端口起始值 最小值:0 最大值:65535
data.intellRuleParam.excludePort.end	int32	网络端口结束值 最小值:0 最大值:65535
data.customRuleParam	object	自定义规则配置参数
data.customRuleParam.direction	string	数据包方向,枚举型。可选参数[BOTH,LAN-WAN,WAN-LAN], BOTH:双向, LAN-WAN:LAN到WAN, WAN-LAN:WAN到LAN 枚举值: BOTH:LAN<->WAN LAN-WAN:LAN->WAN WAN-LAN:WAN->LAN]
data.customRuleParam.protocolNum	uint8	协议号, 范围:1-255 最小值:1 最大值:255
data.customRuleParam.portRange	object[]	指定端口或范围
data.customRuleParam.portRange.start	int32	网络端口起始值 最小值:0 最大值:65535
data.customRuleParam.portRange.end	int32	网络端口结束值 最小值:0 最大值:65535
data.customRuleParam.ipRange	object[]	指定IP地址或范围
data.customRuleParam.ipRange.start	string	ip范围的起始位置

data.customRuleParam .ipRange.end	string	ip范围的终止位置
data.customRuleParam .ipRange.bits	uint8	ip掩码简写
data.customRuleParam .ipRange.mask	string	ip掩码
data.customRuleParam .domains	string[]	匹配目标域名
data.appPath	string	应用规则所属的具体应用路径,如应用a1,所属类型是t1, appPath 就是t1/a1 最小长度:1 最大长度:287
data.userdefApplication	string	自定义应用规则所属的具体应用名称

请求成功样例:

```
{
  "message": "",
  "data": {
    "enable": true,
    "ruleOwner": "CUSTOM",
    "intellRuleParam": {
      "sensitivity": "MID",
      "excludePort": [
        {
          "start": 0
        }
      ],
      "appPath": "test",
      "userdefApplication": "test",
      "customRuleParam": {
        "domains": [
          "www.sangfor.com"
        ],
        "portRange": [
          {
            "start": 0
          }
        ],
        "protocolNum": 1,
        "ipRange": [
          {
            "bits": 0,
            "end": "192.168.1.1",
            "start": "192.168.1.1",
            "mask": "192.168.1.1"
          }
        ],
        "direction": "BOTH"
      },
      "uuid": "07150a520899461e956bd9f0a69ea82d",
      "name": "test",
      "description": ""
    },
    "code": 0
  }
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

2.3.1.12 批量删除自定义应用规则

功能说明：批量删除自定义应用规则

1) API请求数据

POST

https://{host}/api/batch/v1/namespaces/@namespace/appidentifyrules?_method=delete

2) 输入参数

• URL参数

参数	类型	是否必选	描述
deleteapp	bool	否	是否删除空应用

• Body参数

参数	类型	是否必选	描述
objs	object[]	是	应用规则数组
objs.name	string	是	规则名称 最小长度:1 最大长度:95
objs.description	string	否	规则描述 最小长度:0 最大长度:1024
objs.enable	bool	否	启禁用 默认值:true
objs.ruleOwner	string	否	规则库属性,枚举型。可选参数[FEATURE,INTELL,CUSTOM], FEATURE:特征应用规则,INTELL:智能应用规则,CUSTOM:自定义应用规则 枚举值:[FEATURE:特征应用规则 INTELL:智能应用规则 CUSTOM:自定义应用规则]
objs.intellRuleParam	object	是	应用智能识别规则的配置参数
objs.intellRuleParam.sensitivity	string	否	灵敏度,枚举型。可选参数[LOWER,LOW,MID,HIGH], LOWER:极低, LOW:低, MID:中, HIGH:高 默认值:true 枚举值:[LOWER:极低 LOW:低 MID:中 HIGH:高]
objs.intellRuleParam.excludePort	object[]	否	排除扫描端口列表,格式:[{"start": 1}, {"start": 2}]
objs.intellRuleParam.excludePort.start	int32	是	网络端口起始值 最小值:0 最大值:65535

objs.intellRuleParam.excludePort.end	int32	否	网络端口结束值 最小值:0 最大值:65535
objs.customRuleParam	object	是	自定义规则配置参数
objs.customRuleParam.direction	string	否	数据包方向,枚举型。可选参数[BOTH,LAN-WAN,WAN-LAN], BOTH:双向, LAN-WAN:LAN到WAN, WAN-LAN:WAN到LAN 默认值:true 枚举值:[BOTH:LAN<->WAN LAN-WAN:LAN->WAN WAN-LAN:WAN->LAN]
objs.customRuleParam.protocolNum	uint8	是	协议号, 范围:1-255 最小值:1 最大值:255
objs.customRuleParam.portRange	object[]	否	指定端口或范围
objs.customRuleParam.portRange.start	int32	是	网络端口起始值 最小值:0 最大值:65535
objs.customRuleParam.portRange.end	int32	否	网络端口结束值 最小值:0 最大值:65535
objs.customRuleParam.ipRange	object[]	否	指定IP地址或范围
objs.customRuleParam.ipRange.start	string	是	ip范围的起始位置
objs.customRuleParam.ipRange.end	string	否	ip范围的终止位置
objs.customRuleParam.ipRange.bits	uint8	否	ip掩码简写
objs.customRuleParam.ipRange.mask	string	否	ip掩码
objs.customRuleParam.domains	string[]	否	匹配目标域名
objs.appPath	string	是	应用规则所属的具体应用路径,如应用a1,所属类型是t1, appPath 就是t1/a1 最小长度:1 最大长度:287
objs.userdefApplication	string	是	自定义应用规则所属的具体应用名称

请求样例:

```
POST https://192.168.1.1/api/batch/v1/namespaces/public/appidentifyrules?_method=delete
[
  {
    "enable": true,
    "ruleOwner": "CUSTOM",
    "userdefApplication": "test",
    "name": "test",
```



```
    "customRuleParam": {
      "domains": [
        "www.sangfor.com"
      ],
      "portRange": [
        {
          "start": 0
        }
      ],
      "protocolNum": 1,
      "ipRange": [
        {
          "bits": 0,
          "end": "192.168.1.1",
          "start": "192.168.1.1",
          "mask": "192.168.1.1"
        }
      ],
      "direction": "BOTH"
    },
    "intellRuleParam": {
      "sensitivity": "MID",
      "excludePort": [
        {
          "start": 0
        }
      ]
    },
    "appPath": "test",
    "description": ""
  }
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	应用规则数组
data.uuid	string	唯一标识id
data.name	string	规则名称 最小长度:1 最大长度:95
data.description	string	规则描述 最小长度:0 最大长度:1024
data.enable	bool	启禁用
data.ruleOwner	string	规则库属性,枚举型。可选参数[FEATURE,INTELL,CUSTOM], FEATURE:特征应用规则,INTELL:智能应用规则,CUSTOM:自定义应用规则 枚举值: FEATURE:特征应用规则 INTELL:智能应用规则 CUSTOM:自定义应用规则]
data.intellRuleParam	object	应用智能识别规则的配置参数

data.intellRuleParam.sensitivity	string	灵敏度,枚举型。可选参数[LOWER,LOW,MID,HIGH], LOWER:极低, LOW:低, MID:中, HIGH:高 枚举值: LOWER:极低 LOW:低 MID:中 HIGH:高]
data.intellRuleParam.excludePort	object[]	排除扫描端口列表,格式:[{"start": 1}, {"start": 2}]
data.intellRuleParam.excludePort.start	int32	网络端口起始值 最小值:0 最大值:65535
data.intellRuleParam.excludePort.end	int32	网络端口结束值 最小值:0 最大值:65535
data.customRuleParam	object	自定义规则配置参数
data.customRuleParam.direction	string	数据包方向,枚举型。可选参数[BOTH,LAN-WAN,WAN-LAN], BOTH:双向, LAN-WAN:LAN到WAN, WAN-LAN:WAN到LAN 枚举值: BOTH:LAN<->WAN LAN-WAN:LAN->WAN WAN-LAN:WAN->LAN]
data.customRuleParam.protocolNum	uint8	协议号, 范围:1-255 最小值:1 最大值:255
data.customRuleParam.portRange	object[]	指定端口或范围
data.customRuleParam.portRange.start	int32	网络端口起始值 最小值:0 最大值:65535
data.customRuleParam.portRange.end	int32	网络端口结束值 最小值:0 最大值:65535
data.customRuleParam.ipRange	object[]	指定IP地址或范围
data.customRuleParam.ipRange.start	string	ip范围的起始位置
data.customRuleParam.ipRange.end	string	ip范围的终止位置
data.customRuleParam.ipRange.bits	uint8	ip掩码简写
data.customRuleParam.ipRange.mask	string	ip掩码
data.customRuleParam.domains	string[]	匹配目标域名

data.appPath	string	应用规则所属的具体应用路径,如应用a1,所属类型是t1, appPath 就是t1/a1 最小长度:1 最大长度:287
data.userdefApplication	string	自定义应用规则所属的具体应用名称

请求成功样例:

```
{
  "message": "",
  "data": [
    {
      "enable": true,
      "ruleOwner": "CUSTOM",
      "intellRuleParam": {
        "sensitivity": "MID",
        "excludePort": [
          {
            "start": 0
          }
        ]
      },
      "appPath": "test",
      "userdefApplication": "test",
      "customRuleParam": {
        "domains": [
          "www.sangfor.com"
        ],
        "portRange": [
          {
            "start": 0
          }
        ],
        "protocolNum": 1,
        "ipRange": [
          {
            "bits": 0,
            "end": "192.168.1.1",
            "start": "192.168.1.1",
            "mask": "192.168.1.1"
          }
        ],
        "direction": "BOTH"
      },
      "uuid": "07150a520899461e956bd9f0a69ea82d",
      "name": "test",
      "description": ""
    }
  ],
  "code": 0
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

2.3.1.13 修改应用规则

功能说明: 修改应用规则

1) API请求数据

PATCH https://{host}/api/v1/namespaces/@namespace/appidentifyrules/@name

2) 输入参数

• URL参数

参数	类型	是否必选	描述
_arrayop	string	否	对象属性是数组的操作控制 枚举值:[add:添加对应的数组元素 remove:删除对应的数组元素]
deleteapp	bool	否	是否删除空应用

• Body参数

参数	类型	是否必选	描述
obj	object	是	应用识别规则对象
obj.name	string	是	规则名称 最小长度:1 最大长度:95
obj.description	string	否	规则描述 最小长度:0 最大长度:1024
obj.enable	bool	否	启禁用 默认值:true
obj.ruleOwner	string	否	规则库属性,枚举型。可选参数[FEATURE,INTELL,CUSTOM], FEATURE:特征应用规则,INTELL:智能应用规则,CUSTOM:自定义应用规则 枚举值:[FEATURE:特征应用规则 INTELL:智能应用规则 CUSTOM:自定义应用规则]
obj.intellRuleParam	object	是	应用智能识别规则的配置参数
obj.intellRuleParam.sensitivity	string	否	灵敏度,枚举型。可选参数[LOWER,LOW,MID,HIGH], LOWER:极低, LOW:低, MID:中, HIGH:高 默认值:true 枚举值:[LOWER:极低 LOW:低 MID:中 HIGH:高]
obj.intellRuleParam.excludePort	object[]	否	排除扫描端口列表,格式:[{"start": 1}, {"start": 2}]
obj.intellRuleParam.excludePort.start	int32	是	网络端口起始值 最小值:0 最大值:65535

obj.intellRuleParam.excludePort.end	int32	否	网络端口结束值 最小值:0 最大值:65535
obj.customRuleParam	object	是	自定义规则配置参数
obj.customRuleParam.direction	string	否	数据包方向,枚举型。可选参数[BOTH,LAN-WAN,WAN-LAN], BOTH:双向, LAN-WAN:LAN到WAN, WAN-LAN:WAN到LAN 默认值:true 枚举值:[BOTH:LAN<->WAN LAN-WAN:LAN->WAN WAN-LAN:WAN->LAN]
obj.customRuleParam.protocolNum	uint8	是	协议号, 范围:1-255 最小值:1 最大值:255
obj.customRuleParam.portRange	object[]	否	指定端口或范围
obj.customRuleParam.portRange.start	int32	是	网络端口起始值 最小值:0 最大值:65535
obj.customRuleParam.portRange.end	int32	否	网络端口结束值 最小值:0 最大值:65535
obj.customRuleParam.ipRange	object[]	否	指定IP地址或范围
obj.customRuleParam.ipRange.start	string	是	ip范围的起始位置
obj.customRuleParam.ipRange.end	string	否	ip范围的终止位置
obj.customRuleParam.ipRange.bits	uint8	否	ip掩码简写
obj.customRuleParam.ipRange.mask	string	否	ip掩码
obj.customRuleParam.domains	string[]	否	匹配目标域名
obj.appPath	string	是	应用规则所属的具体应用路径,如应用a1,所属类型是t1, appPath 就是t1/a1 最小长度:1 最大长度:287
obj.userdefApplication	string	是	自定义应用规则所属的具体应用名称

请求样例:

```
PATCH https://192.168.1.1/api/v1/namespaces/public/appidentifyrules/admin
{
  "enable": true,
  "ruleOwner": "CUSTOM",
  "userdefApplication": "test",
  "name": "test",
  "customRuleParam": {
```

```
{
  "domains": [
    "www.sangfor.com"
  ],
  "portRange": [
    {
      "start": 0
    }
  ],
  "protocolNum": 1,
  "ipRange": [
    {
      "bits": 0,
      "end": "192.168.1.1",
      "start": "192.168.1.1",
      "mask": "192.168.1.1"
    }
  ],
  "direction": "BOTH"
},
{
  "intellRuleParam": {
    "sensitivity": "MID",
    "excludePort": [
      {
        "start": 0
      }
    ]
  },
  "appPath": "test",
  "description": ""
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	应用识别规则对象
data.uuid	string	唯一标识id
data.name	string	规则名称 最小长度:1 最大长度:95
data.description	string	规则描述 最小长度:0 最大长度:1024
data.enable	bool	启禁用
data.ruleOwner	string	规则库属性,枚举型。可选参数[FEATURE,INTELL,CUSTOM], FEATURE:特征应用规则,INTELL:智能应用规则,CUSTOM:自定义应用规则 枚举值: FEATURE:特征应用规则 INTELL:智能应用规则 CUSTOM:自定义应用规则]
data.intellRuleParam	object	应用智能识别规则的配置参数

data.intellRuleParam.sensitivity	string	灵敏度,枚举型。可选参数[LOWER,LOW,MID,HIGH], LOWER:极低, LOW:低, MID:中, HIGH:高 枚举值: LOWER:极低 LOW:低 MID:中 HIGH:高]
data.intellRuleParam.excludePort	object[]	排除扫描端口列表,格式:[{"start": 1}, {"start": 2}]
data.intellRuleParam.excludePort.start	int32	网络端口起始值 最小值:0 最大值:65535
data.intellRuleParam.excludePort.end	int32	网络端口结束值 最小值:0 最大值:65535
data.customRuleParam	object	自定义规则配置参数
data.customRuleParam.direction	string	数据包方向,枚举型。可选参数[BOTH,LAN-WAN,WAN-LAN], BOTH:双向, LAN-WAN:LAN到WAN, WAN-LAN:WAN到LAN 枚举值: BOTH:LAN<->WAN LAN-WAN:LAN->WAN WAN-LAN:WAN->LAN]
data.customRuleParam.protocolNum	uint8	协议号, 范围:1-255 最小值:1 最大值:255
data.customRuleParam.portRange	object[]	指定端口或范围
data.customRuleParam.portRange.start	int32	网络端口起始值 最小值:0 最大值:65535
data.customRuleParam.portRange.end	int32	网络端口结束值 最小值:0 最大值:65535
data.customRuleParam.ipRange	object[]	指定IP地址或范围
data.customRuleParam.ipRange.start	string	ip范围的起始位置
data.customRuleParam.ipRange.end	string	ip范围的终止位置
data.customRuleParam.ipRange.bits	uint8	ip掩码简写
data.customRuleParam.ipRange.mask	string	ip掩码
data.customRuleParam.domains	string[]	匹配目标域名

data.appPath	string	应用规则所属的具体应用路径,如应用a1,所属类型是t1, appPath 就是t1/a1 最小长度:1 最大长度:287
data.userdefApplication	string	自定义应用规则所属的具体应用名称

请求成功样例:

```
{
  "message": "",
  "data": {
    "enable": true,
    "ruleOwner": "CUSTOM",
    "intellRuleParam": {
      "sensitivity": "MID",
      "excludePort": [
        {
          "start": 0
        }
      ]
    },
    "appPath": "test",
    "userdefApplication": "test",
    "customRuleParam": {
      "domains": [
        "www.sangfor.com"
      ],
      "portRange": [
        {
          "start": 0
        }
      ],
      "protocolNum": 1,
      "ipRange": [
        {
          "bits": 0,
          "end": "192.168.1.1",
          "start": "192.168.1.1",
          "mask": "192.168.1.1"
        }
      ],
      "direction": "BOTH"
    },
    "uuid": "07150a520899461e956bd9f0a69ea82d",
    "name": "test",
    "description": ""
  },
  "code": 0
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

2.3.1.14 批量启禁用应用规则(通过规则名称启禁用)

功能说明: 批量启禁用应用规则(通过规则名称启禁用)

1) API请求数据

PATCH <https://{host}/api/batch/v1/namespaces/@namespace/appidentifrules>

2) 输入参数

- Body参数

参数	类型	是否必选	描述
objs	object[]	是	用户启禁用应用规则列表
objs.name	string	是	规则名称 最小长度:1 最大长度:95
objs.enable	bool	否	启禁用 默认值:true

请求样例:

批量启禁用应用规则(通过规则名称启禁用), name为规则名称, enable为启用或者禁用

```
PATCH https://192.168.1.1/api/batch/v1/namespaces/public/appidentifyrules
[
  {
    "name": "LLMNR[1]",
    "enable": false
  },
  {
    "name": "mDNS",
    "enable": false
  },
  {
    "name": "DNS",
    "enable": false
  }
]
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	应用规则数组
data.uuid	string	唯一标识id
data.name	string	规则名称 最小长度:1 最大长度:95
data.description	string	规则描述 最小长度:0 最大长度:1024
data.enable	bool	启禁用
data.ruleOwner	string	规则库属性,枚举型。可选参数[FEATURE,INTELL,CUSTOM], FEATURE:特征应用规则,INTELL:智能应用规则,CUSTOM:自定义应用规则 枚举值: FEATURE:特征应用规则 INTELL:智能应用规则 CUSTOM:自定义应用规则]

data.intellRuleParam	object	应用智能识别规则的配置参数
data.intellRuleParam.sensitivity	string	灵敏度,枚举型。可选参数[LOWER,LOW,MID,HIGH], LOWER:极低, LOW:低, MID:中, HIGH:高 枚举值: LOWER:极低 LOW:低 MID:中 HIGH:高]
data.intellRuleParam.excludePort	object[]	排除扫描端口列表,格式:[{"start": 1}, {"start": 2}]
data.intellRuleParam.excludePort.start	int32	网络端口起始值 最小值:0 最大值:65535
data.intellRuleParam.excludePort.end	int32	网络端口结束值 最小值:0 最大值:65535
data.customRuleParam	object	自定义规则配置参数
data.customRuleParam.direction	string	数据包方向,枚举型。可选参数[BOTH,LAN-WAN,WAN-LAN], BOTH:双向, LAN-WAN:LAN到WAN, WAN-LAN:WAN到LAN 枚举值: BOTH:LAN<->WAN LAN-WAN:LAN->WAN WAN-LAN:WAN->LAN]
data.customRuleParam.protocolNum	uint8	协议号, 范围:1-255 最小值:1 最大值:255
data.customRuleParam.portRange	object[]	指定端口或范围
data.customRuleParam.portRange.start	int32	网络端口起始值 最小值:0 最大值:65535
data.customRuleParam.portRange.end	int32	网络端口结束值 最小值:0 最大值:65535
data.customRuleParam.ipRange	object[]	指定IP地址或范围
data.customRuleParam.ipRange.start	string	ip范围的起始位置
data.customRuleParam.ipRange.end	string	ip范围的终止位置
data.customRuleParam.ipRange.bits	uint8	ip掩码简写
data.customRuleParam.ipRange.mask	string	ip掩码

data.customRuleParam.domains	string[]	匹配目标域名
data.appPath	string	应用规则所属的具体应用路径,如应用a1,所属类型是t1, appPath 就是t1/a1 最小长度:1 最大长度:287
data.userdefApplication	string	自定义应用规则所属的具体应用名称

请求成功样例:

```
{
  "data": [
    {
      "enable": false,
      "ruleOwner": "FEATURE",
      "appPath": "DNS/DNS协议",
      "uuid": "00000000000000000306000340434635",
      "name": "DNS",
      "description": "DNS协议"
    },
    {
      "enable": false,
      "ruleOwner": "FEATURE",
      "appPath": "DNS/mDNS协议",
      "uuid": "00000000000000000306000050575131",
      "name": "mDNS",
      "description": "组播DNS"
    },
    {
      "enable": false,
      "ruleOwner": "FEATURE",
      "appPath": "DNS/LLMNR协议",
      "uuid": "00000000000000000306000517260511",
      "name": "LLMNR[1]",
      "description": ""
    }
  ],
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

2.3.1.15 批量启禁用应用规则(通过应用名称启禁用)

功能说明: 批量启禁用应用规则(通过应用名称启禁用)

1) API请求数据

PATCH	https://{host}/api/batch/v1/namespaces/@namespace/appidentifyrulesenable
-------	--

2) 输入参数

- Body参数

参数	类型	是否必选	描述
objs	object	是	用户启禁用应用规则列表
objs.apps	object[]	是	应用名称

objs.apps.name	string	是	应用名称 最小长度:1 最大长度:287
objs.enable	bool	是	启禁用
objs.libtype	string	是	应用所属库类型, 枚举型。可选参数[FEATURE,INTELL,CUSTOM], FEATURE:特征库,INTELL:智能库,CUSTOM:自定义库 枚举值: FEATURE:特征库 INTELL:智能库 CUSTOM:自定义库]

请求样例:

批量启禁用应用规则(通过应用名称启禁用), name为应用名称, 需要填写appPath字段的数值

```
GET https://192.168.1.1/api/batch/v1/namespaces/public/appidentifyrulesenable
{
  "enable": false,
  "libtype": "FEATURE",
  "apps": [
    {
      "name": "DNS/mDNS协议"
    },
    {
      "name": "DNS/DNS协议"
    }
  ]
}
```

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	应用规则数组
data.uuid	string	唯一标识id
data.name	string	规则名称 最小长度:1 最大长度:95
data.description	string	规则描述 最小长度:0 最大长度:1024
data.enable	bool	启禁用
data.ruleOwner	string	规则库属性,枚举型。可选参数[FEATURE,INTELL,CUSTOM], FEATURE:特征应用规则,INTELL:智能应用规则,CUSTOM:自定义应用规则 枚举值: FEATURE:特征应用规则 INTELL:智能应用规则 CUSTOM:自定义应用规则]

data.intellRuleParam	object	应用智能识别规则的配置参数
data.intellRuleParam.sensitivity	string	灵敏度,枚举型。可选参数[LOWER,LOW,MID,HIGH], LOWER:极低, LOW:低, MID:中, HIGH:高 枚举值: LOWER:极低 LOW:低 MID:中 HIGH:高]
data.intellRuleParam.excludePort	object[]	排除扫描端口列表,格式:[{"start": 1}, {"start": 2}]
data.intellRuleParam.excludePort.start	int32	网络端口起始值 最小值:0 最大值:65535
data.intellRuleParam.excludePort.end	int32	网络端口结束值 最小值:0 最大值:65535
data.customRuleParam	object	自定义规则配置参数
data.customRuleParam.direction	string	数据包方向,枚举型。可选参数[BOTH,LAN-WAN,WAN-LAN], BOTH:双向, LAN-WAN:LAN到WAN, WAN-LAN:WAN到LAN 枚举值: BOTH:LAN<->WAN LAN-WAN:LAN->WAN WAN-LAN:WAN->LAN]
data.customRuleParam.protocolNum	uint8	协议号, 范围:1-255 最小值:1 最大值:255
data.customRuleParam.portRange	object[]	指定端口或范围
data.customRuleParam.portRange.start	int32	网络端口起始值 最小值:0 最大值:65535
data.customRuleParam.portRange.end	int32	网络端口结束值 最小值:0 最大值:65535
data.customRuleParam.ipRange	object[]	指定IP地址或范围
data.customRuleParam.ipRange.start	string	ip范围的起始位置
data.customRuleParam.ipRange.end	string	ip范围的终止位置
data.customRuleParam.ipRange.bits	uint8	ip掩码简写
data.customRuleParam.ipRange.mask	string	ip掩码

data.customRuleParam .domains	string[]	匹配目标域名
data.appPath	string	应用规则所属的具体应用路径,如应用a1,所属类型是t1, appPath 就是t1/a1 最小长度:1 最大长度:287
data.userdefApplication	string	自定义应用规则所属的具体应用名称

请求成功样例:

```
{
  "data": [
    {
      "enable": true,
      "ruleOwner": "FEATURE",
      "appPath": "DNS/DNS协议",
      "uuid": "00000000000000000306000340434635",
      "name": "DNS",
      "description": "DNS协议"
    },
    {
      "enable": false,
      "ruleOwner": "FEATURE",
      "appPath": "DNS/mDNS协议",
      "uuid": "00000000000000000306000050575131",
      "name": "mDNS",
      "description": "组播DNS"
    }
  ],
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

2.3.1.16 获取当前的自定义优先级

功能说明: 获取当前的自定义优先级

1) API请求数据

GET

https://{host}/api/v1/namespaces/@namespace/customruleconf

2) 输入参数

请求样例:

获取当前的自定义优先级

```
GET https://192.168.1.1/api/v1/namespaces/public/customruleconf
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	用户自定义规则优先

data.isUserFirst	bool	是否用户自定义优先
------------------	------	-----------

请求成功样例:

```
{
  "data": {
    "isUserFirst": false
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

2.3.1.17 修改自定义优先级

功能说明: 修改自定义优先级

1) API请求数据

PATCH https://{host}/api/v1/namespaces/@namespace/customruleconf

2) 输入参数

• Body参数

参数	类型	是否必选	描述
obj	object	是	用户自定义规则优先
obj.isUserFirst	bool	否	是否用户自定义优先

请求样例:

修改自定义优先级

```
PATCH https://192.168.1.1/api/v1/namespaces/public/customruleconf
{
  "isUserFirst": false
}
```

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	用户自定义规则优先
data.isUserFirst	bool	是否用户自定义优先

请求成功样例:

```
{
  "data": {
    "isUserFirst": false
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

2.4 URL分类库

2.4.1 获取URL分类库信息

功能说明: 获取URL分类库信息

1) API请求数据

GET https://{host}/api/v1/namespaces/@namespace/urllib

2) 输入参数

- URL参数

参数	类型	是否必选	描述
_start	int32	否	起始位置, 从 0 开始 最小值:0 最大值:2000000
_length	int32	否	获取最大长度, 限制最大值, 如果数据不够可能返回小于 length 最小值:1 最大值:200 默认值:true
_search	string	否	模糊搜索关键字 最大长度:95
_select	string	否	选择字段, 字段用逗号分隔 最大长度:1000

请求样例:

获取所有的url列表信息, 查询条件: 起始位置为0, 最多展示100条

GET https://192.168.1.1/api/v1/namespaces/public/urllib?_start=0&_length=100

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	分页数据
data.totalItems	int32	总共多少项目
data.totalPages	int32	总共多少页
data.pageNumber	int32	当前页码, 从 1 开始
data.pageSize	int32	每页多大

data.itemsOffset	int32	当前条目偏移, 从 0 开始
data.itemLength	int32	数据列表长度
data.items	object[]	有效数据列表
data.items.uuid	string	URL 唯一标识id
data.items.name	string	URL组名称 最小长度:1 最大长度:95
data.items.description	string	URL组描述
data.items.tag	string	分组标签 最大长度:1024
data.items.urlType	string	用于区分当前URL分组是内置还是用户自定类型 枚举值:[INSIDE_URL:内置URL类型 USER_URL:用户自定义URL类型]
data.items.isMalice	bool	恶意链接库标识
data.items.urlItems	string[]	用户URL列表
data.items.keywords	string[]	用户关键字列表

请求成功样例:

```
{
  "data": {
    "itemsOffset": 0,
    "items": [
      {
        "urlType": "INSIDE_URL",
        "urlItems": [],
        "keywords": [],
        "tag": "Web应用",
        "isMalice": false,
        "uuid": "0000000000000000000000000000a00f7df1f",
        "name": "微博",
        "description": "包括提供一种可以即时发布消息，类似博客系统的非正式的迷你型博客"
      }
    ],
    "totalItems": 1,
    "pageNumber": 1,
    "itemLength": 1,
    "totalPages": 1,
    "pageSize": 100
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

2.4.2 获取指定url分类信息

功能说明: 获取指定url分类信息

1) API请求数据

GET https://{host}/api/v1/namespaces/@namespace/urllib/@urlName

2) 输入参数

- URL 参数

参数	类型	是否必选	描述
_select	string	否	选择字段, 字段用逗号分隔 最大长度:1000

请求样例:

获取指定的url分类信息

GET https://192.168.1.1/api/v1/namespaces/public/urllib/微博

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	URL分类
data.uuid	string	URL 唯一标识id
data.name	string	URL组名称 最小长度:1 最大长度:95
data.description	string	URL组描述
data.tag	string	分组标签 最大长度:1024
data.urlType	string	用于区分当前URL分组是内置还是用户自定义类型 枚举值: INSIDE_URL:内置URL类型 USER_URL:用户自定义URL类型]
data.isMalice	bool	恶意链接库标识
data.urlItems	string[]	用户URL列表
data.keywords	string[]	用户关键字列表

请求成功样例:

```
{
  "data": {
    "urlType": "INSIDE_URL",
    "urlItems": [],
    "keywords": [],
    "tag": "Web应用",
    "isMalice": false,
    "uuid": "00000000000000000000000000000000a00f7df1f",
    "name": "微博",
    "description": "包括提供一种可以即时发布消息，类似博客系统的非正式的迷你型博客"
  },
  "code": 0,
  "message": "成功"
}
```

```
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

2.4.3 添加自定义用户url数据

功能说明: 添加自定义用户url数据

1) API请求数据

POST https://{host}/api/v1/namespaces/@namespace/urllib

2) 输入参数

- Body参数

参数	类型	是否必选	描述
obj	object	是	URL分类
obj.name	string	是	URL组名称 最小长度:1 最大长度:95
obj.description	string	否	URL组描述
obj.tag	string	否	分组标签 最大长度:1024
obj.urlType	string	否	用于区分当前URL分组是内置还是用户自定义类型 默认值:true 枚举值:[INSIDE_URL:内置URL类型 USER_URL:用户自定义URL类型]
obj.isMalice	bool	是	恶意链接库标识
obj.urlItems	string[]	否	用户URL列表
obj.keywords	string[]	否	用户关键字列表

请求样例:

新增url类型对象

```
POST https://192.168.1.1/api/v1/namespaces/public/urllib
{
  "urlItems": [
    "www.testurl.com"
  ],
  "keywords": [],
  "id": "",
  "name": "test",
  "description": "this is a url"
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	URL分类
data.uuid	string	URL 唯一标识id
data.name	string	URL组名称 最小长度:1 最大长度:95
data.description	string	URL组描述
data.tag	string	分组标签 最大长度:1024
data.urlType	string	用于区分当前URL分组是内置还是用户自定义类型 枚举值: INSIDE_URL:内置URL类型 USER_URL:用户自定义URL类型]
data.isMalice	bool	恶意链接库标识
data.urlItems	string[]	用户URL列表
data.keywords	string[]	用户关键字列表

请求成功样例:

```
{
  "data": {
    "urlType": "USER_URL",
    "urlItems": [
      "www.testurl.com"
    ],
    "keywords": [],
    "tag": "",
    "isMalice": false,
    "uuid": "1A26383E72A94E15A9B91BC39B449AF7",
    "name": "test",
    "description": "this is a url"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

2.4.4 修改url分组信息

功能说明: 修改url分组信息

1) API请求数据

PATCH https://{host}/api/v1/namespaces/@namespace/urllib/@name

2) 输入参数

• URL参数

参数	类型	是否必选	描述
_arrayop	string	否	对象属性是数组的操作控制 枚举值:[add:添加对应的数组元素 remove:删除对应的数组元素]

• Body参数

参数	类型	是否必选	描述
obj	object	是	URL分类
obj.name	string	是	URL组名称 最小长度:1 最大长度:95
obj.description	string	否	URL组描述
obj.tag	string	否	分组标签 最大长度:1024
obj.urlType	string	否	用于区分当前URL分组是内置还是用户自定义类型 默认值:true 枚举值:[INSIDE_URL:内置URL类型 USER_URL:用户自定义URL类型]
obj.isMalice	bool	是	恶意链接库标识
obj.urlItems	string[]	否	用户URL列表
obj.keywords	string[]	否	用户关键字列表

请求样例:

修改url类型对象, 修改名为test的url类型对象的描述为modify test, 其它不做修改

```
PATCH https://192.168.1.1/api/v1/namespaces/public/urllib/test
{
  "description": "modify test"
}
```

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	URL分类
data.uuid	string	URL 唯一标识id

data.name	string	URL组名称 最小长度:1 最大长度:95
data.description	string	URL组描述
data.tag	string	分组标签 最大长度:1024
data.urlType	string	用于区分当前URL分组是内置还是用户自定类型 枚举值:[INSIDE_URL:内置URL类型 USER_URL:用户自定义URL类型]
data.isMalice	bool	恶意链接库标识
data.urlItems	string[]	用户URL列表
data.keywords	string[]	用户关键字列表

请求成功样例:

```
{
  "data": {
    "urlType": "USER_URL",
    "urlItems": [
      "www.testurl.com"
    ],
    "keywords": [],
    "tag": "",
    "isMalice": false,
    "uuid": "1A26383E72A94E15A9B91BC39B449AF7",
    "name": "test",
    "description": "modify test"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

2.4.5 删除用户自定义url数据

功能说明: 删除用户自定义url数据

1) API请求数据

DELETE https://{host}/api/v1/namespaces/@namespace/urlib/@name

2) 输入参数

请求样例:

删除名为test的url类型对象

DELETE https://192.168.1.1/api/v1/namespaces/public/urlib/test

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	URL分类
data.uuid	string	URL 唯一标识id
data.name	string	URL组名称 最小长度:1 最大长度:95
data.description	string	URL组描述
data.tag	string	分组标签 最大长度:1024
data.urlType	string	用于区分当前URL分组是内置还是用户自定义类型 枚举值:[INSIDE_URL:内置URL类型 USER_URL:用户自定义URL类型]
data.isMalice	bool	恶意链接库标识
data.urlItems	string[]	用户URL列表
data.keywords	string[]	用户关键字列表

请求成功样例:

```
{
  "data": {
    "urlType": "USER_URL",
    "urlItems": [
      "www.testurl.com"
    ],
    "keywords": [],
    "tag": "",
    "isMalice": false,
    "uuid": "1A26383E72A94E15A9B91BC39B449AF7",
    "name": "test",
    "description": "modify test"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

2.4.6 批量删除URL信息

功能说明: 批量删除URL信息

1) API请求数据

POST

`https://{host}/api/batch/v1/namespaces/@namespace/urllib?_method=delete`

2) 输入参数

• Body参数

参数	类型	是否必选	描述
obj	object[]	是	URL规则列表
obj.name	string	是	URL组名称 最小长度:1 最大长度:95
obj.description	string	否	URL组描述
obj.tag	string	否	分组标签 最大长度:1024
obj.urlType	string	否	用于区分当前URL分组是内置还是用户自定义类型 默认值:true 枚举值:[INSIDE_URL:内置URL类型 USER_URL:用户自定义URL类型]
obj.isMalice	bool	是	恶意链接库标识
obj.urlItems	string[]	否	用户URL列表
obj.keywords	string[]	否	用户关键字列表

请求样例:

删除多个url类型对象, 示例删除名为test的url类型对象

POST `https://192.168.1.1/api/v1/namespaces/public/urllib?_method=DELETE`

```
[  
  {  
    "name": "test"  
  }  
]
```

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	URL规则列表
data.uuid	string	URL 唯一标识id
data.name	string	URL组名称 最小长度:1 最大长度:95
data.description	string	URL组描述

data.tag	string	分组标签 最大长度:1024
data.urlType	string	用于区分当前URL分组是内置还是用户自定类型 枚举值:[INSIDE_URL:内置URL类型 USER_URL:用户自定义URL类型]
data.isMalice	bool	恶意链接库标识
data.urlItems	string[]	用户URL列表
data.keywords	string[]	用户关键字列表

请求成功样例:

```
{
  "data": [
    {
      "urlType": "USER_URL",
      "urlItems": [
        "www.testurl.com"
      ],
      "keywords": [],
      "tag": "",
      "isMalice": false,
      "uuid": "1A26383E72A94E15A9B91BC39B449AF7",
      "name": "test",
      "description": "modify test"
    }
  ],
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

2.4.7 url查询

功能说明: url查询

1) API请求数据

POST

https://{host}/api/v1/namespaces/@namespace/queryurl

2) 输入参数

• Body参数

参数	类型	是否必选	描述
obj	object	是	url查询
obj.url	string	是	url地址 最小长度:1 最大长度:4096

请求样例:

URL查询结果, 查询条件: 所有包含www.testurl.com的URL类型对象

POST https://192.168.1.1/api/v1/namespaces/public/queryurl

```
{
  "url": "www.testurl.com"
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	返回url所属分组
data.data	string	url分组信息 最小长度:1

请求成功样例:

```
{
  "data": {
    "data": "你所查询的URL类别为 [test]"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

2.4.8 获取urlLib库版本信息

功能说明: 获取urlLib库版本信息

1) API请求数据

GET https://{host}/api/v1/namespaces/@namespace/urllibinfo

2) 输入参数

请求样例:

获取urlLib库版本信息

GET https://192.168.1.1/api/v1/namespaces/public/urllibinfo

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	urlLib库版本信息
data.date	string	当前库日期 最小长度:2 最大长度:64

data.expire	string	url库序列号信息 最小长度:2 最大长度:128
-------------	--------	---------------------------------

请求成功样例:

```
{
  "data": {
    "date": "2018-01-29 09:00:00",
    "expire": "2020-10-04"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

2.5 服务

2.5.1 获取预定义服务或自定义服务或服务组列表，根据uri参数servType指定

功能说明：通过此接口，可以获取预定义服务或自定义服务或服务组列表，根据uri参数servType指定，servType的可选值为PREDEF_SERV、USRDEF_SERV、SERV_GRP

1) API请求数据

GET <https://{host}/api/v1/namespaces/@namespace/services>

2) 输入参数

• URL 参数

参数	类型	是否必选	描述
_start	int32	否	起始位置, 从 0 开始 最小值:0 最大值:2000000
servType	string	否	过滤服务类别 枚举值:[PREDEF_SERV:预定义服务 USRDEF_SERV:自定义服务 SERV:服务 SERV_GRP:服务组]
_length	int32	否	获取最大长度, 限制最大值, 如果数据不够 可能返回小于 length 最小值:1 最大值:200 默认值:true
_search	string	否	模糊搜索关键字 最大长度:95
_select	string	否	选择字段, 字段用逗号分隔 最大长度:1000

__nameprefix	string	否	服务名字前缀过滤 最小长度:1 最大长度:95
_sortby	string	否	指定排序字段 最大长度:100
queryListModule	string	否	请求服务列表的模块名 枚举值:[OBJ_WAF:对象-WEB应用防护]
_order	string	否	指定排序方式（正序/倒序） 默认值:true 枚举值:[asc:按升序排列 desc:按降序排列]

请求样例:

获取服务信息列表，查询条件：服务类型为自定义服务，起始位置为0，最多展示100条

GET https://192.168.1.1/api/v1/namespaces/public/services?servType=USRDEF_SERV&_start=0&length=100

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	分页数据
data.totalItems	int32	总共多少项目
data.totalPages	int32	总共多少页
data.pageNumber	int32	当前页码，从 1 开始
data.pageSize	int32	每页多大
data.itemsOffset	int32	当前条目偏移，从 0 开始
data.itemLength	int32	数据列表长度
data.items	object[]	有效数据列表
data.items.uuid	string	唯一资源id
data.items.name	string	服务名称 最小长度:1 最大长度:95
data.items.description	string	描述字段 最小长度:0 最大长度:95

data.items.servType	string	预定义服务或自定义服务或服务组 枚举值:[PREDEF_SERV:预定义服务 USRDEF_SERV:自定义服务 SERV_GRP:服务组]
data.items.tcpEntrys	object[]	tcp条目列表
data.items.tcpEntrys.srcPorts	object[]	源端口信息
data.items.tcpEntrys.srcPorts.start	int32	网络端口起始值 最小值:0 最大值:65535
data.items.tcpEntrys.srcPorts.end	int32	网络端口结束值 最小值:0 最大值:65535
data.items.tcpEntrys.dstPorts	object[]	目的端口信息
data.items.tcpEntrys.dstPorts.start	int32	起始端口 最小值:0 最大值:65535
data.items.tcpEntrys.dstPorts.end	int32	结束端口 最小值:0 最大值:65535
data.items.udpEntrys	object[]	udp条目列表
data.items.udpEntrys.srcPorts	object[]	源端口信息
data.items.udpEntrys.srcPorts.start	int32	网络端口起始值 最小值:0 最大值:65535
data.items.udpEntrys.srcPorts.end	int32	网络端口结束值 最小值:0 最大值:65535
data.items.udpEntrys.dstPorts	object[]	目的端口信息
data.items.udpEntrys.dstPorts.start	int32	起始端口 最小值:0 最大值:65535
data.items.udpEntrys.dstPorts.end	int32	结束端口 最小值:0 最大值:65535
data.items.icmpEntrys	object[]	icmp条目列表
data.items.icmpEntrys.type	uint8	类型, 255表示全部 最小值:0 最大值:255
data.items.icmpEntrys.code	uint8	代码, 255表示全部 最小值:0 最大值:255

data.items.icmpv6Entry s	object[]	icmpv6条目列表
data.items.icmpv6Entry s.type	uint8	类型，255表示全部 最小值:0 最大值:255
data.items.icmpv6Entry s.code	uint8	代码，255表示全部 最小值:0 最大值:255
data.items.other	uint16[]	其他协议号列表,其中256表示除tcp,udp,icmp,icmpv6 之外的所有其他协议号
data.items.servsInfo	string[]	服务组勾选的服务
data.items.hasref	bool	标识是否被引用

请求成功样例:

```
{
  "data": {
    "itemsOffset": 0,
    "items": [
      {
        "hasref": false,
        "tcpEntry": {
          "srcPorts": [
            {
              "start": 0,
              "end": 65535
            }
          ],
          "dstPorts": [
            {
              "start": 22
            }
          ]
        }
      }
    ],
    "name": "test",
    "uuid": "2496D4F5411D42E782E3DC2076CC62EC",
    "servType": "USRDEF_SERV",
    "description": ""
  },
  "totalItems": 1,
  "pageNumber": 1,
  "itemLength": 1,
  "totalPages": 1,
  "pageSize": 100
},
{
  "code": 0,
  "message": "成功"
}
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

2.5.2 获取单条服务或服务组信息

功能说明：获取单条服务或服务组信息

1) API请求数据

GET `https://{host}/api/v1/namespaces/@namespace/services/@name`

2) 输入参数

请求样例:

获取单个服务信息, 查询条件: 查看名为test的服务

GET `https://192.168.1.1/api/v1/namespaces/public/services/test`

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	服务或服务组信息
data.uuid	string	唯一资源id
data.name	string	服务名称 最小长度:1 最大长度:95
data.description	string	描述字段 最小长度:0 最大长度:95
data.servType	string	预定义服务或自定义服务或服务组 枚举值:[PREDEF_SERV:预定义服务 USRDEF_SERV:自定义服务 SERV_GRP:服务组]
data.tcpEntrys	object[]	tcp条目列表
data.tcpEntrys.srcPorts	object[]	源端口信息
data.tcpEntrys.srcPorts.start	int32	网络端口起始值 最小值:0 最大值:65535
data.tcpEntrys.srcPorts.end	int32	网络端口结束值 最小值:0 最大值:65535
data.tcpEntrys.dstPorts	object[]	目的端口信息
data.tcpEntrys.dstPorts.start	int32	起始端口 最小值:0 最大值:65535
data.tcpEntrys.dstPorts.end	int32	结束端口 最小值:0 最大值:65535
data.udpEntrys	object[]	udp条目列表
data.udpEntrys.srcPorts	object[]	源端口信息

data.udpEntrys.srcPorts.start	int32	网络端口起始值 最小值:0 最大值:65535
data.udpEntrys.srcPorts.end	int32	网络端口结束值 最小值:0 最大值:65535
data.udpEntrys.dstPorts	object[]	目的端口信息
data.udpEntrys.dstPorts.start	int32	起始端口 最小值:0 最大值:65535
data.udpEntrys.dstPorts.end	int32	结束端口 最小值:0 最大值:65535
data.icmpEntrys	object[]	icmp条目列表
data.icmpEntrys.type	uint8	类型, 255表示全部 最小值:0 最大值:255
data.icmpEntrys.code	uint8	代码, 255表示全部 最小值:0 最大值:255
data.icmpv6Entrys	object[]	icmpv6条目列表
data.icmpv6Entrys.type	uint8	类型, 255表示全部 最小值:0 最大值:255
data.icmpv6Entrys.code	uint8	代码, 255表示全部 最小值:0 最大值:255
data.other	uint16[]	其他协议号列表,其中256表示除tcp,udp,icmp,icmpv6之外的所有其他协议号
data.servsInfo	string[]	服务组勾选的服务
data.hasref	bool	标识是否被引用

请求成功样例:

```
{
  "data": {
    "hasref": false,
    "tcpEntrys": [
      {
        "srcPorts": [
          {
            "start": 0,
            "end": 65535
          }
        ],
        "dstPorts": [
          {
            "start": 22
          }
        ]
      }
    ],
    "name": "test",
    "uuid": "2496D4F5411D42E782E3DC2076CC62EC",
    "servType": "USRDEF_SERV",
  }
}
```



```

    "description": ""
  },
  "code": 0,
  "message": "成功"
}

```

请求失败样例:

```

{
  "data": "",
  "code": 22,
  "message": "无效参数"
}

```

2.5.3 批量获取服务或服务组信息

功能说明: 批量获取服务或服务组信息

1) API请求数据

POST

https://{host}/api/batch/v1/namespaces/@namespace/services?_method=get

2) 输入参数

- Body参数

参数	类型	是否必选	描述
obj	object[]	是	服务和服务组列表
obj.name	string	是	服务名称 最小长度:1 最大长度:95
obj.description	string	否	描述字段 最小长度:0 最大长度:95
obj.servType	string	是	预定义服务或自定义服务或服务组 枚举值:[PREDEF_SERV:预定义服务 USRDEF_SERV:自定义服务 SERV_GRP:服务组]
obj.tcpEntrys	object[]	否	tcp条目列表
obj.tcpEntrys.srcPorts	object[]	否	源端口信息
obj.tcpEntrys.srcPorts.start	int32	是	网络端口起始值 最小值:0 最大值:65535
obj.tcpEntrys.srcPorts.end	int32	否	网络端口结束值 最小值:0 最大值:65535
obj.tcpEntrys.dstPorts	object[]	是	目的端口信息
obj.tcpEntrys.dstPorts.start	int32	是	起始端口 最小值:0 最大值:65535
obj.tcpEntrys.dstPorts.end	int32	否	结束端口 最小值:0 最大值:65535

obj.udpEntrys	object[]	否	udp条目列表
obj.udpEntrys.srcPorts	object[]	否	源端口信息
obj.udpEntrys.srcPorts.start	int32	是	网络端口起始值 最小值:0 最大值:65535
obj.udpEntrys.srcPorts.end	int32	否	网络端口结束值 最小值:0 最大值:65535
obj.udpEntrys.dstPorts	object[]	是	目的端口信息
obj.udpEntrys.dstPorts.start	int32	是	起始端口 最小值:0 最大值:65535
obj.udpEntrys.dstPorts.end	int32	否	结束端口 最小值:0 最大值:65535
obj.icmpEntrys	object[]	否	icmp条目列表
obj.icmpEntrys.type	uint8	是	类型, 255表示全部 最小值:0 最大值:255
obj.icmpEntrys.code	uint8	是	代码, 255表示全部 最小值:0 最大值:255
obj.icmpv6Entrys	object[]	否	icmpv6条目列表
obj.icmpv6Entrys.type	uint8	是	类型, 255表示全部 最小值:0 最大值:255
obj.icmpv6Entrys.code	uint8	是	代码, 255表示全部 最小值:0 最大值:255
obj.other	uint16[]	否	其他协议号列表,其中256表示除tcp,udp,icmp,icmpv6之外的所有其他协议号
obj.servsInfo	string[]	否	服务组勾选的服务
obj.hasref	bool	否	标识是否被引用

请求样例:

一次性获取多条服务信息, 查询条件: 获取服务test1,test2的信息

```
POST https://192.168.1.1/api/batch/v1/namespaces/public/services?_method=GET
[
  {
    "name": "test1"
  },
  {
    "name": "test2"
  }
]
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	服务和服务组列表
data.uuid	string	唯一资源id
data.name	string	服务名称 最小长度:1 最大长度:95
data.description	string	描述字段 最小长度:0 最大长度:95
data.servType	string	预定义服务或自定义服务或服务组 枚举值:[PREDEF_SERV:预定义服务 USRDEF_SERV:自定义服务 SERV_GRP:服务组]
data.tcpEntrys	object[]	tcp条目列表
data.tcpEntrys.srcPorts	object[]	源端口信息
data.tcpEntrys.srcPorts.start	int32	网络端口起始值 最小值:0 最大值:65535
data.tcpEntrys.srcPorts.end	int32	网络端口结束值 最小值:0 最大值:65535
data.tcpEntrys.dstPorts	object[]	目的端口信息
data.tcpEntrys.dstPorts.start	int32	起始端口 最小值:0 最大值:65535
data.tcpEntrys.dstPorts.end	int32	结束端口 最小值:0 最大值:65535
data.udpEntrys	object[]	udp条目列表
data.udpEntrys.srcPorts	object[]	源端口信息
data.udpEntrys.srcPorts.start	int32	网络端口起始值 最小值:0 最大值:65535
data.udpEntrys.srcPorts.end	int32	网络端口结束值 最小值:0 最大值:65535
data.udpEntrys.dstPorts	object[]	目的端口信息

data.udpEntrys.dstPorts.start	int32	起始端口 最小值:0 最大值:65535
data.udpEntrys.dstPorts.end	int32	结束端口 最小值:0 最大值:65535
data.icmpEntrys	object[]	icmp条目列表
data.icmpEntrys.type	uint8	类型, 255表示全部 最小值:0 最大值:255
data.icmpEntrys.code	uint8	代码, 255表示全部 最小值:0 最大值:255
data.icmpv6Entrys	object[]	icmpv6条目列表
data.icmpv6Entrys.type	uint8	类型, 255表示全部 最小值:0 最大值:255
data.icmpv6Entrys.code	uint8	代码, 255表示全部 最小值:0 最大值:255
data.other	uint16[]	其他协议号列表,其中256表示除tcp,udp,icmp,icmpv6之外的所有其他协议号
data.servsInfo	string[]	服务组勾选的服务
data.hasref	bool	标识是否被引用

请求成功样例:

```
{
  "data": [
    {
      "hasref": true,
      "tcpEntrys": [
        {
          "srcPorts": [
            {
              "start": 0,
              "end": 65535
            }
          ],
          "dstPorts": [
            {
              "start": 22
            }
          ]
        }
      ],
      "name": "test1",
      "uuid": "E0F779F6D7B74025BD36FCF32AA963A5",
      "servType": "USRDEF_SERV",
      "description": ""
    },
    {
      "hasref": true,
      "tcpEntrys": [
        {
          "srcPorts": [
            {
              "start": 0,
              "end": 65535
            }
          ]
        }
      ]
    }
  ]
}
```

```

      "dstPorts": [
        {
          "start": 22
        }
      ],
    },
    "name": "test2",
    "uuid": "37AECA161142466AA55BF7760CA66813",
    "servType": "USRDEF_SERV",
    "description": ""
  }
],
"code": 0,
"message": "成功"
}

```

请求失败样例:

```

{
  "data": "",
  "code": 22,
  "message": "无效参数"
}

```

2.5.4 添加服务或服务组

功能说明: 添加服务或服务组

1) API请求数据

POST <https://{host}/api/v1/namespaces/@namespace/services>

2) 输入参数

- Body参数

参数	类型	是否必选	描述
obj	object	是	服务或服务组信息
obj.name	string	是	服务名称 最小长度:1 最大长度:95
obj.description	string	否	描述字段 最小长度:0 最大长度:95
obj.servType	string	是	预定义服务或自定义服务或服务组 枚举值:[PREDEF_SERV:预定义服务 USRDEF_SERV:自定义服务 SERV_GRP:服务组]
obj.tcpEntrys	object[]	否	tcp条目列表
obj.tcpEntrys.srcPorts	object[]	否	源端口信息
obj.tcpEntrys.srcPorts.start	int32	是	网络端口起始值 最小值:0 最大值:65535
obj.tcpEntrys.srcPorts.end	int32	否	网络端口结束值 最小值:0 最大值:65535

obj.tcpEntrys.dstPorts	object[]	是	目的端口信息
obj.tcpEntrys.dstPorts.start	int32	是	起始端口 最小值:0 最大值:65535
obj.tcpEntrys.dstPorts.end	int32	否	结束端口 最小值:0 最大值:65535
obj.udpEntrys	object[]	否	udp条目列表
obj.udpEntrys.srcPorts	object[]	否	源端口信息
obj.udpEntrys.srcPorts.start	int32	是	网络端口起始值 最小值:0 最大值:65535
obj.udpEntrys.srcPorts.end	int32	否	网络端口结束值 最小值:0 最大值:65535
obj.udpEntrys.dstPorts	object[]	是	目的端口信息
obj.udpEntrys.dstPorts.start	int32	是	起始端口 最小值:0 最大值:65535
obj.udpEntrys.dstPorts.end	int32	否	结束端口 最小值:0 最大值:65535
obj.icmpEntrys	object[]	否	icmp条目列表
obj.icmpEntrys.type	uint8	是	类型, 255表示全部 最小值:0 最大值:255
obj.icmpEntrys.code	uint8	是	代码, 255表示全部 最小值:0 最大值:255
obj.icmpv6Entrys	object[]	否	icmpv6条目列表
obj.icmpv6Entrys.type	uint8	是	类型, 255表示全部 最小值:0 最大值:255
obj.icmpv6Entrys.code	uint8	是	代码, 255表示全部 最小值:0 最大值:255
obj.other	uint16[]	否	其他协议号列表,其中256表示除tcp,udp,icmp,icmpv6之外的所有其他协议号
obj.servsInfo	string[]	否	服务组勾选的服务
obj.hasref	bool	否	标识是否被引用

请求样例:

添加单个服务, 服务名字为test, tcp目的端口为22, 源端口为0-65535

```
POST https://192.168.1.1/api/v1/namespaces/public/services
{
  "tcpEntrys": [
    {
      "srcPorts": [
```

```

        {
            "start": 0,
            "end": 65535
        },
        "dstPorts": [
            {
                "start": 22
            }
        ]
    },
    "servType": "USRDEF_SERV",
    "name": "test",
    "description": ""
}

```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	服务或服务组信息
data.uuid	string	唯一资源id
data.name	string	服务名称 最小长度:1 最大长度:95
data.description	string	描述字段 最小长度:0 最大长度:95
data.servType	string	预定义服务或自定义服务或服务组 枚举值:[PREDEF_SERV:预定义服务 USRDEF_SERV:自定义服务 SERV_GRP:服务组]
data.tcpEntrys	object[]	tcp条目列表
data.tcpEntrys.srcPorts	object[]	源端口信息
data.tcpEntrys.srcPorts.start	int32	网络端口起始值 最小值:0 最大值:65535
data.tcpEntrys.srcPorts.end	int32	网络端口结束值 最小值:0 最大值:65535
data.tcpEntrys.dstPorts	object[]	目的端口信息
data.tcpEntrys.dstPorts.start	int32	起始端口 最小值:0 最大值:65535
data.tcpEntrys.dstPorts.end	int32	结束端口 最小值:0 最大值:65535
data.udpEntrys	object[]	udp条目列表

data.udpEntrys.srcPorts	object[]	源端口信息
data.udpEntrys.srcPorts.start	int32	网络端口起始值 最小值:0 最大值:65535
data.udpEntrys.srcPorts.end	int32	网络端口结束值 最小值:0 最大值:65535
data.udpEntrys.dstPorts	object[]	目的端口信息
data.udpEntrys.dstPorts.start	int32	起始端口 最小值:0 最大值:65535
data.udpEntrys.dstPorts.end	int32	结束端口 最小值:0 最大值:65535
data.icmpEntrys	object[]	icmp条目列表
data.icmpEntrys.type	uint8	类型, 255表示全部 最小值:0 最大值:255
data.icmpEntrys.code	uint8	代码, 255表示全部 最小值:0 最大值:255
data.icmpv6Entrys	object[]	icmpv6条目列表
data.icmpv6Entrys.type	uint8	类型, 255表示全部 最小值:0 最大值:255
data.icmpv6Entrys.code	uint8	代码, 255表示全部 最小值:0 最大值:255
data.other	uint16[]	其他协议号列表,其中256表示除tcp,udp,icmp,icmpv6之外的所有其他协议号
data.servsInfo	string[]	服务组勾选的服务
data.hasref	bool	标识是否被引用

请求成功样例:

```
{
  "data": {
    "tcpEntrys": [
      {
        "srcPorts": [
          {
            "start": 0,
            "end": 65535
          }
        ],
        "dstPorts": [
          {
            "start": 22
          }
        ]
      }
    ],
    "name": "test",
```



```
{
  "uuid": "3732A58BC8AC42EBB5DAC0242462A9A1",
  "servType": "USRDEF_SERV",
  "description": ""
},
{
  "code": 0,
  "message": "成功"
}
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

2.5.5 修改指定服务或服务组

功能说明: 修改指定服务或服务组

1) API请求数据

PATCH <https://{host}/api/v1/namespaces/@namespace/services/@name>

2) 输入参数

• URL 参数

参数	类型	是否必选	描述
_arrayop	string	否	对象属性是数组的操作控制 枚举值: add:添加对应的数组元素 remove:删除对应的数组元素]

• Body参数

参数	类型	是否必选	描述
obj	object	是	服务或服务组信息
obj.name	string	是	服务名称 最小长度:1 最大长度:95
obj.description	string	否	描述字段 最小长度:0 最大长度:95
obj.servType	string	是	预定义服务或自定义服务或服务组 枚举值: PREDEF_SERV:预定义服务 USRDEF_SERV:自定义服务 SERV_GRP:服务组]
obj.tcpEntrys	object[]	否	tcp条目列表
obj.tcpEntrys.srcPorts	object[]	否	源端口信息
obj.tcpEntrys.srcPorts.start	int32	是	网络端口起始值 最小值:0 最大值:65535

obj.tcpEntrys.srcPorts.end	int32	否	网络端口结束值 最小值:0 最大值:65535
obj.tcpEntrys.dstPorts	object[]	是	目的端口信息
obj.tcpEntrys.dstPorts.start	int32	是	起始端口 最小值:0 最大值:65535
obj.tcpEntrys.dstPorts.end	int32	否	结束端口 最小值:0 最大值:65535
obj.udpEntrys	object[]	否	udp条目列表
obj.udpEntrys.srcPorts	object[]	否	源端口信息
obj.udpEntrys.srcPorts.start	int32	是	网络端口起始值 最小值:0 最大值:65535
obj.udpEntrys.srcPorts.end	int32	否	网络端口结束值 最小值:0 最大值:65535
obj.udpEntrys.dstPorts	object[]	是	目的端口信息
obj.udpEntrys.dstPorts.start	int32	是	起始端口 最小值:0 最大值:65535
obj.udpEntrys.dstPorts.end	int32	否	结束端口 最小值:0 最大值:65535
obj.icmpEntrys	object[]	否	icmp条目列表
obj.icmpEntrys.type	uint8	是	类型, 255表示全部 最小值:0 最大值:255
obj.icmpEntrys.code	uint8	是	代码, 255表示全部 最小值:0 最大值:255
obj.icmpv6Entrys	object[]	否	icmpv6条目列表
obj.icmpv6Entrys.type	uint8	是	类型, 255表示全部 最小值:0 最大值:255
obj.icmpv6Entrys.code	uint8	是	代码, 255表示全部 最小值:0 最大值:255
obj.other	uint16[]	否	其他协议号列表,其中256表示除tcp,udp,icmp,icmpv6之外的所有其他协议号
obj.servsInfo	string[]	否	服务组勾选的服务
obj.hasref	bool	否	标识是否被引用

请求样例:

增量修改单个服务, 将服务test的tcp目的端口改为80, 其他部分不改

```
PATCH https://192.168.1.1/api/v1/namespaces/public/services/test
{
  "tcpEntrys": [
    {
      "srcPorts": [
        {
          "start": 0,
          "end": 65535
        }
      ],
      "dstPorts": [
        {
          "start": 80
        }
      ]
    }
  ]
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	服务或服务组信息
data.uuid	string	唯一资源id
data.name	string	服务名称 最小长度:1 最大长度:95
data.description	string	描述字段 最小长度:0 最大长度:95
data.servType	string	预定义服务或自定义服务或服务组 枚举值:[PREDEF_SERV:预定义服务 USRDEF_SERV:自定义服务 SERV_GRP:服务组]
data.tcpEntrys	object[]	tcp条目列表
data.tcpEntrys.srcPorts	object[]	源端口信息
data.tcpEntrys.srcPorts.start	int32	网络端口起始值 最小值:0 最大值:65535
data.tcpEntrys.srcPorts.end	int32	网络端口结束值 最小值:0 最大值:65535
data.tcpEntrys.dstPorts	object[]	目的端口信息
data.tcpEntrys.dstPorts.start	int32	起始端口 最小值:0 最大值:65535
data.tcpEntrys.dstPorts.end	int32	结束端口 最小值:0 最大值:65535

data.udpEntrys	object[]	udp条目列表
data.udpEntrys.srcPorts	object[]	源端口信息
data.udpEntrys.srcPorts.start	int32	网络端口起始值 最小值:0 最大值:65535
data.udpEntrys.srcPorts.end	int32	网络端口结束值 最小值:0 最大值:65535
data.udpEntrys.dstPorts	object[]	目的端口信息
data.udpEntrys.dstPorts.start	int32	起始端口 最小值:0 最大值:65535
data.udpEntrys.dstPorts.end	int32	结束端口 最小值:0 最大值:65535
data.icmpEntrys	object[]	icmp条目列表
data.icmpEntrys.type	uint8	类型, 255表示全部 最小值:0 最大值:255
data.icmpEntrys.code	uint8	代码, 255表示全部 最小值:0 最大值:255
data.icmpv6Entrys	object[]	icmpv6条目列表
data.icmpv6Entrys.type	uint8	类型, 255表示全部 最小值:0 最大值:255
data.icmpv6Entrys.code	uint8	代码, 255表示全部 最小值:0 最大值:255
data.other	uint16[]	其他协议号列表,其中256表示除tcp,udp,icmp,icmpv6之外的所有其他协议号
data.servsInfo	string[]	服务组勾选的服务
data.hasref	bool	标识是否被引用

请求成功样例:

```
{
  "data": {
    "tcpEntrys": [
      {
        "srcPorts": [
          {
            "start": 0,
            "end": 65535
          }
        ],
        "dstPorts": [
          {
            "start": 80
          }
        ]
      }
    ]
  }
}
```

```

    },
    "name": "test",
    "uuid": "3732A58BC8AC42EBB5DAC0242462A9A1",
    "servType": "USRDEF_SERV",
    "description": ""
  },
  "code": 0,
  "message": "成功"
}

```

请求失败样例:

```

{
  "data": "",
  "code": 22,
  "message": "无效参数"
}

```

2.5.6 全量修改指定服务或服务组

功能说明: 全量修改指定服务或服务组

1) API请求数据

PUT <https://{host}/api/v1/namespaces/@namespace/services/@name>

2) 输入参数

• Body参数

参数	类型	是否必选	描述
obj	object	是	服务或服务组信息
obj.name	string	是	服务名称 最小长度:1 最大长度:95
obj.description	string	否	描述字段 最小长度:0 最大长度:95
obj.servType	string	是	预定义服务或自定义服务或服务组 枚举值:[PREDEF_SERV:预定义服务 USRDEF_SERV:自定义服务 SERV_GRP:服务组]
obj.tcpEntrys	object[]	否	tcp条目列表
obj.tcpEntrys.srcPorts	object[]	否	源端口信息
obj.tcpEntrys.srcPorts.start	int32	是	网络端口起始值 最小值:0 最大值:65535
obj.tcpEntrys.srcPorts.end	int32	否	网络端口结束值 最小值:0 最大值:65535
obj.tcpEntrys.dstPorts	object[]	是	目的端口信息
obj.tcpEntrys.dstPorts.start	int32	是	起始端口 最小值:0 最大值:65535

obj.tcpEntrys.dstPorts.end	int32	否	结束端口 最小值:0 最大值:65535
obj.udpEntrys	object[]	否	udp条目列表
obj.udpEntrys.srcPorts	object[]	否	源端口信息
obj.udpEntrys.srcPorts.start	int32	是	网络端口起始值 最小值:0 最大值:65535
obj.udpEntrys.srcPorts.end	int32	否	网络端口结束值 最小值:0 最大值:65535
obj.udpEntrys.dstPorts	object[]	是	目的端口信息
obj.udpEntrys.dstPorts.start	int32	是	起始端口 最小值:0 最大值:65535
obj.udpEntrys.dstPorts.end	int32	否	结束端口 最小值:0 最大值:65535
obj.icmpEntrys	object[]	否	icmp条目列表
obj.icmpEntrys.type	uint8	是	类型, 255表示全部 最小值:0 最大值:255
obj.icmpEntrys.code	uint8	是	代码, 255表示全部 最小值:0 最大值:255
obj.icmpv6Entrys	object[]	否	icmpv6条目列表
obj.icmpv6Entrys.type	uint8	是	类型, 255表示全部 最小值:0 最大值:255
obj.icmpv6Entrys.code	uint8	是	代码, 255表示全部 最小值:0 最大值:255
obj.other	uint16[]	否	其他协议号列表,其中256表示除tcp,udp,icmp,icmpv6之外的所有其他协议号
obj.servsInfo	string[]	否	服务组勾选的服务
obj.hasref	bool	否	标识是否被引用

请求样例:

全量修改单个服务, 将服务test的tcp目的端口改为80

```
PUT https://192.168.1.1/api/v1/namespaces/public/services/test
{
  "tcpEntrys": [
    {
      "srcPorts": [
        {
          "start": 0,
          "end": 65535
        }
      ],
      "dstPorts": [
        {

```

```

        "start": 80
      }
    ]
  },
  "servType": "USRDEF_SERV",
  "name": "test",
  "description": ""
}

```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	服务或服务组信息
data.uuid	string	唯一资源id
data.name	string	服务名称 最小长度:1 最大长度:95
data.description	string	描述字段 最小长度:0 最大长度:95
data.servType	string	预定义服务或自定义服务或服务组 枚举值:[PREDEF_SERV:预定义服务 USRDEF_SERV:自定义服务 SERV_GRP:服务组]
data.tcpEntrys	object[]	tcp条目列表
data.tcpEntrys.srcPorts	object[]	源端口信息
data.tcpEntrys.srcPorts.start	int32	网络端口起始值 最小值:0 最大值:65535
data.tcpEntrys.srcPorts.end	int32	网络端口结束值 最小值:0 最大值:65535
data.tcpEntrys.dstPorts	object[]	目的端口信息
data.tcpEntrys.dstPorts.start	int32	起始端口 最小值:0 最大值:65535
data.tcpEntrys.dstPorts.end	int32	结束端口 最小值:0 最大值:65535
data.udpEntrys	object[]	udp条目列表
data.udpEntrys.srcPorts	object[]	源端口信息

data.udpEntrys.srcPorts.start	int32	网络端口起始值 最小值:0 最大值:65535
data.udpEntrys.srcPorts.end	int32	网络端口结束值 最小值:0 最大值:65535
data.udpEntrys.dstPorts	object[]	目的端口信息
data.udpEntrys.dstPorts.start	int32	起始端口 最小值:0 最大值:65535
data.udpEntrys.dstPorts.end	int32	结束端口 最小值:0 最大值:65535
data.icmpEntrys	object[]	icmp条目列表
data.icmpEntrys.type	uint8	类型, 255表示全部 最小值:0 最大值:255
data.icmpEntrys.code	uint8	代码, 255表示全部 最小值:0 最大值:255
data.icmpv6Entrys	object[]	icmpv6条目列表
data.icmpv6Entrys.type	uint8	类型, 255表示全部 最小值:0 最大值:255
data.icmpv6Entrys.code	uint8	代码, 255表示全部 最小值:0 最大值:255
data.other	uint16[]	其他协议号列表,其中256表示除tcp,udp,icmp,icmpv6之外的所有其他协议号
data.servsInfo	string[]	服务组勾选的服务
data.hasref	bool	标识是否被引用

请求成功样例:

```
{
  "data": {
    "tcpEntrys": [
      {
        "srcPorts": [
          {
            "start": 0,
            "end": 65535
          }
        ],
        "dstPorts": [
          {
            "start": 80
          }
        ]
      }
    ],
    "name": "test",
    "uuid": "3732A58BC8AC42EBB5DAC0242462A9A1",
    "servType": "USRDEF_SERV",
    "description": ""
  }
}
```



```
    },
    "code": 0,
    "message": "成功"
  }
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

2.5.7 删除指定服务或服务组

功能说明: 删除指定服务或服务组

1) API请求数据

DELETE https://{host}/api/v1/namespaces/@namespace/services/@name

2) 输入参数

请求样例:

删除单个服务, 服务名为test

DELETE https://192.168.1.1/api/v1/namespaces/public/services/test

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	服务或服务组信息
data.uuid	string	唯一资源id
data.name	string	服务名称 最小长度:1 最大长度:95
data.description	string	描述字段 最小长度:0 最大长度:95
data.servType	string	预定义服务或自定义服务或服务组 枚举值: PREDEF_SERV:预定义服务 USRDEF_SERV:自定义服务 SERV_GRP:服务组]
data.tcpEntrys	object[]	tcp条目列表
data.tcpEntrys.srcPorts	object[]	源端口信息
data.tcpEntrys.srcPorts.start	int32	网络端口起始值 最小值:0 最大值:65535

data.tcpEntrys.srcPorts.end	int32	网络端口结束值 最小值:0 最大值:65535
data.tcpEntrys.dstPorts	object[]	目的端口信息
data.tcpEntrys.dstPorts.start	int32	起始端口 最小值:0 最大值:65535
data.tcpEntrys.dstPorts.end	int32	结束端口 最小值:0 最大值:65535
data.udpEntrys	object[]	udp条目列表
data.udpEntrys.srcPorts	object[]	源端口信息
data.udpEntrys.srcPorts.start	int32	网络端口起始值 最小值:0 最大值:65535
data.udpEntrys.srcPorts.end	int32	网络端口结束值 最小值:0 最大值:65535
data.udpEntrys.dstPorts	object[]	目的端口信息
data.udpEntrys.dstPorts.start	int32	起始端口 最小值:0 最大值:65535
data.udpEntrys.dstPorts.end	int32	结束端口 最小值:0 最大值:65535
data.icmpEntrys	object[]	icmp条目列表
data.icmpEntrys.type	uint8	类型, 255表示全部 最小值:0 最大值:255
data.icmpEntrys.code	uint8	代码, 255表示全部 最小值:0 最大值:255
data.icmpv6Entrys	object[]	icmpv6条目列表
data.icmpv6Entrys.type	uint8	类型, 255表示全部 最小值:0 最大值:255
data.icmpv6Entrys.code	uint8	代码, 255表示全部 最小值:0 最大值:255
data.other	uint16[]	其他协议号列表,其中256表示除tcp,udp,icmp,icmpv6之外的所有其他协议号
data.servsInfo	string[]	服务组勾选的服务
data.hasref	bool	标识是否被引用

请求成功样例:

```
{
  "data": {
    "tcpEntrys": [
      {
        "srcPorts": [
          {
            "start": 0,
            "end": 65535
          }
        ],
        "dstPorts": [
          {
            "start": 22
          }
        ]
      }
    ],
    "name": "test",
    "uuid": "3732A58BC8AC42EBB5DAC0242462A9A1",
    "servType": "USRDEF_SERV",
    "description": ""
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

2.5.8 批量删除服务或服务组

功能说明: 批量删除服务或服务组

1) API请求数据

POST

https://{host}/api/batch/v1/namespaces/@namespace/services?_method=delete

2) 输入参数

• Body参数

参数	类型	是否必选	描述
obj	object[]	是	服务和服组列表
obj.name	string	是	服务名称 最小长度:1 最大长度:95
obj.description	string	否	描述字段 最小长度:0 最大长度:95
obj.servType	string	是	预定义服务或自定义服务或服务组 枚举值:[PREDEF_SERV:预定义服务 USRDEF_SERV:自定义服务 SERV_GRP:服务组]
obj.tcpEntrys	object[]	否	tcp条目列表
obj.tcpEntrys.srcPorts	object[]	否	源端口信息

obj.tcpEntrys.srcPorts.start	int32	是	网络端口起始值 最小值:0 最大值:65535
obj.tcpEntrys.srcPorts.end	int32	否	网络端口结束值 最小值:0 最大值:65535
obj.tcpEntrys.dstPorts	object[]	是	目的端口信息
obj.tcpEntrys.dstPorts.start	int32	是	起始端口 最小值:0 最大值:65535
obj.tcpEntrys.dstPorts.end	int32	否	结束端口 最小值:0 最大值:65535
obj.udpEntrys	object[]	否	udp条目列表
obj.udpEntrys.srcPorts	object[]	否	源端口信息
obj.udpEntrys.srcPorts.start	int32	是	网络端口起始值 最小值:0 最大值:65535
obj.udpEntrys.srcPorts.end	int32	否	网络端口结束值 最小值:0 最大值:65535
obj.udpEntrys.dstPorts	object[]	是	目的端口信息
obj.udpEntrys.dstPorts.start	int32	是	起始端口 最小值:0 最大值:65535
obj.udpEntrys.dstPorts.end	int32	否	结束端口 最小值:0 最大值:65535
obj.icmpEntrys	object[]	否	icmp条目列表
obj.icmpEntrys.type	uint8	是	类型, 255表示全部 最小值:0 最大值:255
obj.icmpEntrys.code	uint8	是	代码, 255表示全部 最小值:0 最大值:255
obj.icmpv6Entrys	object[]	否	icmpv6条目列表
obj.icmpv6Entrys.type	uint8	是	类型, 255表示全部 最小值:0 最大值:255
obj.icmpv6Entrys.code	uint8	是	代码, 255表示全部 最小值:0 最大值:255
obj.other	uint16[]	否	其他协议号列表,其中256表示除tcp,udp,icmp,icmpv6之外的所有其他协议号
obj.servsInfo	string[]	否	服务组勾选的服务
obj.hasref	bool	否	标识是否被引用

请求样例:

一次性删除多个服务, 服务名为test1,test2

```
POST https://192.168.1.1/api/batch/v1/namespaces/public/services?_method=DELETE
[
  {
    "name": "test1"
  },
  {
    "name": "test2"
  }
]
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	服务和服务组列表
data.uuid	string	唯一资源id
data.name	string	服务名称 最小长度:1 最大长度:95
data.description	string	描述字段 最小长度:0 最大长度:95
data.servType	string	预定义服务或自定义服务或服务组 枚举值:[PREDEF_SERV:预定义服务 USRDEF_SERV:自定义服务 SERV_GRP:服务组]
data.tcpEntrys	object[]	tcp条目列表
data.tcpEntrys.srcPorts	object[]	源端口信息
data.tcpEntrys.srcPorts.start	int32	网络端口起始值 最小值:0 最大值:65535
data.tcpEntrys.srcPorts.end	int32	网络端口结束值 最小值:0 最大值:65535
data.tcpEntrys.dstPorts	object[]	目的端口信息
data.tcpEntrys.dstPorts.start	int32	起始端口 最小值:0 最大值:65535
data.tcpEntrys.dstPorts.end	int32	结束端口 最小值:0 最大值:65535
data.udpEntrys	object[]	udp条目列表

data.udpEntrys.srcPorts	object[]	源端口信息
data.udpEntrys.srcPorts.start	int32	网络端口起始值 最小值:0 最大值:65535
data.udpEntrys.srcPorts.end	int32	网络端口结束值 最小值:0 最大值:65535
data.udpEntrys.dstPorts	object[]	目的端口信息
data.udpEntrys.dstPorts.start	int32	起始端口 最小值:0 最大值:65535
data.udpEntrys.dstPorts.end	int32	结束端口 最小值:0 最大值:65535
data.icmpEntrys	object[]	icmp条目列表
data.icmpEntrys.type	uint8	类型, 255表示全部 最小值:0 最大值:255
data.icmpEntrys.code	uint8	代码, 255表示全部 最小值:0 最大值:255
data.icmpv6Entrys	object[]	icmpv6条目列表
data.icmpv6Entrys.type	uint8	类型, 255表示全部 最小值:0 最大值:255
data.icmpv6Entrys.code	uint8	代码, 255表示全部 最小值:0 最大值:255
data.other	uint16[]	其他协议号列表,其中256表示除tcp,udp,icmp,icmpv6之外的所有其他协议号
data.servsInfo	string[]	服务组勾选的服务
data.hasref	bool	标识是否被引用

请求成功样例:

```
{
  "data": [
    {
      "tcpEntrys": [
        {
          "srcPorts": [
            {
              "start": 0,
              "end": 65535
            }
          ],
          "dstPorts": [
            {
              "start": 22
            }
          ]
        }
      ],
    }
  ],
}
```

```
    "name": "test1",
    "uuid": "0B656A4A007B4F1A83007337419A0FC5",
    "servType": "USRDEF_SERV",
    "description": ""
  },
  {
    "tcpEntrys": [
      {
        "srcPorts": [
          {
            "start": 0,
            "end": 65535
          }
        ],
        "dstPorts": [
          {
            "start": 80
          }
        ]
      }
    ],
    "name": "test2",
    "uuid": "80818027CBDE4B6CA29D1838564FFF7D",
    "servType": "USRDEF_SERV",
    "description": ""
  }
],
"code": 0,
"message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

2.6 IP地址库

2.6.1 IP归属地

2.6.1.1 获取IP归属地策略

功能说明: 通过此接口, 获取符合条件的IP归属地策略

1) API请求数据

GET

https://{host}/api/v1/namespaces/@namespace/iplocations

2) 输入参数

- URL 参数

参数	类型	是否必选	描述
_start	int32	否	起始位置, 从 0 开始 最小值:0 最大值:2000000
_search	string	否	模糊搜索关键字 最大长度:95

_length	int32	否	获取最大长度，限制最大值，如果数据不够可能返回小于 length 最小值:1 最大值:200 默认值:true
---------	-------	---	--

请求样例:

获取IP归属地策略

GET https://192.168.1.1/api/v1/namespaces/public/iplocations

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	分页数据
data.totalItems	int32	总共多少项目
data.totalPages	int32	总共多少页
data.pageNumber	int32	当前页码，从 1 开始
data.pageSize	int32	每页多大
data.itemsOffset	int32	当前条目偏移，从 0 开始
data.itemLength	int32	数据列表长度
data.items	object[]	有效数据列表
data.items.uuid	string	唯一标识id
data.items.IPAddress	string	只支持IPv4地址，格式如下 1、单个IP地址，例如：192.168.0.1 2、IP范围，起始IP-结束IP(起始IP小于结束IP)，例如： 192.168.0.1-192.168.0.150 3、IP地址/子网掩码，例如： 192.168.0.1/255.255.255.0或192.168.0.1/24 最大长度:35
data.items.ipRange	object	只支持IPv4地址，格式如下 1、单个IP地址，例如：192.168.0.1 2、IP范围，起始IP-结束IP(起始IP小于结束IP)，例如： 192.168.0.1-192.168.0.150 3、IP地址/子网掩码，例如： 192.168.0.1/255.255.255.0或192.168.0.1/24 格式:ip-range
data.items.ipRange.start	string	ip范围的起始位置 格式:ip-address
data.items.ipRange.end	string	ip范围的终止位置 格式:ip-address
data.items.ipRange.bits	uint8	ip掩码简写

data.items.ipRange.mask	string	ip掩码 格式:ip-address
data.items.updateTime	string	更新时间
data.items.boardAndCountry	string	正确的归属地

请求成功样例:

```
{
  "data": {
    "itemsOffset": 0,
    "items": [
      {
        "updateTime": "2020-06-29 10:40",
        "IPAddress": "2.2.0.0/255.255.0.0",
        "boardAndCountry": "亚太地区/不丹",
        "uuid": "6C8FB1DE9E6B468EB05F7BEEBD783331",
        "ipRange": {
          "start": "2.2.0.0",
          "end": "2.2.255.255"
        }
      },
      {
        "updateTime": "2020-06-29 10:40",
        "IPAddress": "2.1.2.0/24",
        "boardAndCountry": "亚太地区/不丹",
        "uuid": "74156337831F448E806A977898840ABB",
        "ipRange": {
          "start": "2.1.2.0",
          "end": "2.1.2.255"
        }
      },
      {
        "updateTime": "2020-06-29 10:40",
        "IPAddress": "2.1.1.11-2.1.1.111",
        "boardAndCountry": "亚太地区/不丹",
        "uuid": "35BF1B6B076647C7B3624E4D398974ED",
        "ipRange": {
          "start": "2.1.1.11",
          "end": "2.1.1.111"
        }
      },
      {
        "updateTime": "2020-06-29 10:39",
        "IPAddress": "2.1.1.1",
        "boardAndCountry": "亚太地区/不丹",
        "uuid": "3701E46404084A3DAB8C6C14E158A08E",
        "ipRange": {
          "start": "2.1.1.1",
          "end": "2.1.1.1"
        }
      },
      {
        "updateTime": "2020-06-29 17:37",
        "IPAddress": "20.1.1.1",
        "boardAndCountry": "非洲地区/埃塞俄比亚",
        "uuid": "1C72AAA35BCA4DD58E9850F975846D52",
        "ipRange": {
          "start": "20.1.1.1",
          "end": "20.1.1.1"
        }
      },
      {
        "updateTime": "2020-06-29 17:37",
        "IPAddress": "20.1.1.2",
        "boardAndCountry": "非洲地区/埃塞俄比亚",
        "uuid": "C0B727627EAC4473A3D4E3A246B8B5D6",
        "ipRange": {
          "start": "20.1.1.2",
          "end": "20.1.1.2"
        }
      }
    ]
  }
}
```

```
      "updateTime": "2020-06-29 17:37",
      "IPAddress": "20.1.1.3",
      "boardAndCountry": "非洲地区/埃塞俄比亚",
      "uuid": "E81F7180BF1B42E0961B4A719951E53C",
      "ipRange": {
        "start": "20.1.1.3",
        "end": "20.1.1.3"
      }
    },
    "totalItems": 7,
    "pageNumber": 1,
    "itemLength": 7,
    "totalPages": 1,
    "pageSize": 100
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

2.6.1.2 查询一条IP归属地策略

功能说明: 通过此接口, 可以根据名称获取该条IP归属地策略

1) API请求数据

GET

https://{host}/api/v1/namespaces/@namespace/iplocations/@policyname

2) 输入参数

请求样例:

查询一条IP归属地策略

```
GET https://192.168.1.1/api/v1/namespaces/public/iplocations/20.1.1.3
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	IP归属地
data.uuid	string	唯一标识id
data.IPAddress	string	只支持IPv4地址, 格式如下 1、单个IP地址, 例如: 192.168.0.1 2、IP范围, 起始IP-结束IP(起始IP小于结束IP), 例如: 192.168.0.1-192.168.0.150 3、IP地址/子网掩码, 例如: 192.168.0.1/255.255.255.0或192.168.0.1/24 最大长度:35

data.ipRange	object	只支持IPv4地址，格式如下 1、单个IP地址，例如：192.168.0.1 2、IP范围，起始IP-结束IP(起始IP小于结束IP)，例如： 192.168.0.1-192.168.0.150 3、IP地址/子网掩码，例如： 192.168.0.1/255.255.255.0或192.168.0.1/24 格式:ip-range
data.ipRange.start	string	ip范围的起始位置 格式:ip-address
data.ipRange.end	string	ip范围的终止位置 格式:ip-address
data.ipRange.bits	uint8	ip掩码简写
data.ipRange.mask	string	ip掩码 格式:ip-address
data.updateTime	string	更新时间
data.boardAndCountry	string	正确的归属地

请求成功样例：

```
{
  "data": {
    "updateTime": "2020-06-29 17:37",
    "IPAddress": "20.1.1.3",
    "boardAndCountry": "非洲地区/埃塞俄比亚",
    "uuid": "E81F7180BF1B42E0961B4A719951E53C",
    "ipRange": {
      "start": "20.1.1.3",
      "end": "20.1.1.3"
    }
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

2.6.1.3 添加一条IP归属地策略

功能说明：通过此接口，可以添加一条IP归属地策略

1) API请求数据

POST https://{host}/api/v1/namespaces/@namespace/iplocations

2) 输入参数

- Body参数

参数	类型	是否必选	描述
policy	object	是	IP归属地

policy.IPAddress	string	是	只支持IPv4地址，格式如下 1、单个IP地址，例如：192.168.0.1 2、IP范围，起始IP–结束IP(起始IP小于结束IP)，例如： 192.168.0.1–192.168.0.150 3、IP地址/子网掩码，例如： 192.168.0.1/255.255.255.0或192.168.0.1/24 最大长度:35
policy.ipRange	object	是	只支持IPv4地址，格式如下 1、单个IP地址，例如：192.168.0.1 2、IP范围，起始IP–结束IP(起始IP小于结束IP)，例如： 192.168.0.1–192.168.0.150 3、IP地址/子网掩码，例如： 192.168.0.1/255.255.255.0或192.168.0.1/24 格式:ip-range
policy.ipRange.start	string	是	ip范围的起始位置 格式:ip-address
policy.ipRange.end	string	否	ip范围的终止位置 格式:ip-address
policy.ipRange.bits	uint8	否	ip掩码简写
policy.ipRange.mask	string	否	ip掩码 格式:ip-address
policy.updateTime	string	是	更新时间
policy.boardAndCountry	string	否	正确的归属地

请求样例：

添加一条IP归属地策略

```
POST https://192.168.1.1/api/v1/namespaces/public/iplocations
{
  "IPAddress": "20.1.1.1",
  "boardAndCountry": "非洲地区/埃塞俄比亚"
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	IP归属地
data.uuid	string	唯一标识id

data.IPAddress	string	只支持IPv4地址，格式如下 1、单个IP地址，例如：192.168.0.1 2、IP范围，起始IP-结束IP(起始IP小于结束IP)，例如： 192.168.0.1-192.168.0.150 3、IP地址/子网掩码，例如： 192.168.0.1/255.255.255.0或192.168.0.1/24 最大长度:35
data.ipRange	object	只支持IPv4地址，格式如下 1、单个IP地址，例如：192.168.0.1 2、IP范围，起始IP-结束IP(起始IP小于结束IP)，例如： 192.168.0.1-192.168.0.150 3、IP地址/子网掩码，例如： 192.168.0.1/255.255.255.0或192.168.0.1/24 格式:ip-range
data.ipRange.start	string	ip范围的起始位置 格式:ip-address
data.ipRange.end	string	ip范围的终止位置 格式:ip-address
data.ipRange.bits	uint8	ip掩码简写
data.ipRange.mask	string	ip掩码 格式:ip-address
data.updateTime	string	更新时间
data.boardAndCountry	string	正确的归属地

请求成功样例：

```
{
  "data": {
    "updateTime": "2020-06-29 17:37",
    "IPAddress": "20.1.1.1",
    "boardAndCountry": "非洲地区/埃塞俄比亚",
    "uuid": "1C72AAA35BCA4DD58E9850F975846D52",
    "ipRange": {
      "start": "20.1.1.1",
      "end": "20.1.1.1"
    }
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

2.6.1.4 修改IP归属地策略

功能说明：通过此接口，可以通过策略名，修改对应的IP归属地策略

1) API请求数据

PATCH <https://{host}/api/v1/namespaces/@namespace/iplocations/@policyname>

2) 输入参数

• Body参数

参数	类型	是否必选	描述
policy	object	是	IP归属地
policy.IPAddress	string	是	只支持IPv4地址，格式如下 1、单个IP地址，例如：192.168.0.1 2、IP范围，起始IP-结束IP(起始IP小于结束IP)，例如： 192.168.0.1-192.168.0.150 3、IP地址/子网掩码，例如： 192.168.0.1/255.255.255.0或192.168.0.1/24 最大长度:35
policy.ipRange	object	是	只支持IPv4地址，格式如下 1、单个IP地址，例如：192.168.0.1 2、IP范围，起始IP-结束IP(起始IP小于结束IP)，例如： 192.168.0.1-192.168.0.150 3、IP地址/子网掩码，例如： 192.168.0.1/255.255.255.0或192.168.0.1/24 格式:ip-range
policy.ipRange.start	string	是	ip范围的起始位置 格式:ip-address
policy.ipRange.end	string	否	ip范围的终止位置 格式:ip-address
policy.ipRange.bits	uint8	否	ip掩码简写
policy.ipRange.mask	string	否	ip掩码 格式:ip-address
policy.updateTime	string	是	更新时间
policy.boardAndCountry	string	否	正确的归属地

请求样例：

修改IP归属地策略

```
PATCH https://192.168.1.1/api/v1/namespaces/public/iplocations//11.1.1.0%2f24
{
  "IPAddress": "11.1.1.0/24",
  "boardAndCountry": "非洲地区/南非"
}
```

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	IP归属地
data.uuid	string	唯一标识id

data.IPAddress	string	只支持IPv4地址，格式如下 1、单个IP地址，例如：192.168.0.1 2、IP范围，起始IP-结束IP(起始IP小于结束IP)，例如： 192.168.0.1-192.168.0.150 3、IP地址/子网掩码，例如： 192.168.0.1/255.255.255.0或192.168.0.1/24 最大长度:35
data.ipRange	object	只支持IPv4地址，格式如下 1、单个IP地址，例如：192.168.0.1 2、IP范围，起始IP-结束IP(起始IP小于结束IP)，例如： 192.168.0.1-192.168.0.150 3、IP地址/子网掩码，例如： 192.168.0.1/255.255.255.0或192.168.0.1/24 格式:ip-range
data.ipRange.start	string	ip范围的起始位置 格式:ip-address
data.ipRange.end	string	ip范围的终止位置 格式:ip-address
data.ipRange.bits	uint8	ip掩码简写
data.ipRange.mask	string	ip掩码 格式:ip-address
data.updateTime	string	更新时间
data.boardAndCountry	string	正确的归属地

请求成功样例：

```
{
  "data": {
    "updateTime": "2020-06-29 17:43",
    "IPAddress": "11.1.1.0/24",
    "boardAndCountry": "非洲地区/南非",
    "uuid": "4F4070E0C8F94612A6C15B6756F3C416",
    "ipRange": {
      "start": "11.1.1.0",
      "end": "11.1.1.255"
    }
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

2.6.1.5 删除IP归属地策略

功能说明：通过此接口，可以通过策略名，删除对应的IP归属地策略

1) API请求数据

DELETE <https://{host}/api/v1/namespaces/@namespace/iplocations/@policyname>

2) 输入参数

请求样例:

删除IP归属地策略

DELETE https://192.168.1.1/api/v1/namespaces/public/iplocations//11.1.1.0%2f24

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	IP归属地
data.uuid	string	唯一标识id
data.IPAddress	string	只支持IPv4地址, 格式如下 1、单个IP地址, 例如: 192.168.0.1 2、IP范围, 起始IP-结束IP(起始IP小于结束IP), 例如: 192.168.0.1-192.168.0.150 3、IP地址/子网掩码, 例如: 192.168.0.1/255.255.255.0或192.168.0.1/24 最大长度:35
data.ipRange	object	只支持IPv4地址, 格式如下 1、单个IP地址, 例如: 192.168.0.1 2、IP范围, 起始IP-结束IP(起始IP小于结束IP), 例如: 192.168.0.1-192.168.0.150 3、IP地址/子网掩码, 例如: 192.168.0.1/255.255.255.0或192.168.0.1/24 格式:ip-range
data.ipRange.start	string	ip范围的起始位置 格式:ip-address
data.ipRange.end	string	ip范围的终止位置 格式:ip-address
data.ipRange.bits	uint8	ip掩码简写
data.ipRange.mask	string	ip掩码 格式:ip-address
data.updateTime	string	更新时间
data.boardAndCountry	string	正确的归属地

请求成功样例:

```
{
  "data": {
    "updateTime": "2020-06-29 17:43",
    "IPAddress": "11.1.1.0/24",
    "boardAndCountry": "非洲地区/南非",
    "uuid": "4F4070E0C8F94612A6C15B6756F3C416",
    "ipRange": {
      "start": "11.1.1.0",
      "end": "11.1.1.255"
    }
  },
}
```



```

    "code": 0,
    "message": "成功"
  }

```

请求失败样例:

```

{
  "data": "",
  "code": 22,
  "message": "无效参数"
}

```

2.6.1.6 批量删除IP归属地策略

功能说明: 通过此接口, 可以通过策略名, 批量删除对应的IP归属地策略

1) API请求数据

POST

https://{host}/api/batch/v1/namespaces/@namespace/iplocations?_method=delete

2) 输入参数

• Body参数

参数	类型	是否必选	描述
polycys	object[]	是	IP归属地
polycys.IPAddress	string	是	只支持IPv4地址, 格式如下 1、单个IP地址, 例如: 192.168.0.1 2、IP范围, 起始IP-结束IP(起始IP小于结束IP), 例如: 192.168.0.1-192.168.0.150 3、IP地址/子网掩码, 例如: 192.168.0.1/255.255.255.0或192.168.0.1/24 最大长度:35
polycys.ipRange	object	是	只支持IPv4地址, 格式如下 1、单个IP地址, 例如: 192.168.0.1 2、IP范围, 起始IP-结束IP(起始IP小于结束IP), 例如: 192.168.0.1-192.168.0.150 3、IP地址/子网掩码, 例如: 192.168.0.1/255.255.255.0或192.168.0.1/24 格式:ip-range
polycys.ipRange.start	string	是	ip范围的起始位置 格式:ip-address
polycys.ipRange.end	string	否	ip范围的终止位置 格式:ip-address
polycys.ipRange.bits	uint8	否	ip掩码简写
polycys.ipRange.mask	string	否	ip掩码 格式:ip-address
polycys.updateTime	string	是	更新时间
polycys.boardAndCountry	string	否	正确的归属地

请求样例:

批量删除IP归属地策略

```
POST https://192.168.1.1/api/batch/v1/namespaces/public/iplocations?_method=DELETE
[
  {
    "IPAddress": "20.1.1.1"
  },
  {
    "IPAddress": "20.1.1.2"
  }
]
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	IP归属地
data.uuid	string	唯一标识id
data.IPAddress	string	只支持IPv4地址，格式如下 1、单个IP地址，例如：192.168.0.1 2、IP范围，起始IP-结束IP(起始IP小于结束IP)，例如： 192.168.0.1-192.168.0.150 3、IP地址/子网掩码，例如： 192.168.0.1/255.255.255.0或192.168.0.1/24 最大长度:35
data.ipRange	object	只支持IPv4地址，格式如下 1、单个IP地址，例如：192.168.0.1 2、IP范围，起始IP-结束IP(起始IP小于结束IP)，例如： 192.168.0.1-192.168.0.150 3、IP地址/子网掩码，例如： 192.168.0.1/255.255.255.0或192.168.0.1/24 格式:ip-range
data.ipRange.start	string	ip范围的起始位置 格式:ip-address
data.ipRange.end	string	ip范围的终止位置 格式:ip-address
data.ipRange.bits	uint8	ip掩码简写
data.ipRange.mask	string	ip掩码 格式:ip-address
data.updateTime	string	更新时间
data.boardAndCountry	string	正确的归属地

请求成功样例：

```
{
  "data": [
    {
      "updateTime": "2020-06-29 17:37",
      "IPAddress": "20.1.1.1",
      "boardAndCountry": "非洲地区/埃塞俄比亚",
      "uuid": "1C72AAA35BCA4DD58E9850F975846D52",
      "ipRange": {
```

```

        "start": "20.1.1.1",
        "end": "20.1.1.1"
    },
    {
        "updateTime": "2020-06-29 17:37",
        "IPAddress": "20.1.1.2",
        "boardAndCountry": "非洲地区/埃塞俄比亚",
        "uuid": "C0B727627EAC4473A3D4E3A246B8B5D6",
        "ipRange": {
            "start": "20.1.1.2",
            "end": "20.1.1.2"
        }
    }
],
"code": 0,
"message": "成功"
}

```

请求失败样例:

```

{
  "data": "",
  "code": 22,
  "message": "无效参数"
}

```

2.6.1.7 全量修改IP归属地策略

功能说明: 通过此接口, 可以通过策略名, 修改对应的IP归属地策略

1) API请求数据

PUT <https://{host}/api/v1/namespaces/@namespace/iplocations/@polycname>

2) 输入参数

• Body参数

参数	类型	是否必选	描述
policy	object	是	IP归属地
policy.IPAddress	string	是	只支持IPv4地址, 格式如下 1、单个IP地址, 例如: 192.168.0.1 2、IP范围, 起始IP-结束IP(起始IP小于结束IP), 例如: 192.168.0.1-192.168.0.150 3、IP地址/子网掩码, 例如: 192.168.0.1/255.255.255.0或192.168.0.1/24 最大长度:35
policy.ipRange	object	是	只支持IPv4地址, 格式如下 1、单个IP地址, 例如: 192.168.0.1 2、IP范围, 起始IP-结束IP(起始IP小于结束IP), 例如: 192.168.0.1-192.168.0.150 3、IP地址/子网掩码, 例如: 192.168.0.1/255.255.255.0或192.168.0.1/24 格式:ip-range
policy.ipRange.start	string	是	ip范围的起始位置 格式:ip-address

policy.ipRange.end	string	否	ip范围的终止位置 格式:ip-address
policy.ipRange.bits	uint8	否	ip掩码简写
policy.ipRange.mask	string	否	ip掩码 格式:ip-address
policy.updateTime	string	是	更新时间
policy.boardAndCountry	string	否	正确的归属地

请求样例:

全量修改IP归属地策略

```
PUT https://192.168.1.1/api/v1/namespaces/public/iplocations/test
{
  "IPAddress": "20.1.1.1",
  "boardAndCountry": "非洲地区/埃塞俄比亚"
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	IP归属地
data.uuid	string	唯一标识id
data.IPAddress	string	只支持IPv4地址, 格式如下 1、单个IP地址, 例如: 192.168.0.1 2、IP范围, 起始IP-结束IP(起始IP小于结束IP), 例如: 192.168.0.1-192.168.0.150 3、IP地址/子网掩码, 例如: 192.168.0.1/255.255.255.0或192.168.0.1/24 最大长度:35
data.ipRange	object	只支持IPv4地址, 格式如下 1、单个IP地址, 例如: 192.168.0.1 2、IP范围, 起始IP-结束IP(起始IP小于结束IP), 例如: 192.168.0.1-192.168.0.150 3、IP地址/子网掩码, 例如: 192.168.0.1/255.255.255.0或192.168.0.1/24 格式:ip-range
data.ipRange.start	string	ip范围的起始位置 格式:ip-address
data.ipRange.end	string	ip范围的终止位置 格式:ip-address
data.ipRange.bits	uint8	ip掩码简写
data.ipRange.mask	string	ip掩码 格式:ip-address
data.updateTime	string	更新时间

data.boardAndCountry	string	正确的归属地
----------------------	--------	--------

请求成功样例:

```
{
  "data": {
    "updateTime": "2020-06-29 17:49",
    "IPAddress": "20.1.1.1",
    "boardAndCountry": "非洲地区/埃塞俄比亚",
    "uuid": "0B2111EDD0044E0E85CFB23C43B9CDFE",
    "ipRange": {
      "start": "20.1.1.1",
      "end": "20.1.1.1"
    }
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

2.6.1.8 查询IP归属地

功能说明: 通过此接口, 根据IPv4地址, 查询该地址所属的国家或地区

1) API请求数据

GET <https://{host}/api/v1/namespaces/@namespace/iplocation>

2) 输入参数

- URL 参数

参数	类型	是否必选	描述
_ip	string	否	IP地址

请求样例:

查询IP归属地

GET https://192.168.1.1/api/v1/namespaces/public/iplocation?_ip=200.200.155.25

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	IP归属地查询结果
data.ip	string	IP地址
data.location	string	归属地

请求成功样例:

```
{
  "data": {
    "ip": "200.200.155.25",
    "location": "巴西"
  }
}
```

```
    },
    "code": 0,
    "message": "成功"
  }
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

2.6.1.9 查看本机地理位置

功能说明：查看用户配置的本机地理位置

1) API请求数据

GET

https://{host}/api/v1/namespaces/@namespace/attackmap/locallocation

2) 输入参数

请求样例：

查看本机地理位置

GET https://192.168.1.1/api/v1/namespaces/public/attackmap/locallocation

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	设备当前地理位置
data.board	string	大洲 最大长度:95
data.country	string	国家 最大长度:95

请求成功样例：

```
{
  "data": {
    "board": "非洲地区",
    "country": "阿尔及利亚"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

2.6.1.10 设置本机地理位置

功能说明：设置本机地理位置

1) API请求数据

PATCH https://{host}/api/v1/namespaces/@namespace/attackmap/locallocation

2) 输入参数

• Body参数

参数	类型	是否必选	描述
obj	object	是	设备当前地理位置
obj.board	string	是	大洲 最大长度:95
obj.country	string	是	国家 最大长度:95

请求样例：

设置本机地理位置

PATCH https://192.168.1.1/api/v1/namespaces/public/attackmap/locallocation

```
{
  "board": "非洲地区",
  "country": "阿尔及利亚"
}
```

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	设备当前地理位置
data.board	string	大洲 最大长度:95
data.country	string	国家 最大长度:95

请求成功样例：

```
{
  "data": {
    "board": "非洲地区",
    "country": "阿尔及利亚"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

2.6.2 国家地区列表

2.6.2.1 查看国家地区列表

功能说明：通过此接口，根据过滤条件查看国家地区

1) API请求数据

GET <https://{host}/api/v1/namespaces/@namespace/boardcountrylist>

2) 输入参数

• URL参数

参数	类型	是否必选	描述
_start	int32	否	起始位置, 从 0 开始 最小值:0 最大值:2000000
type	string	否	类型 枚举值:[BOARD:大洲 COUNTRY:国家 PROVINCE:省份]
_search	string	否	模糊搜索关键字 最大长度:95
_length	int32	否	获取最大长度, 限制最大值, 如果数据不够可能返回小于 length 最小值:1 最大值:200 默认值:true

请求样例:

查看国家地区列表

GET <https://192.168.1.1/api/v1/namespaces/public/boardcountrylist?type=BOARD>

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	分页数据
data.totalItems	int32	总共多少项目
data.totalPages	int32	总共多少页
data.pageNumber	int32	当前页码, 从 1 开始
data.pageSize	int32	每页多大
data.itemsOffset	int32	当前条目偏移, 从 0 开始
data.itemLength	int32	数据列表长度

data.items	object[]	有效数据列表
data.items.uuid	string	唯一标识ID
data.items.name	string	国家/地区全称，内容为全部[/大洲]/[国家]/[省份]， []中内容可以缺省 最大长度:128
data.items.board	string	大洲 最大长度:128
data.items.country	string	国家 最大长度:128
data.items.province	string	省份 最大长度:128
data.items.type	string	类型 枚举值:[BOARD:大洲 COUNTRY:国家 PROVINCE:省份]

请求成功样例：

```
{
  "data": {
    "itemsOffset": 0,
    "items": [
      {
        "type": "BOARD",
        "board": "非洲地区",
        "uuid": "A2A5CD1753EE410AA81D3D383FAED531",
        "name": "非洲地区"
      },
      {
        "type": "BOARD",
        "board": "亚太地区",
        "uuid": "912C7FEE2FDA4DA798C27645919C38D0",
        "name": "亚太地区"
      },
      {
        "type": "BOARD",
        "board": "北美地区",
        "uuid": "D1EE4D97893F4E86B496B88EBF963A68",
        "name": "北美地区"
      },
      {
        "type": "BOARD",
        "board": "拉美地区",
        "uuid": "6014313FE6994BFEA966996462D77666",
        "name": "拉美地区"
      },
      {
        "type": "BOARD",
        "board": "欧洲地区",
        "uuid": "43CF191AD69B4C6D8EED423A2B80D500",
        "name": "欧洲地区"
      }
    ],
    "totalItems": 5,
    "pageNumber": 1,
    "itemLength": 5,
    "totalPages": 1,
    "pageSize": 100
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

2.6.3 isp地址段

2.6.3.1 查询IP地址对应的ISP地址段名称

功能说明：查询IP地址对应的ISP地址段名称

1) API请求数据

GET `https://{host}/api/v1/namespaces/@namespace/ispinfo`

2) 输入参数

- URL 参数

参数	类型	是否必选	描述
ip	string	否	IP地址

请求样例：

查询IP地址对应的ISP，查询条件：查询IP地址172.16.200.12对应的ISP

GET `https://192.168.1.1/api/v1/namespaces/public/ispinfo?ip=172.16.200.12`

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	返回IP地址所属的IP地址库名称
data.isp_name	string	IP地址库名称 最小长度:1 最大长度:95

请求成功样例：

```
{
  "data": {
    "isp_name": "test"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

2.6.3.2 获取所有ISP地址段

功能说明：获取所有ISP地址段

1) API请求数据

GET `https://{host}/api/v1/namespaces/@namespace/isp`

2) 输入参数

- URL参数

参数	类型	是否必选	描述
_start	int32	否	起始位置, 从 0 开始 最小值:0 最大值:2000000
_length	int32	否	获取最大长度, 限制最大值, 如果数据不够可能返回小于 length 最小值:1 最大值:200 默认值:true
_select	string	否	选择字段, 字段用逗号分隔 最大长度:1000

请求样例:

获取多条ISP地址段的配置信息, 查询条件: 起始位置为0, 最多展示100条

GET `https://192.168.1.1/api/v1/namespaces/public/isp?_start=0&_length=100`

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	分页数据
data.totalItems	int32	总共多少项目
data.totalPages	int32	总共多少页
data.pageNumber	int32	当前页码, 从 1 开始
data.pageSize	int32	每页多大
data.itemsOffset	int32	当前条目偏移, 从 0 开始
data.itemLength	int32	数据列表长度
data.items	object[]	有效数据列表
data.items.uuid	string	资源唯一id
data.items.name	string	isp资源名字 最小长度:1 最大长度:95
data.items.isdefault	bool	区分是用户自定义还是系统预定义
data.items.ipentrys	object[]	ip地址、地址段、范围信息

data.items.ipentrys.start	string	ip范围的起始位置
data.items.ipentrys.end	string	ip范围的终止位置
data.items.ipentrys.bits	uint8	ip掩码简写
data.items.ipentrys.mask	string	ip掩码
data.items.whois	string[]	whois信息
data.items.hasref	bool	标识是否被引用

请求成功样例：

```
{
  "data": {
    "itemsOffset": 0,
    "items": [
      {
        "whois": [
          "test whois"
        ],
        "hasref": false,
        "isdefault": false,
        "uuid": "37C4E44C0AFA4BCAAF84F525051725BF",
        "name": "test",
        "ipentrys": [
          {
            "start": "12.6.6.6",
            "end": "12.9.6.6"
          }
        ]
      }
    ],
    "totalItems": 1,
    "pageNumber": 1,
    "itemLength": 1,
    "totalPages": 1,
    "pageSize": 100
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

2.6.3.3 获取单个ISP地址段

功能说明：获取单个ISP地址段

1) API请求数据

GET

https://{host}/api/v1/namespaces/@namespace/isp/@name

2) 输入参数

请求样例：

获取单条ISP地址段的配置信息，查询条件：查看名为test的ISP地址段

GET https://192.168.1.1/api/v1/namespaces/public/isp/test

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	isp地址段
data.uuid	string	资源唯一id
data.name	string	isp资源名字 最小长度:1 最大长度:95
data.isdefault	bool	区分是用户自定义还是系统预定义
data.ipentrys	object[]	ip地址、地址段、范围信息
data.ipentrys.start	string	ip范围的起始位置
data.ipentrys.end	string	ip范围的终止位置
data.ipentrys.bits	uint8	ip掩码简写
data.ipentrys.mask	string	ip掩码
data.whois	string[]	whois信息
data.hasref	bool	标识是否被引用

请求成功样例:

```
{
  "data": {
    "whois": [
      "test whois"
    ],
    "hasref": false,
    "isdefault": false,
    "uuid": "37C4E44C0AFA4BCAAF84F525051725BF",
    "name": "test",
    "ipentrys": [
      {
        "start": "12.6.6.6",
        "end": "12.9.6.6"
      }
    ]
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

2.6.3.4 添加单个ISP地址段

功能说明: 添加单个ISP地址段

1) API请求数据

POST https://{host}/api/v1/namespaces/@namespace/isp

2) 输入参数

• Body参数

参数	类型	是否必选	描述
obj	object	是	isp地址段
obj.name	string	是	isp资源名字 最小长度:1 最大长度:95
obj.isdefault	bool	否	区分是用户自定义还是系统预定义
obj.ipentrys	object[]	否	ip地址、地址段、范围信息
obj.ipentrys.start	string	是	ip范围的起始位置
obj.ipentrys.end	string	否	ip范围的终止位置
obj.ipentrys.bits	uint8	否	ip掩码简写
obj.ipentrys.mask	string	否	ip掩码
obj.whois	string[]	否	whois信息
obj.hasref	bool	否	标识是否被引用

请求样例:

新增单条ISP地址段

POST https://192.168.1.1/api/v1/namespaces/public/isp

```
{
  "whois": [
    "test whois"
  ],
  "name": "test",
  "ipentrys": [
    {
      "start": "12.6.6.6",
      "end": "12.6.6.12"
    }
  ]
}
```

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	isp地址段
data.uuid	string	资源唯一id
data.name	string	isp资源名字 最小长度:1 最大长度:95
data.isdefault	bool	区分是用户自定义还是系统预定义
data.ipentrys	object[]	ip地址、地址段、范围信息

data.ipentrys.start	string	ip范围的起始位置
data.ipentrys.end	string	ip范围的终止位置
data.ipentrys.bits	uint8	ip掩码简写
data.ipentrys.mask	string	ip掩码
data.whois	string[]	whois信息
data.hasref	bool	标识是否被引用

请求成功样例:

```
{
  "data": {
    "whois": [
      "test whois"
    ],
    "isdefault": false,
    "uuid": "F409D34C12BD43FF9D2556068ED0A554",
    "name": "test",
    "ipentrys": [
      {
        "start": "12.6.6.6",
        "end": "12.6.6.12"
      }
    ]
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

2.6.3.5 修改指定的ISP地址段

功能说明: 修改指定的ISP地址段

1) API请求数据

PATCH <https://{host}/api/v1/namespaces/@namespace/isp/@name>

2) 输入参数

• URL 参数

参数	类型	是否必选	描述
_arrayop	string	否	对象属性是数组的操作控制 枚举值: add:添加对应的数组元素 remove:删除对应的数组元素]

• Body参数

参数	类型	是否必选	描述
obj	object	是	isp地址段
obj.name	string	是	isp资源名字 最小长度:1 最大长度:95

obj.isdefault	bool	否	区分是用户自定义还是系统预定义
obj.ipentrys	object[]	否	ip地址、地址段、范围信息
obj.ipentrys.start	string	是	ip范围的起始位置
obj.ipentrys.end	string	否	ip范围的终止位置
obj.ipentrys.bits	uint8	否	ip掩码简写
obj.ipentrys.mask	string	否	ip掩码
obj.whois	string[]	否	whois信息
obj.hasref	bool	否	标识是否被引用

请求样例:

修改单条ISP地址段, 修改名为testother的ISP地址段, 将IP地址段设置为19.3.36.36, 其余不改

```
PATCH https://192.168.1.1/api/v1/namespaces/public/isp/testother
{
  "ipentrys": [
    {
      "start": "19.3.36.36"
    }
  ]
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	isp地址段
data.uuid	string	资源唯一id
data.name	string	isp资源名字 最小长度:1 最大长度:95
data.isdefault	bool	区分是用户自定义还是系统预定义
data.ipentrys	object[]	ip地址、地址段、范围信息
data.ipentrys.start	string	ip范围的起始位置
data.ipentrys.end	string	ip范围的终止位置
data.ipentrys.bits	uint8	ip掩码简写
data.ipentrys.mask	string	ip掩码
data.whois	string[]	whois信息
data.hasref	bool	标识是否被引用

请求成功样例:

```
{
  "data": {
    "whois": [
      "testother whois"
    ],
    "isdefault": false,
    "uuid": "89BC0AAB968C4DC7A38B877B17F64F95",
    "name": "testother",
```



```
{
  "ipentrys": [
    {
      "start": "19.3.36.36"
    }
  ],
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

2.6.3.6 全量修改指定的ISP地址段

功能说明: 全量修改指定的ISP地址段

1) API请求数据

PUT <https://{host}/api/v1/namespaces/@namespace/isp/@name>

2) 输入参数

• Body参数

参数	类型	是否必选	描述
obj	object	是	isp地址段
obj.name	string	是	isp资源名字 最小长度:1 最大长度:95
obj.isdefault	bool	否	区分是用户自定义还是系统预定义
obj.ipentrys	object[]	否	ip地址、地址段、范围信息
obj.ipentrys.start	string	是	ip范围的起始位置
obj.ipentrys.end	string	否	ip范围的终止位置
obj.ipentrys.bits	uint8	否	ip掩码简写
obj.ipentrys.mask	string	否	ip掩码
obj.whois	string[]	否	whois信息
obj.hasref	bool	否	标识是否被引用

请求样例:

全量更新一条ISP地址段, 更新名为testother的ISP地址段, 修改whois为testother whois, IP地址段置为126.6.0/24, 其余全部默认

```
PUT https://192.168.1.1/api/v1/namespaces/public/isp/testother
{
  "whois": [
    "testother whois"
  ],
  "name": "testother",
  "ipentrys": [
    {
      "start": "126.6.6.0",
      "bits": 24
    }
  ]
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	isp地址段
data.uuid	string	资源唯一id
data.name	string	isp资源名字 最小长度:1 最大长度:95
data.isdefault	bool	区分是用户自定义还是系统预定义
data.ipentrys	object[]	ip地址、地址段、范围信息
data.ipentrys.start	string	ip范围的起始位置
data.ipentrys.end	string	ip范围的终止位置
data.ipentrys.bits	uint8	ip掩码简写
data.ipentrys.mask	string	ip掩码
data.whois	string[]	whois信息
data.hasref	bool	标识是否被引用

请求成功样例:

```
{
  "data": {
    "whois": [
      "testother whois"
    ],
    "isdefault": false,
    "uuid": "89BC0AAB968C4DC7A38B877B17F64F95",
    "name": "testother",
    "ipentrys": [
      {
        "start": "126.6.6.0",
        "end": "126.6.6.255"
      }
    ]
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

2.6.3.7 删除指定的ISP地址段

功能说明: 删除指定的ISP地址段

1) API请求数据

DELETE https://{host}/api/v1/namespaces/@namespace/isp/@name

2) 输入参数

请求样例:

删除一条ISP地址段，删除名为test的ISP地址段

DELETE https://192.168.1.1/api/v1/namespaces/public/isp/test

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	isp地址段
data.uuid	string	资源唯一id
data.name	string	isp资源名字 最小长度:1 最大长度:95
data.isdefault	bool	区分是用户自定义还是系统预定义
data.ipentrys	object[]	ip地址、地址段、范围信息
data.ipentrys.start	string	ip范围的起始位置
data.ipentrys.end	string	ip范围的终止位置
data.ipentrys.bits	uint8	ip掩码简写
data.ipentrys.mask	string	ip掩码
data.whois	string[]	whois信息
data.hasref	bool	标识是否被引用

请求成功样例:

```
{
  "data": {
    "whois": [
      "test whois"
    ],
    "isdefault": false,
    "uuid": "37C4E44C0AFA4BCAAF84F525051725BF",
    "name": "test",
    "ipentrys": [
      {
        "start": "12.6.6.6",
        "end": "12.9.6.6"
      }
    ]
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

2.6.3.8 批量删除指定的ISP地址段

功能说明：批量删除指定的ISP地址段

1) API请求数据

POST https://{host}/api/batch/v1/namespaces/@namespace/isp?_method=delete

2) 输入参数

- Body参数

参数	类型	是否必选	描述
obj	object[]	是	ISP地址段列表
obj.name	string	是	isp资源名字 最小长度:1 最大长度:95
obj.isdefault	bool	否	区分是用户自定义还是系统预定义
obj.ipentrys	object[]	否	ip地址、地址段、范围信息
obj.ipentrys.start	string	是	ip范围的起始位置
obj.ipentrys.end	string	否	ip范围的终止位置
obj.ipentrys.bits	uint8	否	ip掩码简写
obj.ipentrys.mask	string	否	ip掩码
obj.whois	string[]	否	whois信息
obj.hasref	bool	否	标识是否被引用

请求样例：

删除多条ISP地址段，删除名为test和testother的ISP地址段

```
POST https://192.168.1.1/api/batch/v1/namespaces/public/isp?_method=DELETE
[
  {
    "name": "test"
  },
  {
    "name": "testother"
  }
]
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	ISP地址段列表
data.uuid	string	资源唯一id
data.name	string	isp资源名字 最小长度:1 最大长度:95
data.isdefault	bool	区分是用户自定义还是系统预定义

data.ipentrys	object[]	ip地址、地址段、范围信息
data.ipentrys.start	string	ip范围的起始位置
data.ipentrys.end	string	ip范围的终止位置
data.ipentrys.bits	uint8	ip掩码简写
data.ipentrys.mask	string	ip掩码
data.whois	string[]	whois信息
data.hasref	bool	标识是否被引用

请求成功样例：

```
{
  "data": [
    {
      "whois": [
        "testother whois"
      ],
      "isdefault": false,
      "uuid": "89BC0AAB968C4DC7A38B877B17F64F95",
      "name": "testother",
      "ipentrys": [
        {
          "start": "19.3.36.36"
        }
      ]
    },
    {
      "whois": [
        "test whois"
      ],
      "isdefault": false,
      "uuid": "F409D34C12BD43FF9D2556068ED0A554",
      "name": "test",
      "ipentrys": [
        {
          "start": "12.6.6.6",
          "end": "12.6.6.12"
        }
      ]
    }
  ],
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

2.7 时间计划

2.7.1 获取所有时间计划

功能说明：获取所有时间计划

1) API请求数据

GET

https://{host}/api/v1/namespaces/@namespace/schedules

2) 输入参数

• URL参数

参数	类型	是否必选	描述
_start	int32	否	起始位置, 从 0 开始 最小值:0 最大值:2000000
scheduleType	string	否	过滤时间计划类型 枚举值:[SINGLE:单次时间计划 PERIODICAL:循环时间计划]
_select	string	否	选择字段, 字段用逗号分隔 最大长度:1000
_length	int32	否	获取最大长度, 限制最大值, 如果数据不够可能返回小于 length 最小值:1 最大值:200 默认值:true
_sortBy	string	否	指定排序字段 最大长度:100
_order	string	否	指定排序方式 (正序/倒序) 默认值:true 枚举值:[asc:按升序排列 desc:按降序排列]

请求样例:

获取多条时间计划的配置信息, 查询条件: 起始位置为0, 最多展示100条

GET https://192.168.1.1/api/v1/namespaces/public/schedules?_start=0&_length=100

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	分页数据
data.totalItems	int32	总共多少项目
data.totalPages	int32	总共多少页
data.pageNumber	int32	当前页码, 从 1 开始
data.pageSize	int32	每页多大
data.itemsOffset	int32	当前条目偏移, 从 0 开始
data.itemLength	int32	数据列表长度
data.items	object[]	有效数据列表

data.items.uuid	string	唯一资源id
data.items.name	string	时间计划名称 最小长度:1 最大长度:95
data.items.description	string	描述字段 最小长度:0 最大长度:95
data.items.scheduleType	string	时间计划类型 枚举值:[SINGLE:单次时间计划 PERIODICAL:循环时间计划]
data.items.isdefault	bool	区分是用户自定义时间计划和系统预定义时间计划
data.items.hasref	bool	标识是否被引用
data.items.singleSchedule	object	单次时间计划规则
data.items.singleSchedule.startDateTime	string	开始日期时间
data.items.singleSchedule.endDateTime	string	结束日期时间
data.items.periodicalSchedule	object	循环时间计划规则
data.items.periodicalSchedule.singleDetail	object[]	单次循环时间段
data.items.periodicalSchedule.singleDetail.startDateTime	string	开始日期时间
data.items.periodicalSchedule.singleDetail.endDateTime	string	结束日期时间
data.items.periodicalSchedule.periodicalDetail	object[]	循环时间计划周期规则
data.items.periodicalSchedule.periodicalDetail.week	string[]	循环周期
data.items.periodicalSchedule.periodicalDetail.startTime	string	开始时间
data.items.periodicalSchedule.periodicalDetail.endTime	string	结束时间

请求成功样例:

```
{
  "data": {
    "itemsOffset": 0,
    "items": [
      {
        "hasref": false,
        "scheduleType": "SINGLE",
```

```
{
  "singleSchedule": {
    "startDateTime": "2020-06-23 10:28:00",
    "endDateTime": "2020-06-30 10:28:00"
  },
  "isdefault": false,
  "uuid": "2E9E092C85E82A5EB3AA7C5635F119A2",
  "name": "single_test",
  "description": ""
},
{
  "hasref": false,
  "scheduleType": "PERIODICAL",
  "periodicalSchedule": {
    "periodicalDetail": [
      {
        "startTime": "08:00:00",
        "endTime": "18:00:00",
        "week": [
          "MONDAY",
          "TUESDAY",
          "WEDNESDAY"
        ]
      }
    ]
  },
  "singleDetail": []
},
{
  "isdefault": false,
  "uuid": "702850A5CE50232AB33D0F8BEA0D55CE",
  "name": "periodic_test",
  "description": ""
}
],
"totalItems": 2,
"pageNumber": 1,
"itemLength": 2,
"totalPages": 1,
"pageSize": 100
},
"code": 0,
"message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

2.7.2 获取单个时间计划

功能说明：获取单个时间计划

1) API请求数据

GET <https://{host}/api/v1/namespaces/@namespace/schedules/@name>

2) 输入参数

请求样例：

获取单条时间计划的配置信息，查询条件：查看名为single_test的时间计划

GET https://192.168.1.1/api/v1/namespaces/public/schedules/single_test

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码

message	string	错误信息
data	object	时间计划
data.uuid	string	唯一资源id
data.name	string	时间计划名称 最小长度:1 最大长度:95
data.description	string	描述字段 最小长度:0 最大长度:95
data.scheduleType	string	时间计划类型 枚举值:[SINGLE:单次时间计划 PERIODICAL:循环时间计划]
data.isdefault	bool	区分是用户自定义时间计划和系统预定义时间计划
data.hasref	bool	标识是否被引用
data.singleSchedule	object	单次时间计划规则
data.singleSchedule.startDateTime	string	开始日期时间
data.singleSchedule.endDateTime	string	结束日期时间
data.periodicalSchedule	object	循环时间计划规则
data.periodicalSchedule.singleDetail	object[]	单次循环时间段
data.periodicalSchedule.singleDetail.startTime	string	开始日期时间
data.periodicalSchedule.singleDetail.endTime	string	结束日期时间
data.periodicalSchedule.periodicalDetail	object[]	循环时间计划周期规则
data.periodicalSchedule.periodicalDetail.week	string[]	循环周期
data.periodicalSchedule.periodicalDetail.startTime	string	开始时间
data.periodicalSchedule.periodicalDetail.endTime	string	结束时间

请求成功样例:

```
{
  "data": {
    "hasref": false,
    "scheduleType": "SINGLE",
    "singleSchedule": {
      "startDateTime": "2020-06-23 10:28:00",
      "endDateTime": "2020-06-30 10:28:00"
    }
  }
}
```

```
{
  "isdefault": false,
  "uuid": "2E9E092C85E84A5EB3AA7C5635F119A2",
  "name": "single_test",
  "description": ""
},
{
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

2.7.3 添加单个时间计划

功能说明: 添加单个时间计划

1) API请求数据

POST <https://{host}/api/v1/namespaces/@namespace/schedules>

2) 输入参数

• Body参数

参数	类型	是否必选	描述
obj	object	是	时间计划
obj.name	string	是	时间计划名称 最小长度:1 最大长度:95
obj.description	string	否	描述字段 最小长度:0 最大长度:95
obj.scheduleType	string	是	时间计划类型 枚举值:[SINGLE:单次时间计划 PERIODICAL:循环时间计划]
obj.isdefault	bool	否	区分是用户自定义时间计划和系统预定义时间计划
obj.hasref	bool	否	标识是否被引用
obj.singleSchedule	object	否	单次时间计划规则
obj.singleSchedule.startDateTime	string	是	开始日期时间
obj.singleSchedule.endDateTime	string	是	结束日期时间
obj.periodicalSchedule	object	否	循环时间计划规则
obj.periodicalSchedule.singleDetail	object[]	否	单次循环时间段

obj.periodicalSchedule.singleDetail.startDateTime	string	是	开始日期时间
obj.periodicalSchedule.singleDetail.endDateTime	string	是	结束日期时间
obj.periodicalSchedule.periodicalDetail	object[]	否	循环时间计划周期规则
obj.periodicalSchedule.periodicalDetail.week	string[]	是	循环周期
obj.periodicalSchedule.periodicalDetail.startTime	string	是	开始时间
obj.periodicalSchedule.periodicalDetail.endTime	string	是	结束时间

请求样例:

新增单条循环时间计划

POST https://192.168.1.1/api/v1/namespaces/public/schedules

```
{
  "periodicalSchedule": {
    "periodicalDetail": [
      {
        "startTime": "08:00:00",
        "endTime": "18:00:00",
        "week": [
          "MONDAY",
          "TUESDAY",
          "WEDNESDAY"
        ]
      }
    ],
    "singleDetail": []
  },
  "scheduleType": "PERIODICAL",
  "isdefault": false,
  "name": "periodic_test",
  "description": ""
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	时间计划
data.uuid	string	唯一资源id
data.name	string	时间计划名称 最小长度:1 最大长度:95
data.description	string	描述字段 最小长度:0 最大长度:95

data.scheduleType	string	时间计划类型 枚举值:[SINGLE:单次时间计划 PERIODICAL:循环时间计划]
data.isdefault	bool	区分是用户自定义时间计划和系统预定义时间计划
data.hasref	bool	标识是否被引用
data.singleSchedule	object	单次时间计划规则
data.singleSchedule.startDateTime	string	开始日期时间
data.singleSchedule.endDateTime	string	结束日期时间
data.periodicalSchedule	object	循环时间计划规则
data.periodicalSchedule.singleDetail	object[]	单次循环时间段
data.periodicalSchedule.singleDetail.startTime	string	开始日期时间
data.periodicalSchedule.singleDetail.endTime	string	结束日期时间
data.periodicalSchedule.periodicalDetail	object[]	循环时间计划周期规则
data.periodicalSchedule.periodicalDetail.week	string[]	循环周期
data.periodicalSchedule.periodicalDetail.startTime	string	开始时间
data.periodicalSchedule.periodicalDetail.endTime	string	结束时间

请求成功样例:

```
{
  "data": {
    "periodicalSchedule": {
      "periodicalDetail": [
        {
          "startTime": "08:00:00",
          "endTime": "18:00:00",
          "week": [
            "MONDAY",
            "TUESDAY",
            "WEDNESDAY"
          ]
        }
      ]
    },
    "singleDetail": []
  },
  "scheduleType": "PERIODICAL",
  "isdefault": false,
  "uuid": "702850A5CE50432AB33D0F8BEA0D55CE",
  "name": "periodic_test",
  "description": ""
},
```

```

    "code": 0,
    "message": "成功"
  }

```

请求失败样例:

```

{
  "data": "",
  "code": 22,
  "message": "无效参数"
}

```

2.7.4 修改指定的时间计划

功能说明: 修改指定的时间计划

1) API请求数据

PATCH <https://{host}/api/v1/namespaces/@namespace/schedules/@name>

2) 输入参数

• URL参数

参数	类型	是否必选	描述
_arrayop	string	否	对象属性是数组的操作控制 枚举值: add:添加对应的数组元素 remove:删除对应的数组元素]

• Body参数

参数	类型	是否必选	描述
obj	object	是	时间计划
obj.name	string	是	时间计划名称 最小长度:1 最大长度:95
obj.description	string	否	描述字段 最小长度:0 最大长度:95
obj.scheduleType	string	是	时间计划类型 枚举值: SINGLE:单次时间计划 PERIODICAL:循环时间计划]
obj.isdefault	bool	否	区分是用户自定义时间计划和系统预定义时间计划
obj.hasref	bool	否	标识是否被引用
obj.singleSchedule	object	否	单次时间计划规则
obj.singleSchedule.startDateTime	string	是	开始日期时间
obj.singleSchedule.endDateTime	string	是	结束日期时间
obj.periodicalSchedule	object	否	循环时间计划规则

obj.periodicalSchedule.singleDetail	object[]	否	单次循环时间段
obj.periodicalSchedule.singleDetail.startDateTime	string	是	开始日期时间
obj.periodicalSchedule.singleDetail.endDateTime	string	是	结束日期时间
obj.periodicalSchedule.periodicalDetail	object[]	否	循环时间计划周期规则
obj.periodicalSchedule.periodicalDetail.week	string[]	是	循环周期
obj.periodicalSchedule.periodicalDetail.startTime	string	是	开始时间
obj.periodicalSchedule.periodicalDetail.endTime	string	是	结束时间

请求样例:

修改单条循环时间计划, 修改名为periodic_test的循环时间计划, 起止时间段为2020-09-01 10:41:51到2020-09-30 10:41:51, 其余不改

```
PATCH https://192.168.1.1/api/v1/namespaces/public/schedules/periodic_test
{
  "periodicalSchedule": {
    "singleDetail": [
      {
        "startDateTime": "2020-09-01 00:00:00",
        "endDateTime": "2020-09-30 10:00:00"
      }
    ]
  }
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	时间计划
data.uuid	string	唯一资源id
data.name	string	时间计划名称 最小长度:1 最大长度:95
data.description	string	描述字段 最小长度:0 最大长度:95

data.scheduleType	string	时间计划类型 枚举值:[SINGLE:单次时间计划 PERIODICAL:循环时间计划]
data.isdefault	bool	区分是用户自定义时间计划和系统预定义时间计划
data.hasref	bool	标识是否被引用
data.singleSchedule	object	单次时间计划规则
data.singleSchedule.startDateTime	string	开始日期时间
data.singleSchedule.endDateTime	string	结束日期时间
data.periodicalSchedule	object	循环时间计划规则
data.periodicalSchedule.singleDetail	object[]	单次循环时间段
data.periodicalSchedule.singleDetail.startTime	string	开始日期时间
data.periodicalSchedule.singleDetail.endTime	string	结束日期时间
data.periodicalSchedule.periodicalDetail	object[]	循环时间计划周期规则
data.periodicalSchedule.periodicalDetail.week	string[]	循环周期
data.periodicalSchedule.periodicalDetail.startTime	string	开始时间
data.periodicalSchedule.periodicalDetail.endTime	string	结束时间

请求成功样例:

```
{
  "data": {
    "periodicalSchedule": {
      "periodicalDetail": [
        {
          "startTime": "08:00:00",
          "endTime": "18:00:00",
          "week": [
            "MONDAY",
            "TUESDAY",
            "WEDNESDAY"
          ]
        }
      ],
      "singleDetail": [
        {
          "startDateTime": "2020-09-01 00:00:00",
          "endDateTime": "2020-09-30 10:00:00"
        }
      ]
    },
    "scheduleType": "PERIODICAL",
  }
}
```

```

    "isdefault": false,
    "uuid": "702850A5CE50432AB33D0F8BEA0D55CE",
    "name": "periodic_test",
    "description": ""
  },
  "code": 0,
  "message": "成功"
}

```

请求失败样例:

```

{
  "data": "",
  "code": 22,
  "message": "无效参数"
}

```

2.7.5 全量修改指定的时间计划

功能说明: 全量修改指定的时间计划

1) API请求数据

PUT <https://{host}/api/v1/namespaces/@namespace/schedules/@name>

2) 输入参数

- Body参数

参数	类型	是否必选	描述
obj	object	是	时间计划
obj.name	string	是	时间计划名称 最小长度:1 最大长度:95
obj.description	string	否	描述字段 最小长度:0 最大长度:95
obj.scheduleType	string	是	时间计划类型 枚举值:[SINGLE:单次时间计划 PERIODICAL:循环时间计划]
obj.isdefault	bool	否	区分是用户自定义时间计划和系统预定义时间计划
obj.hasref	bool	否	标识是否被引用
obj.singleSchedule	object	否	单次时间计划规则
obj.singleSchedule.startDateTime	string	是	开始日期时间
obj.singleSchedule.endDateTime	string	是	结束日期时间
obj.periodicalSchedule	object	否	循环时间计划规则
obj.periodicalSchedule.singleDetail	object[]	否	单次循环时间段
obj.periodicalSchedule.singleDetail.startDateTime	string	是	开始日期时间

obj.periodicalSchedule.singleDetail.endTime	string	是	结束日期时间
obj.periodicalSchedule.periodicalDetail	object[]	否	循环时间计划周期规则
obj.periodicalSchedule.periodicalDetail.week	string[]	是	循环周期
obj.periodicalSchedule.periodicalDetail.startTime	string	是	开始时间
obj.periodicalSchedule.periodicalDetail.endTime	string	是	结束时间

请求样例:

全量更新一条时间计划,更新名为single_test的单次时间计划,开始时间日期为2020-07-10 16:00:00,结束时间日期为2020-07-11 17:00:00,其余全部默认

```
PUT https://192.168.1.1/api/v1/namespaces/public/schedules/single_test
{
  "singleSchedule": {
    "startTime": "2020-07-10 16:00:00",
    "endTime": "2020-07-11 17:00:00"
  },
  "name": "single_test",
  "scheduleType": "SINGLE"
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	时间计划
data.uuid	string	唯一资源id
data.name	string	时间计划名称 最小长度:1 最大长度:95
data.description	string	描述字段 最小长度:0 最大长度:95
data.scheduleType	string	时间计划类型 枚举值: SINGLE:单次时间计划 PERIODICAL:循环时间计划]
data.isdefault	bool	区分是用户自定义时间计划和系统预定义时间计划
data.hasref	bool	标识是否被引用
data.singleSchedule	object	单次时间计划规则

data.singleSchedule.startDateTime	string	开始日期时间
data.singleSchedule.endDateTime	string	结束日期时间
data.periodicalSchedule	object	循环时间计划规则
data.periodicalSchedule.singleDetail	object[]	单次循环时间段
data.periodicalSchedule.singleDetail.startDateTime	string	开始日期时间
data.periodicalSchedule.singleDetail.endDateTime	string	结束日期时间
data.periodicalSchedule.periodicalDetail	object[]	循环时间计划周期规则
data.periodicalSchedule.periodicalDetail.week	string[]	循环周期
data.periodicalSchedule.periodicalDetail.startTime	string	开始时间
data.periodicalSchedule.periodicalDetail.endTime	string	结束时间

请求成功样例:

```
{
  "data": {
    "scheduleType": "SINGLE",
    "isdefault": false,
    "singleSchedule": {
      "startDateTime": "2020-07-10 16:00:00",
      "endDateTime": "2020-07-11 17:00:00"
    },
    "uuid": "2E9E092C85E84A5EB3AA7C5635F119A2",
    "name": "single_test"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

2.7.6 删除指定的时间计划

功能说明: 删除指定的时间计划

1) API请求数据

DELETE https://{host}/api/v1/namespaces/@namespace/schedules/@name

2) 输入参数

请求样例:

删除一条时间计划，删除名为single_test的时间计划

DELETE https://192.168.1.1/api/v1/namespaces/public/schedules/single_test

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	时间计划
data.uuid	string	唯一资源id
data.name	string	时间计划名称 最小长度:1 最大长度:95
data.description	string	描述字段 最小长度:0 最大长度:95
data.scheduleType	string	时间计划类型 枚举值:[SINGLE:单次时间计划 PERIODICAL:循环时间计划]
data.isdefault	bool	区分是用户自定义时间计划和系统预定义时间计划
data.hasref	bool	标识是否被引用
data.singleSchedule	object	单次时间计划规则
data.singleSchedule.startDateTime	string	开始日期时间
data.singleSchedule.endDateTime	string	结束日期时间
data.periodicalSchedule	object	循环时间计划规则
data.periodicalSchedule.singleDetail	object[]	单次循环时间段
data.periodicalSchedule.singleDetail.startTime	string	开始日期时间
data.periodicalSchedule.singleDetail.endTime	string	结束日期时间
data.periodicalSchedule.periodicalDetail	object[]	循环时间计划周期规则
data.periodicalSchedule.periodicalDetail.week	string[]	循环周期

data.periodicalSchedule .periodicalDetail.startTime	string	开始时间
data.periodicalSchedule .periodicalDetail.endTime	string	结束时间

请求成功样例:

```
{
  "data": {
    "scheduleType": "SINGLE",
    "isdefault": false,
    "singleSchedule": {
      "startDateTime": "2020-07-10 16:00:00",
      "endDateTime": "2020-07-11 17:00:00"
    },
    "uuid": "2E9E092C85E84A5EB3AA7C5635F119A2",
    "name": "single_test"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

2.7.7 批量删除指定的时间计划

功能说明: 批量删除指定的时间计划

1) API请求数据

POST

https://{host}/api/batch/v1/namespaces/@namespace/schedules?_method=delete

2) 输入参数

• Body参数

参数	类型	是否必选	描述
obj	object[]	是	时间计划列表
obj.name	string	是	时间计划名称 最小长度:1 最大长度:95
obj.description	string	否	描述字段 最小长度:0 最大长度:95
obj.scheduleType	string	是	时间计划类型 枚举值:[SINGLE:单次时间计划 PERIODICAL:循环时间计划]
obj.isdefault	bool	否	区分是用户自定义时间计划和系统预定义时间计划

obj.hasref	bool	否	标识是否被引用
obj.singleSchedule	object	否	单次时间计划规则
obj.singleSchedule.start DateTime	string	是	开始日期时间
obj.singleSchedule.end DateTime	string	是	结束日期时间
obj.periodicalSchedule	object	否	循环时间计划规则
obj.periodicalSchedule. singleDetail	object[]	否	单次循环时间段
obj.periodicalSchedule. singleDetail.startDateTi me	string	是	开始日期时间
obj.periodicalSchedule. singleDetail.endDateTi me	string	是	结束日期时间
obj.periodicalSchedule. periodicalDetail	object[]	否	循环时间计划周期规则
obj.periodicalSchedule. periodicalDetail.week	string[]	是	循环周期
obj.periodicalSchedule. periodicalDetail.startTim e	string	是	开始时间
obj.periodicalSchedule. periodicalDetail.endTim e	string	是	结束时间

请求样例:

删除多条时间计划, 删除名为single_test和periodic_test的时间计划

```
POST https://192.168.1.1/api/batch/v1/namespaces/public/schedules?_method=DELETE
[
  {
    "name": "single_test"
  },
  {
    "name": "periodic_test"
  }
]
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	时间计划列表
data.uuid	string	唯一资源id
data.name	string	时间计划名称 最小长度:1 最大长度:95

data.description	string	描述字段 最小长度:0 最大长度:95
data.scheduleType	string	时间计划类型 枚举值:[SINGLE:单次时间计划 PERIODICAL:循环时间计划]
data.isdefault	bool	区分是用户自定义时间计划和系统预定义时间计划
data.hasref	bool	标识是否被引用
data.singleSchedule	object	单次时间计划规则
data.singleSchedule.startDateTime	string	开始日期时间
data.singleSchedule.endDateTime	string	结束日期时间
data.periodicalSchedule	object	循环时间计划规则
data.periodicalSchedule.singleDetail	object[]	单次循环时间段
data.periodicalSchedule.singleDetail.startTime	string	开始日期时间
data.periodicalSchedule.singleDetail.endTime	string	结束日期时间
data.periodicalSchedule.periodicalDetail	object[]	循环时间计划周期规则
data.periodicalSchedule.periodicalDetail.week	string[]	循环周期
data.periodicalSchedule.periodicalDetail.startTime	string	开始时间
data.periodicalSchedule.periodicalDetail.endTime	string	结束时间

请求成功样例:

```
{
  "data": [
    {
      "scheduleType": "SINGLE",
      "isdefault": false,
      "singleSchedule": {
        "startDateTime": "2020-07-10 16:00:00",
        "endDateTime": "2020-07-11 17:00:00"
      },
      "uuid": "8A2B7A37E1D3441F812ED0E5AF883BB1",
      "name": "single_test"
    },
    {
      "periodicalSchedule": {
        "periodicalDetail": [
          {
            "startTime": "08:00:00",
            "endTime": "18:00:00",

```

```
        "week": [
          "MONDAY",
          "TUESDAY",
          "WEDNESDAY"
        ]
      },
      "singleDetail": [
        {
          "startDateTime": "2020-09-01 00:00:00",
          "endDateTime": "2020-09-30 10:00:00"
        }
      ],
      "scheduleType": "PERIODICAL",
      "isdefault": false,
      "uuid": "702850A5CE50432AB33D0F8BEA0D55CE",
      "name": "periodic_test",
      "description": ""
    }
  ],
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3 网络

3.1 区域

3.1.1 获取区域信息列表

功能说明：获取区域信息列表

1) API请求数据

GET `https://{host}/api/v1/namespaces/@namespace/zones`

2) 输入参数

• URL参数

参数	类型	是否必选	描述
_start	int32	否	起始位置, 从 0 开始 最小值:0 最大值:2000000
_length	int32	否	获取最大长度, 限制最大值, 如果数据不够可能返回小于 length 最小值:1 最大值:200 默认值:true
_order	string	否	指定排序方式 (正序/倒序) 默认值:true 枚举值:[asc:按升序排列 desc:按降序排列]
_search	string	否	模糊搜索关键字 最大长度:95
_select	string	否	选择字段, 字段用逗号分隔 最大长度:1000
_sortBy	string	否	指定排序字段 最大长度:100
forwardType	string	否	过滤参数: 转发类型 枚举值:[VIRTUAL:虚拟网线类型 SWITCH:二层转发类型 ROUTE:三层转发类型]
itf	string	否	过滤参数: 接口名称
mirror	bool	否	过滤参数: 是否为内网区域

请求样例:

获取多个区域配置信息

GET `https://192.168.1.1/api/v1/namespaces/public/zones`

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	分页数据
data.totalItems	int32	总共多少项目
data.totalPages	int32	总共多少页
data.pageNumber	int32	当前页码，从 1 开始
data.pageSize	int32	每页多大
data.itemsOffset	int32	当前条目偏移，从 0 开始
data.itemLength	int32	数据列表长度
data.items	object[]	有效数据列表
data.items.uuid	string	唯一标识id
data.items.name	string	区域名称 最小长度:1 最大长度:95
data.items.forwardType	string	转发类型 枚举值:[VIRTUAL:虚拟网线类型 SWITCH:二层转发类型 ROUTE:三层转发类型]
data.items.interfaces	string[]	已选择的网络接口
data.items.hasref	bool	标识是否被引用

请求成功样例：

```
{
  "data": {
    "itemsOffset": 0,
    "items": [
      {
        "hasref": false,
        "interfaces": [
          "eth0"
        ],
        "forwardType": "ROUTE",
        "uuid": "6718BC6A446845DD963AF075A9396812",
        "name": "manager"
      },
      {
        "hasref": false,
        "interfaces": [],
        "forwardType": "SWITCH",
        "uuid": "0000000000000000000000000900000001",
        "name": "L2_trust_A"
      },
      {
        "hasref": false,
        "interfaces": [],
        "forwardType": "SWITCH",
        "uuid": "0000000000000000000000000900000002",
        "name": "L2_trust_B"
      }
    ]
  }
}
```

```
{
  "hasref": false,
  "interfaces": [],
  "forwardType": "SWITCH",
  "uuid": "0000000000000000000000000900000003",
  "name": "L2_untrust_A"
},
{
  "hasref": false,
  "interfaces": [],
  "forwardType": "SWITCH",
  "uuid": "0000000000000000000000000900000004",
  "name": "L2_untrust_B"
},
{
  "hasref": false,
  "interfaces": [],
  "forwardType": "ROUTE",
  "uuid": "0000000000000000000000000900000009",
  "name": "L3_manage"
},
{
  "hasref": false,
  "interfaces": [],
  "forwardType": "ROUTE",
  "uuid": "000000000000000000000000090000000a",
  "name": "L3_trust_A"
},
{
  "hasref": false,
  "interfaces": [],
  "forwardType": "ROUTE",
  "uuid": "000000000000000000000000090000000b",
  "name": "L3_trust_B"
},
{
  "hasref": false,
  "interfaces": [],
  "forwardType": "ROUTE",
  "uuid": "000000000000000000000000090000000c",
  "name": "L3_trust_C"
},
{
  "hasref": false,
  "interfaces": [],
  "forwardType": "ROUTE",
  "uuid": "000000000000000000000000090000000d",
  "name": "L3_untrust_A"
},
{
  "hasref": false,
  "interfaces": [],
  "forwardType": "ROUTE",
  "uuid": "000000000000000000000000090000000e",
  "name": "L3_untrust_B"
},
{
  "hasref": false,
  "interfaces": [],
  "forwardType": "ROUTE",
  "uuid": "000000000000000000000000090000000f",
  "name": "L3_untrust_C"
},
{
  "hasref": false,
  "interfaces": [],
  "forwardType": "VIRTUAL",
  "uuid": "0000000000000000000000000900000005",
  "name": "Virtual_trust_A"
},
{
  "hasref": false,
  "interfaces": [],
  "forwardType": "VIRTUAL",
  "uuid": "0000000000000000000000000900000006",
  "name": "Virtual_trust_B"
},
}
```

```
{
  "hasref": false,
  "interfaces": [],
  "forwardType": "VIRTUAL",
  "uuid": "0000000000000000000000000900000007",
  "name": "Virtual_untrust_A"
},
{
  "hasref": false,
  "interfaces": [],
  "forwardType": "VIRTUAL",
  "uuid": "0000000000000000000000000900000008",
  "name": "Virtual_untrust_B"
}
],
"totalItems": 16,
"pageNumber": 1,
"itemLength": 16,
"totalPages": 1,
"pageSize": 100
},
"code": 0,
"message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3.1.2 获取单个区域信息

功能说明: 获取单个区域信息

1) API请求数据

GET <https://{host}/api/v1/namespaces/@namespace/zones/@name>

2) 输入参数

- URL 参数

参数	类型	是否必选	描述
_select	string	否	选择字段, 字段用逗号分隔 最大长度:1000

请求样例:

获取单个区域的配置信息, 查询条件: 查看名为manager的区域

GET <https://192.168.1.1/api/v1/namespaces/public/zones/manager>

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	区域配置
data.uuid	string	唯一标识id

data.name	string	区域名称 最小长度:1 最大长度:95
data.forwardType	string	转发类型 枚举值:[VIRTUAL:虚拟网线类型 SWITCH:二层转发类型 ROUTE:三层转发类型]
data.interfaces	string[]	已选择的网络接口
data.hasref	bool	标识是否被引用

请求成功样例:

```
{
  "data": {
    "hasref": false,
    "interfaces": [
      "eth0"
    ],
    "forwardType": "ROUTE",
    "uuid": "6718BC6A446845DD963AF075A9396812",
    "name": "manager"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3.1.3 添加区域信息

功能说明: 添加区域信息

1) API请求数据

POST https://{host}/api/v1/namespaces/@namespace/zones

2) 输入参数

- Body参数

参数	类型	是否必选	描述
obj	object	是	区域配置
obj.name	string	是	区域名称 最小长度:1 最大长度:95
obj.forwardType	string	否	转发类型 默认值:true 枚举值:[VIRTUAL:虚拟网线类型 SWITCH:二层转发类型 ROUTE:三层转发类型]

obj.interfaces	string[]	是	已选择的网络接口
obj.hasref	bool	否	标识是否被引用

请求样例:

新增单个区域信息

```
POST https://192.168.1.1/api/v1/namespaces/public/zones
{
  "forwardType": "ROUTE",
  "name": "test",
  "interfaces": []
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	区域配置
data.uuid	string	唯一标识id
data.name	string	区域名称 最小长度:1 最大长度:95
data.forwardType	string	转发类型 枚举值:[VIRTUAL:虚拟网线类型 SWITCH:二层转发类型 ROUTE:三层转发类型]
data.interfaces	string[]	已选择的网络接口
data.hasref	bool	标识是否被引用

请求成功样例:

```
{
  "data": {
    "interfaces": [],
    "forwardType": "ROUTE",
    "uuid": "539301B925904B0898FC5BB0AE6E97B4",
    "name": "test"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3.1.4 删除指定区域信息

功能说明：删除指定区域信息

1) API请求数据

DELETE https://{host}/api/v1/namespaces/@namespace/zones/@name

2) 输入参数

请求样例：

删除一个区域，删除名为test的区域

DELETE https://192.168.1.1/api/v1/namespaces/public/zones/test

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	区域配置
data.uuid	string	唯一标识id
data.name	string	区域名称 最小长度:1 最大长度:95
data.forwardType	string	转发类型 枚举值:[VIRTUAL:虚拟网线类型 SWITCH:二层转发类型 ROUTE:三层转发类型]
data.interfaces	string[]	已选择的网络接口
data.hasref	bool	标识是否被引用

请求成功样例：

```
{
  "data": {
    "interfaces": [],
    "forwardType": "SWITCH",
    "uuid": "539301B925904B0898FC5BB0AE6E97B4",
    "name": "test"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3.1.5 批量删除区域信息

功能说明：批量删除区域信息

1) API请求数据

POST

https://{host}/api/batch/v1/namespaces/@namespace/zones?_method=delete

2) 输入参数

• Body参数

参数	类型	是否必选	描述
obj	object[]	是	区域列表
obj.name	string	是	区域名称 最小长度:1 最大长度:95
obj.forwardType	string	否	转发类型 默认值:true 枚举值:[VIRTUAL:虚拟网线类型 SWITCH:二层转发类型 ROUTE:三层转发类型]
obj.interfaces	string[]	是	已选择的网络接口
obj.hasref	bool	否	标识是否被引用

请求样例：

批量删除区域数据

```
POST https://192.168.1.1/api/batch/v1/namespaces/public/zones?_method=DELETE
[
  {
    "name": "L3_untrust_B"
  },
  {
    "name": "L3_untrust_C"
  }
]
```

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	区域列表
data.uuid	string	唯一标识id
data.name	string	区域名称 最小长度:1 最大长度:95

data.forwardType	string	转发类型 枚举值:[VIRTUAL:虚拟网线类型 SWITCH:二层转发类型 ROUTE:三层转发类型]
data.interfaces	string[]	已选择的网络接口
data.hasref	bool	标识是否被引用

请求成功样例:

```
{
  "data": [
    {
      "interfaces": [],
      "forwardType": "ROUTE",
      "uuid": "000000000000000000000000090000000e",
      "name": "L3_untrust_B"
    },
    {
      "interfaces": [],
      "forwardType": "ROUTE",
      "uuid": "000000000000000000000000090000000f",
      "name": "L3_untrust_C"
    }
  ],
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3.1.6 全量修改指定区域信息

功能说明: 全量修改指定区域信息

1) API请求数据

PUT https://{host}/api/v1/namespaces/@namespace/zones/@name

2) 输入参数

- Body参数

参数	类型	是否必选	描述
obj	object	是	区域配置
obj.name	string	是	区域名称 最小长度:1 最大长度:95
obj.forwardType	string	否	转发类型 默认值:true 枚举值:[VIRTUAL:虚拟网线类型 SWITCH:二层转发类型 ROUTE:三层转发类型]

obj.interfaces	string[]	是	已选择的网络接口
obj.hasref	bool	否	标识是否被引用

请求样例:

全量更新一个区域，更新名为test的区域，修改转发类型为二层转发，选择eth3接口

```
PUT https://192.168.1.1/api/v1/namespaces/public/zones/test
{
  "forwardType": "SWITCH",
  "name": "test",
  "interfaces": [
    "eth3"
  ]
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	区域配置
data.uuid	string	唯一标识id
data.name	string	区域名称 最小长度:1 最大长度:95
data.forwardType	string	转发类型 枚举值:[VIRTUAL:虚拟网线类型 SWITCH:二层转发类型 ROUTE:三层转发类型]
data.interfaces	string[]	已选择的网络接口
data.hasref	bool	标识是否被引用

请求成功样例:

```
{
  "data": {
    "interfaces": [
      "eth3"
    ],
    "forwardType": "SWITCH",
    "uuid": "539301B925904B0898FC5BB0AE6E97B4",
    "name": "test"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3.1.7 增量修改指定区域信息

功能说明：增量修改指定区域信息

1) API请求数据

PATCH https://{host}/api/v1/namespaces/@namespace/zones/@name

2) 输入参数

• URL参数

参数	类型	是否必选	描述
_arrayop	string	否	对象属性是数组的操作控制 枚举值:[add:添加对应的数组元素 remove:删除对应的数组元素]

• Body参数

参数	类型	是否必选	描述
obj	object	是	区域配置
obj.name	string	是	区域名称 最小长度:1 最大长度:95
obj.forwardType	string	否	转发类型 默认值:true 枚举值:[VIRTUAL:虚拟网线类型 SWITCH:二层转发类型 ROUTE:三层转发类型]
obj.interfaces	string[]	是	已选择的网络接口
obj.hasref	bool	否	标识是否被引用

请求样例：

修改单个区域，修改区域所选的接口为空

```
PATCH https://192.168.1.1/api/v1/namespaces/public/zones/test
{
  "interfaces": []
}
```

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	区域配置
data.uuid	string	唯一标识id

data.name	string	区域名称 最小长度:1 最大长度:95
data.forwardType	string	转发类型 枚举值:[VIRTUAL:虚拟网线类型 SWITCH:二层转发类型 ROUTE:三层转发类型]
data.interfaces	string[]	已选择的网络接口
data.hasref	bool	标识是否被引用

请求成功样例:

```
{
  "data": {
    "interfaces": [],
    "forwardType": "SWITCH",
    "uuid": "539301B925904B0898FC5BB0AE6E97B4",
    "name": "test"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3.1.8 批量修改区域信息

功能说明: 批量修改区域信息

1) API请求数据

PATCH <https://{host}/api/batch/v1/namespaces/@namespace/zones>

2) 输入参数

• URL 参数

参数	类型	是否必选	描述
_arrayop	string	否	对象属性是数组的操作控制 枚举值:[add:添加对应的数组元素 remove:删除对应的数组元素]

• Body参数

参数	类型	是否必选	描述
obj	object[]	是	区域列表
obj.name	string	是	区域名称 最小长度:1 最大长度:95

obj.forwardType	string	否	转发类型 默认值:true 枚举值:[VIRTUAL:虚拟网线类型 SWITCH:二层转发类型 ROUTE:三层转发类型]
obj.interfaces	string[]	是	已选择的网络接口
obj.hasref	bool	否	标识是否被引用

请求样例:

增量更新多个区域的配置

PATCH https://192.168.1.1/api/batch/v1/namespaces/public/zones

```
[
  {
    "name": "test1",
    "forwardType": "SWITCH"
  },
  {
    "name": "test2",
    "forwardType": "VIRTUAL"
  }
]
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	区域列表
data.uuid	string	唯一标识id
data.name	string	区域名称 最小长度:1 最大长度:95
data.forwardType	string	转发类型 枚举值:[VIRTUAL:虚拟网线类型 SWITCH:二层转发类型 ROUTE:三层转发类型]
data.interfaces	string[]	已选择的网络接口
data.hasref	bool	标识是否被引用

请求成功样例:

```
{
  "data": [
    {
      "interfaces": [],
      "forwardType": "SWITCH",
      "uuid": "C4A1A81DE0344AB1AE4E299A56A3746B",
      "name": "test1"
    },
    {
      "interfaces": [],
```

```
        "forwardType": "VIRTUAL",
        "uuid": "A399C21BD8F04E70B2F379DD8BBAE8C1",
        "name": "test2"
    },
    "code": 0,
    "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3.2 接口

3.2.1 获取接口列表

功能说明: 获取接口列表

1) API请求数据

GET

https://{host}/api/v1/namespaces/@namespace/interfaces

2) 输入参数

- URL 参数

参数	类型	是否必选	描述
_start	int32	否	起始位置, 从 0 开始 最小值:0 最大值:2000000
optionalItf	bool	否	接口列表, 去除被聚合口选择的物理口
zoneItf	bool	否	获取区域可选择的接口
routeItf	bool	否	获取所有三层接口
ifMode	string	否	过滤参数: 接口模式, 用于物理口, 聚合口 枚举值:[ROUTE:路由类型, 接口作为3层接口, 提供IP地址配置, 路由配置转发等功能 SWITCH:交换类型, 接口作为2层接口, 提供二层转发、vlan配置等功能 VIRTUALLINE:虚拟线类型, 接口作为虚拟线的一端, 透明传输报文 BYPASSMIRROR:旁路镜像类型, 该模式进来的报文, 只处理不转发]
_length	int32	否	获取最大长度, 限制最大值, 如果数据不够可能返回小于 length 最小值:1 最大值:200 默认值:true
getChannel	string	否	获取该接口所属于的聚合口的配置
_search	string	否	模糊搜索关键字 最大长度:95

_select	string	否	选择字段, 字段用逗号分隔 最大长度:1000
selectChannel	bool	否	聚合口选择物理口时,对物理口进行过滤
_sortby	string	否	指定排序字段 最大长度:100
_order	string	否	指定排序方式 (正序/倒序) 默认值:true 枚举值:[asc:按升序排列 desc:按降序排列]
ifType	string	否	过滤参数: 接口类型 枚举值:[PHYSICALIF:物理口 SUBIF:子接口 CHANNELIF:汇聚口 VLANIF:vlan对应的3层接口]

请求样例:

获取多个接口配置信息, 查询条件: 起始位置为0, 最多展示100条

GET https://192.168.1.1/api/v1/namespaces/public/interfaces?_start=0&_length=100

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	分页数据
data.totalItems	int32	总共多少项目
data.totalPages	int32	总共多少页
data.pageNumber	int32	当前页码, 从 1 开始
data.pageSize	int32	每页多大
data.itemsOffset	int32	当前条目偏移, 从 0 开始
data.itemLength	int32	数据列表长度
data.items	object[]	有效数据列表
data.items.uuid	string	唯一标识id
data.items.name	string	接口名称
data.items.description	string	接口描述 最小长度:0 最大长度:95
data.items.mtu	int32	最大传输单元 最小值:68 最大值:1796

data.items.ifType	string	表示接口类型 枚举值:[PHYSICALIF:物理口 SUBIF:子接口 CHANNELIF:汇聚口 VLANIF:vlan对应的3层接口]
data.items.mac	string	接口的mac地址
data.items.shutdown	bool	接口的启用或禁用
data.items.ifMode	string	接口的使用类型 枚举值:[ROUTE:路由类型, 接口作为3层接口, 提供IP地址配置, 路由配置转发等功能 SWITCH:交换类型, 接口作为2层接口, 提供二层转发、vlan配置等功能 VIRTUALLINE:虚拟线类型, 接口作为虚拟线的一端, 透明传输报文 BYPASSMIRROR:旁路镜像类型, 该模式进来的报文, 只处理不转发]
data.items.ipv4	object	接口IPv4相关配置
data.items.ipv4.ipv4Mode	string	IPv4地址获取类型 枚举值:[STATIC:静态配置 DHCP:dhcp方式获取 PPPOE:pppoe方式获取]
data.items.ipv4.staticIp	object[]	IPv4地址列表
data.items.ipv4.staticIp.ipAddress	object	IPv4地址 格式:ip-range
data.items.ipv4.staticIp.ipAddress.start	string	ip范围的起始位置 格式:ip-address
data.items.ipv4.staticIp.ipAddress.end	string	ip范围的终止位置 格式:ip-address
data.items.ipv4.staticIp.ipAddress.bits	uint8	ip掩码简写
data.items.ipv4.staticIp.ipAddress.mask	string	ip掩码 格式:ip-address
data.items.ipv4.staticIp.isSync	bool	是否不同步到备机的选项
data.items.ipv4.dhcp	object	dhcp4 client配置
data.items.ipv4.dhcp.gateway	bool	是否设置dhcp gateway
data.items.ipv4.pppoeGroupName	string	pppoe名称
data.items.ipv6	object	接口IPv6相关配置

data.items.ipv6.ipv6Mode	string	IPv6地址获取类型，路由模式有效 枚举值:[STATIC:static IP DHCP6:dhcp]
data.items.ipv6.staticIp	object[]	IPv6地址列表
data.items.ipv6.staticIp.start	string	ip范围的起始位置 格式:ip-address
data.items.ipv6.staticIp.end	string	ip范围的终止位置 格式:ip-address
data.items.ipv6.staticIp.bits	uint8	ip掩码简写
data.items.ipv6.staticIp.mask	string	ip掩码 格式:ip-address
data.items.wanEnable	bool	是否开启接口的wan模式
data.items.bandSwitch	object	接口接收的带宽限制
data.items.bandSwitch.ingressbandSwitch	object	接口接收的带宽限制
data.items.bandSwitch.ingressbandSwitch.value	uint64	数值大小 最小值:0 最大值:1099511627776
data.items.bandSwitch.egressbandSwitch	object	接口发送的带宽限制
data.items.bandSwitch.egressbandSwitch.value	uint64	数值大小 最小值:0 最大值:1099511627776
data.items.manage	object	本地策略控制，接口上控制是否放行对应类型的报文
data.items.manage.ssh	bool	是否启用ssh
data.items.manage.snmp	bool	是否启用snmp
data.items.manage.https	bool	是否启用https
data.items.manage.ping	bool	是否启用ping
data.items.manage.update	bool	是否启用系统升级维护
data.items.vlan	object	vlan配置
data.items.vlan.vlanMode	string	vlan模式 枚举值:[ACCESS:access模式 TRUNK:trunk模式]
data.items.vlan.accessId	uint32	access vlan id 最小值:0 最大值:4094
data.items.vlan.trunkConfig	object	trunk配置

data.items.vlan.trunkConfig.nativeVid	int32	trunk native vlanid 最小值:0 最大值:4094
data.items.vlan.trunkConfig.trunkRange	object[]	trunk vlan 范围
data.items.vlan.trunkConfig.trunkRange.start	uint32	start vlan 最小值:1 最大值:4094
data.items.vlan.trunkConfig.trunkRange.end	uint32	start vlan 最小值:1 最大值:4094
data.items.vlanId	uint32	标识子接口和vlan口的vlanid 最小值:1 最大值:4094
data.items.physicalif	object	物理接口相关配置
data.items.physicalif.speedDuplex	object	速率双工配置
data.items.physicalif.speedDuplex.autoneg	bool	接口速率工作模式自动协商, 为true时speed和duplex不再生效
data.items.physicalif.speedDuplex.speed	uint32	接口速率, autoneg为false时speed配置生效 枚举值:[10:选择10Mbps 100:选择100Mbps 1000:选择1000Mbps 10000:选择10000Mbps]
data.items.physicalif.speedDuplex.duplex	string	双工模式, autoneg为false时duplex配置生效 枚举值:[half:半双工模式 full:全双工模式]
data.items.physicalif.bypassMirror	object	旁路镜像模式有效
data.items.physicalif.bypassMirror.statsEnable	bool	是否启用流量统计
data.items.physicalif.bypassMirror.ipGroup	string[]	关联的网络对象
data.items.subif	object	子接口相关配置
data.items.subif.fatherInterface	string	接口的父接口, 子接口时表示父接口
data.items.channelif	object	汇聚口相关配置
data.items.channelif.channelMode	string	汇聚口的工作模式配置 枚举值:[STATIC:静态 LACP:动态]
data.items.channelif.channelStatic	object	汇聚口静态模式下的相关配置

data.items.channelif.channelStatic.staticMode	string	模式选择 枚举值:[FAILOVER:故障转移 LOADBALANCE:负载均衡]
data.items.channelif.channelStatic.failover	object	故障转移
data.items.channelif.channelStatic.failover.preempt	bool	主备模式下的抢占或非抢占
data.items.channelif.channelStatic.failover.channelPrimary	string	聚合口主备模式的主接口，状态字段，配置无效
data.items.channelif.channelStatic.loadbalance	string	负载均衡的方法 枚举值:[ROUNDROBIN:平衡轮询策略 L3HASH:hash负载分担平衡策略]
data.items.channelif.channelLacp	object	动态模式下的相关配置
data.items.channelif.channelLacp.hashMode	string	哈希策略 枚举值:[MAC:源与目的IP 2TUPLE:源与目的IP、mac 3TUPLE:源与目的IP、mac、port]
data.items.channelif.channelLacp.negotiateMode	string	协商模式 枚举值:[INITIATIVE:源与目的IP PASSIVE:源与目的IP、mac]
data.items.channelif.ethSelect	string[]	选择的物理口列表

请求成功样例：

```
{
  "data": {
    "itemsOffset": 0,
    "items": [
      {
        "ifType": "PHYSICALIF",
        "wanEnable": false,
        "mtu": 1500,
        "bandSwitch": {
          "ingressbandSwitch": {
            "value": 1073741824
          },
          "egressbandSwitch": {
            "value": 1073741824
          }
        },
        "uuid": "0000000000000000000000000500000001",
        "physicalif": {
          "speedDuplex": {
            "duplex": "full",
            "autoneg": true,
            "speed": 1000
          }
        }
      }
    ]
  }
}
```

```
    },
    "manage": {
      "ping": true,
      "snmp": false,
      "https": true,
      "ssh": false
    },
    "ipv4": {
      "staticIp": [],
      "ipv4Mode": "STATIC"
    },
    "mac": "FE:FC:FE:0B:D1:BB",
    "ipv6": {
      "ipv6Mode": "STATIC",
      "staticIp": []
    },
    "ifMode": "ROUTE",
    "name": "eth1",
    "shutdown": false,
    "description": ""
  }
},
"totalItems": 1,
"pageNumber": 1,
"itemLength": 1,
"totalPages": 1,
"pageSize": 100
},
"code": 0,
"message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3.2.2 获取单个接口信息

功能说明: 获取单个接口信息

1) API请求数据

GET

https://{host}/api/v1/namespaces/@namespace/interfaces/@name

2) 输入参数

- URL 参数

参数	类型	是否必选	描述
_select	string	否	选择字段, 字段用逗号分隔 最大长度:1000

请求样例:

获取单个接口的配置信息, 查询条件: 查看名为eth1的物理接口

GET https://192.168.1.1/api/v1/namespaces/public/interfaces/eth1

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码

message	string	错误信息
data	object	接口信息
data.uuid	string	唯一标识id
data.name	string	接口名称
data.description	string	接口描述 最小长度:0 最大长度:95
data.mtu	int32	最大传输单元 最小值:68 最大值:1796
data.ifType	string	表示接口类型 枚举值:[PHYSICALIF:物理口 SUBIF:子接口 CHANNELIF:汇聚口 VLANIF:vlan对应的3层接口]
data.mac	string	接口的mac地址
data.shutdown	bool	接口的启用或禁用
data.ifMode	string	接口的使用类型 枚举值:[ROUTE:路由类型, 接口作为3层接口, 提供IP地址配置, 路由配置转发等功能 SWITCH:交换类型, 接口作为2层接口, 提供二层转发、vlan配置等功能 VIRTUALLINE:虚拟线类型, 接口作为虚拟线的一端, 透明传输报文 BYPASSMIRROR:旁路镜像类型, 该模式进来的报文, 只处理不转发]
data.ipv4	object	接口IPv4相关配置
data.ipv4.ipv4Mode	string	IPv4地址获取类型 枚举值:[STATIC:静态配置 DHCP:dhcp方式获取 PPPOE:pppoe方式获取]
data.ipv4.staticIp	object[]	IPv4地址列表
data.ipv4.staticIp.ipaddress	object	IPv4地址 格式:ip-range
data.ipv4.staticIp.ipaddress.start	string	ip范围的起始位置 格式:ip-address
data.ipv4.staticIp.ipaddress.end	string	ip范围的终止位置 格式:ip-address
data.ipv4.staticIp.ipaddress.bits	uint8	ip掩码简写
data.ipv4.staticIp.ipaddress.mask	string	ip掩码 格式:ip-address

data.ipv4.staticIp.isSync	bool	是否不同步到备机的选项
data.ipv4.dhcp	object	dhcp4 client配置
data.ipv4.dhcp.gateway	bool	是否设置dhcp gateway
data.ipv4.pppoeGroupName	string	pppoe名称
data.ipv6	object	接口IPv6相关配置
data.ipv6.ipv6Mode	string	IPv6地址获取类型，路由模式有效 枚举值:[STATIC:static IP DHCP6:dhcp]
data.ipv6.staticIp	object[]	IPv6地址列表
data.ipv6.staticIp.start	string	ip范围的起始位置 格式:ip-address
data.ipv6.staticIp.end	string	ip范围的终止位置 格式:ip-address
data.ipv6.staticIp.bits	uint8	ip掩码简写
data.ipv6.staticIp.mask	string	ip掩码 格式:ip-address
data.wanEnable	bool	是否开启接口的wan模式
data.bandSwitch	object	接口接收的带宽限制
data.bandSwitch.ingressbandSwitch	object	接口接收的带宽限制
data.bandSwitch.ingressbandSwitch.value	uint64	数值大小 最小值:0 最大值:1099511627776
data.bandSwitch.egressbandSwitch	object	接口发送的带宽限制
data.bandSwitch.egressbandSwitch.value	uint64	数值大小 最小值:0 最大值:1099511627776
data.manage	object	本地策略控制，接口上控制是否放行对应类型的报文
data.manage.ssh	bool	是否启用ssh
data.manage.snmp	bool	是否启用snmp
data.manage.https	bool	是否启用https
data.manage.ping	bool	是否启用ping
data.manage.update	bool	是否启用系统升级维护
data.vlan	object	vlan配置
data.vlan.vlanMode	string	vlan模式 枚举值:[ACCESS:access模式 TRUNK:trunk模式]

data.vlan.accessId	uint32	access vlan id 最小值:0 最大值:4094
data.vlan.trunkConfig	object	trunk配置
data.vlan.trunkConfig.nativeId	int32	trunk native vlanid 最小值:0 最大值:4094
data.vlan.trunkConfig.trunkRange	object[]	trunk vlan 范围
data.vlan.trunkConfig.trunkRange.start	uint32	start vlan 最小值:1 最大值:4094
data.vlan.trunkConfig.trunkRange.end	uint32	start vlan 最小值:1 最大值:4094
data.vlanId	uint32	标识子接口和vlan口的vlanid 最小值:1 最大值:4094
data.physicalif	object	物理接口相关配置
data.physicalif.speedDuplex	object	速率双工配置
data.physicalif.speedDuplex.autoneg	bool	接口速率工作模式自动协商, 为true时speed和duplex不再生效
data.physicalif.speedDuplex.speed	uint32	接口速率, autoneg为false时speed配置生效 枚举值:[10:选择10Mbps 100:选择100Mbps 1000:选择1000Mbps 10000:选择10000Mbps]
data.physicalif.speedDuplex.duplex	string	双工模式, autoneg为false时duplex配置生效 枚举值:[half:半双工模式 full:全双工模式]
data.physicalif.bypassMirror	object	旁路镜像模式有效
data.physicalif.bypassMirror.statsEnable	bool	是否启用流量统计
data.physicalif.bypassMirror.ipGroup	string[]	关联的网络对象
data.subif	object	子接口相关配置
data.subif.fatherInterface	string	接口的父接口, 子接口时表示父接口
data.channelif	object	汇聚口相关配置

data.channelif.channelMode	string	汇聚口的工作模式配置 枚举值:[STATIC:静态 LACP:动态]
data.channelif.channelStatic	object	汇聚口静态模式下的相关配置
data.channelif.channelStatic.staticMode	string	模式选择 枚举值:[FAILOVER:故障转移 LOADBALANCE:负载均衡]
data.channelif.channelStatic.failover	object	故障转移
data.channelif.channelStatic.failover.preempt	bool	主备模式下的抢占或非抢占
data.channelif.channelStatic.failover.channelPrimary	string	聚合口主备模式的主接口，状态字段，配置无效
data.channelif.channelStatic.loadbalance	string	负载均衡的方法 枚举值:[ROUNDROBIN:平衡轮询策略 L3HASH:hash负载分担平衡策略]
data.channelif.channelLacp	object	动态模式下的相关配置
data.channelif.channelLacp.hashMode	string	哈希策略 枚举值:[MAC:源与目的IP 2TUPLE:源与目的IP、mac 3TUPLE:源与目的IP、mac、port]
data.channelif.channelLacp.negotiateMode	string	协商模式 枚举值:[INITIATIVE:源与目的IP PASSIVE:源与目的IP、mac]
data.channelif.ethSelect	string[]	选择的物理口列表

请求成功样例:

```
{
  "data": {
    "ifType": "PHYSICALIF",
    "wanEnable": false,
    "mtu": 1500,
    "bandSwitch": {
      "ingressbandSwitch": {
        "value": 1073741824
      },
      "egressbandSwitch": {
        "value": 1073741824
      }
    },
    "uuid": "0000000000000000000000000500000001",
    "physicalif": {
```

```
"speedDuplex": {
  "duplex": "full",
  "autoneg": true,
  "speed": 1000
},
"manage": {
  "ping": true,
  "snmp": false,
  "https": true,
  "ssh": false
},
"ipv4": {
  "staticIp": [],
  "ipv4Mode": "STATIC"
},
"mac": "FE:FC:FE:0B:D1:BB",
"ipv6": {
  "ipv6Mode": "STATIC",
  "staticIp": []
},
"ifMode": "ROUTE",
"name": "eth1",
"shutdown": false,
"description": ""
},
"code": 0,
"message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3.2.3 添加接口信息

功能说明: 添加接口信息

1) API请求数据

POST https://{host}/api/v1/namespaces/@namespace/interfaces

2) 输入参数

- Body参数

参数	类型	是否必选	描述
obj	object	是	接口信息
obj.name	string	是	接口名称
obj.description	string	否	接口描述 最小长度:0 最大长度:95
obj.mtu	int32	否	最大传输单元 最小值:68 最大值:1796 默认值:true

obj.ifType	string	是	表示接口类型 枚举值:[PHYSICALIF:物理口 SUBIF:子接口 CHANNELIF:汇聚口 VLANIF:vlan对应的3层接口]
obj.mac	string	否	接口的mac地址
obj.shutdown	bool	否	接口的启用或禁用 默认值:true
obj.ifMode	string	否	接口的使用类型 默认值:true 枚举值:[ROUTE:路由类型, 接口作为3层接口, 提供IP地址配置, 路由配置转发等功能 SWITCH:交换类型, 接口作为2层接口, 提供二层转发、vlan配置等功能 VIRTUALLINE:虚拟线类型, 接口作为虚拟线的一端, 透明传输报文 BYPASSMIRROR:旁路镜像类型, 该模式进来的报文, 只处理不转发]
obj.ipv4	object	否	接口IPv4相关配置
obj.ipv4.ipv4Mode	string	否	IPv4地址获取类型 枚举值:[STATIC:静态配置 DHCP:dhcp方式获取 PPPOE:pppoe方式获取]
obj.ipv4.staticIp	object[]	否	IPv4地址列表
obj.ipv4.staticIp.ipaddress	object	是	IPv4地址 格式:ip-range
obj.ipv4.staticIp.ipaddress.start	string	是	ip范围的起始位置 格式:ip-address
obj.ipv4.staticIp.ipaddress.end	string	否	ip范围的终止位置 格式:ip-address
obj.ipv4.staticIp.ipaddress.bits	uint8	否	ip掩码简写
obj.ipv4.staticIp.ipaddress.mask	string	否	ip掩码 格式:ip-address
obj.ipv4.staticIp.isSync	bool	否	是否不同步到备机的选项
obj.ipv4.dhcp	object	否	dhcp4 client配置
obj.ipv4.dhcp.gateway	bool	否	是否设置dhcp gateway
obj.ipv4.pppoeGroupName	string	否	pppoe名称
obj.ipv6	object	否	接口IPv6相关配置

obj.ipv6.ipv6Mode	string	否	IPv6地址获取类型，路由模式有效枚举值:[STATIC:static IP DHCP6:dhcp]
obj.ipv6.staticIp	object[]	否	IPv6地址列表
obj.ipv6.staticIp.start	string	是	ip范围的起始位置 格式:ip-address
obj.ipv6.staticIp.end	string	否	ip范围的终止位置 格式:ip-address
obj.ipv6.staticIp.bits	uint8	否	ip掩码简写
obj.ipv6.staticIp.mask	string	否	ip掩码 格式:ip-address
obj.wanEnable	bool	否	是否开启接口的wan模式
obj.bandSwitch	object	否	接口接收的带宽限制
obj.bandSwitch.ingress bandSwitch	object	否	接口接收的带宽限制
obj.bandSwitch.ingress bandSwitch.value	uint64	否	数值大小 最小值:0 最大值:1099511627776 默认值:true
obj.bandSwitch.egressb andSwitch	object	否	接口发送的带宽限制
obj.bandSwitch.egressb andSwitch.value	uint64	否	数值大小 最小值:0 最大值:1099511627776 默认值:true
obj.manage	object	否	本地策略控制，接口上控制是否放行对应类型的报文
obj.manage.ssh	bool	否	是否启用ssh
obj.manage.snmp	bool	否	是否启用snmp
obj.manage.https	bool	是	是否启用https
obj.manage.ping	bool	否	是否启用ping 默认值:true
obj.manage.update	bool	否	是否启用系统升级维护
obj.vlan	object	否	vlan配置
obj.vlan.vlanMode	string	否	vlan模式 枚举值:[ACCESS:access模式 TRUNK:trunk模式]
obj.vlan.accessId	uint32	否	access vlan id 最小值:0 最大值:4094
obj.vlan.trunkConfig	object	否	trunk配置

obj.vlan.trunkConfig.nativeVid	int32	否	trunk native vlanid 最小值:0 最大值:4094
obj.vlan.trunkConfig.trunkRange	object[]	否	trunk vlan 范围
obj.vlan.trunkConfig.trunkRange.start	uint32	是	start vlan 最小值:1 最大值:4094
obj.vlan.trunkConfig.trunkRange.end	uint32	否	start vlan 最小值:1 最大值:4094
obj.vlanId	uint32	否	标识子接口和vlan口的vlanid 最小值:1 最大值:4094
obj.physicalif	object	否	物理接口相关配置
obj.physicalif.speedDuplex	object	否	速率双工配置
obj.physicalif.speedDuplex.autoneg	bool	否	接口速率工作模式自动协商，为true时speed和duplex不再生效 默认值:true
obj.physicalif.speedDuplex.speed	uint32	否	接口速率，autoneg为false时speed配置生效 默认值:true 枚举值:[10:选择10Mbps 100:选择100Mbps 1000:选择1000Mbps 10000:选择10000Mbps]
obj.physicalif.speedDuplex.duplex	string	否	双工模式，autoneg为false时duplex配置生效 默认值:true 枚举值:[half:半双工模式 full:全双工模式]
obj.physicalif.bypassMirror	object	否	旁路镜像模式有效
obj.physicalif.bypassMirror.statsEnable	bool	否	是否启用流量统计
obj.physicalif.bypassMirror.ipGroup	string[]	否	关联的网络对象
obj.subif	object	否	子接口相关配置
obj.subif.fatherInterface	string	是	接口的父接口，子接口时表示父接口
obj.channelif	object	否	汇聚口相关配置

obj.channelif.channelMode	string	否	汇聚口的工作模式配置 枚举值:[STATIC:静态 LACP:动态]
obj.channelif.channelStatic	object	否	汇聚口静态模式下的相关配置
obj.channelif.channelStatic.staticMode	string	否	模式选择 枚举值:[FAILOVER:故障转移 LOADBALANCE:负载均衡]
obj.channelif.channelStatic.failover	object	否	故障转移
obj.channelif.channelStatic.failover.preempt	bool	是	主备模式下的抢占或非抢占
obj.channelif.channelStatic.failover.channelPrimary	string	否	聚合口主备模式的主接口，状态字段，配置无效
obj.channelif.channelStatic.loadbalance	string	否	负载均衡的方法 枚举值:[ROUNDROBIN:平衡轮询策略 L3HASH:hash负载分担平衡策略]
obj.channelif.channelLacp	object	否	动态模式下的相关配置
obj.channelif.channelLacp.hashMode	string	否	哈希策略 枚举值:[MAC:源与目的IP 2TUPLE:源与目的IP、mac 3TUPLE:源与目的IP、mac、port]
obj.channelif.channelLacp.negotiateMode	string	否	协商模式 枚举值:[INITIATIVE:源与目的IP PASSIVE:源与目的IP、mac]
obj.channelif.ethSelect	string[]	否	选择的物理口列表

请求样例:

新增单个接口信息

```
POST https://192.168.1.1/api/v1/namespaces/public/interfaces
{
  "ifType": "SUBIF",
  "subif": {
    "fatherInterface": "eth0"
  },
  "manage": {
    "ping": false,
    "snmp": false,
    "https": true,
    "ssh": false
  },
  "mtu": 1500,
```

```

    "ipv4": {
      "staticIp": [],
      "ipv4Mode": "STATIC"
    },
    "vlanId": 1,
    "name": "eth0.1",
    "description": ""
  }

```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	接口信息
data.uuid	string	唯一标识id
data.name	string	接口名称
data.description	string	接口描述 最小长度:0 最大长度:95
data.mtu	int32	最大传输单元 最小值:68 最大值:1796
data.ifType	string	表示接口类型 枚举值: PHYSICALIF:物理口 SUBIF:子接口 CHANNELIF:汇聚口 VLANIF:vlan对应的3层接口]
data.mac	string	接口的mac地址
data.shutdown	bool	接口的启用或禁用
data.ifMode	string	接口的使用类型 枚举值: ROUTE:路由类型, 接口作为3层接口, 提供IP地址配置, 路由配置转发等功能 SWITCH:交换类型, 接口作为2层接口, 提供二层转发、vlan配置等功能 VIRTUALLINE:虚拟线类型, 接口作为虚拟线的一端, 透明传输报文 BYPASSMIRROR:旁路镜像类型, 该模式进来的报文, 只处理不转发]
data.ipv4	object	接口IPv4相关配置
data.ipv4.ipv4Mode	string	IPv4地址获取类型 枚举值: STATIC:静态配置 DHCP:dhcp方式获取 PPPOE:pppoe方式获取]

data.ipv4.staticIp	object[]	IPv4地址列表
data.ipv4.staticIp.ipaddress	object	IPv4地址 格式:ip-range
data.ipv4.staticIp.ipaddress.start	string	ip范围的起始位置 格式:ip-address
data.ipv4.staticIp.ipaddress.end	string	ip范围的终止位置 格式:ip-address
data.ipv4.staticIp.ipaddress.bits	uint8	ip掩码简写
data.ipv4.staticIp.ipaddress.mask	string	ip掩码 格式:ip-address
data.ipv4.staticIp.isSync	bool	是否不同步到备机的选项
data.ipv4.dhcp	object	dhcp4 client配置
data.ipv4.dhcp.gateway	bool	是否设置dhcp gateway
data.ipv4.pppoeGroupName	string	pppoe名称
data.ipv6	object	接口IPv6相关配置
data.ipv6.ipv6Mode	string	IPv6地址获取类型, 路由模式有效 枚举值:[STATIC:static IP DHCP6:dhcp]
data.ipv6.staticIp	object[]	IPv6地址列表
data.ipv6.staticIp.start	string	ip范围的起始位置 格式:ip-address
data.ipv6.staticIp.end	string	ip范围的终止位置 格式:ip-address
data.ipv6.staticIp.bits	uint8	ip掩码简写
data.ipv6.staticIp.mask	string	ip掩码 格式:ip-address
data.wanEnable	bool	是否开启接口的wan模式
data.bandSwitch	object	接口接收的带宽限制
data.bandSwitch.ingressbandSwitch	object	接口接收的带宽限制
data.bandSwitch.ingressbandSwitch.value	uint64	数值大小 最小值:0 最大值:1099511627776
data.bandSwitch.egressbandSwitch	object	接口发送的带宽限制
data.bandSwitch.egressbandSwitch.value	uint64	数值大小 最小值:0 最大值:1099511627776
data.manage	object	本地策略控制, 接口上控制是否放行对应类型的报文
data.manage.ssh	bool	是否启用ssh

data.manage.snmp	bool	是否启用snmp
data.manage.https	bool	是否启用https
data.manage.ping	bool	是否启用ping
data.manage.update	bool	是否启用系统升级维护
data.vlan	object	vlan配置
data.vlan.vlanMode	string	vlan模式 枚举值:[ACCESS:access模式 TRUNK:trunk模式]
data.vlan.accessId	uint32	access vlan id 最小值:0 最大值:4094
data.vlan.trunkConfig	object	trunk配置
data.vlan.trunkConfig.nativeId	int32	trunk native vlanid 最小值:0 最大值:4094
data.vlan.trunkConfig.trunkRange	object[]	trunk vlan 范围
data.vlan.trunkConfig.trunkRange.start	uint32	start vlan 最小值:1 最大值:4094
data.vlan.trunkConfig.trunkRange.end	uint32	start vlan 最小值:1 最大值:4094
data.vlanId	uint32	标识子接口和vlan口的vlanid 最小值:1 最大值:4094
data.physicalif	object	物理接口相关配置
data.physicalif.speedDuplex	object	速率双工配置
data.physicalif.speedDuplex.autoneg	bool	接口速率工作模式自动协商, 为true时speed和duplex不再生效
data.physicalif.speedDuplex.speed	uint32	接口速率, autoneg为false时speed配置生效 枚举值:[10:选择10Mbps 100:选择100Mbps 1000:选择1000Mbps 10000:选择10000Mbps]
data.physicalif.speedDuplex.duplex	string	双工模式, autoneg为false时duplex配置生效 枚举值:[half:半双工模式 full:全双工模式]
data.physicalif.bypassMirror	object	旁路镜像模式有效

data.physicalif.bypassMirror.statsEnable	bool	是否启用流量统计
data.physicalif.bypassMirror.ipGroup	string[]	关联的网络对象
data.subif	object	子接口相关配置
data.subif.fatherInterface	string	接口的父接口，子接口时表示父接口
data.channelif	object	汇聚口相关配置
data.channelif.channelMode	string	汇聚口的工作模式配置 枚举值:[STATIC:静态 LACP:动态]
data.channelif.channelStatic	object	汇聚口静态模式下的相关配置
data.channelif.channelStatic.staticMode	string	模式选择 枚举值:[FAILOVER:故障转移 LOADBALANCE:负载均衡]
data.channelif.channelStatic.failover	object	故障转移
data.channelif.channelStatic.failover.preempt	bool	主备模式下的抢占或非抢占
data.channelif.channelStatic.failover.channelPrimary	string	聚合口主备模式的主接口，状态字段，配置无效
data.channelif.channelStatic.loadbalance	string	负载均衡的方法 枚举值:[ROUNDROBIN:平衡轮询策略 L3HASH:hash负载分担平衡策略]
data.channelif.channelLacp	object	动态模式下的相关配置
data.channelif.channelLacp.hashMode	string	哈希策略 枚举值:[MAC:源与目的IP 2TUPLE:源与目的IP、mac 3TUPLE:源与目的IP、mac、port]
data.channelif.channelLacp.negotiateMode	string	协商模式 枚举值:[INITIATIVE:源与目的IP PASSIVE:源与目的IP、mac]
data.channelif.ethSelect	string[]	选择的物理口列表

请求成功样例：


```
{
  "data": {
    "ifType": "SUBIF",
    "mtu": 1500,
    "vlanId": 1,
    "uuid": "0000000000000000000000000600000001",
    "subif": {
      "fatherInterface": "eth0"
    },
    "ipv4": {
      "staticIp": [],
      "ipv4Mode": "STATIC"
    },
    "manage": {
      "ping": false,
      "snmp": false,
      "https": true,
      "ssh": false
    },
    "name": "eth0.1",
    "description": ""
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3.2.4 删除指定接口信息

功能说明: 删除指定接口信息

1) API请求数据

DELETE

https://{host}/api/v1/namespaces/@namespace/interfaces/@name

2) 输入参数

请求样例:

删除一个接口，删除名为eth0.1的子接口

DELETE https://192.168.1.1/api/v1/namespaces/public/interfaces/eth0.1

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	接口信息
data.uuid	string	唯一标识id
data.name	string	接口名称
data.description	string	接口描述 最小长度:0 最大长度:95

data.mtu	int32	最大传输单元 最小值:68 最大值:1796
data.ifType	string	表示接口类型 枚举值:[PHYSICALIF:物理口 SUBIF:子接口 CHANNELIF:汇聚口 VLANIF:vlan对应的3层接口]
data.mac	string	接口的mac地址
data.shutdown	bool	接口的启用或禁用
data.ifMode	string	接口的使用类型 枚举值:[ROUTE:路由类型, 接口作为3层接口, 提供IP地址配置, 路由配置转发等功能 SWITCH:交换类型, 接口作为2层接口, 提供二层转发、vlan配置等功能 VIRTUALLINE:虚拟线类型, 接口作为虚拟线的一端, 透明传输报文 BYPASSMIRROR:旁路镜像类型, 该模式进来的报文, 只处理不转发]
data.ipv4	object	接口IPv4相关配置
data.ipv4.ipv4Mode	string	IPv4地址获取类型 枚举值:[STATIC:静态配置 DHCP:dhcp方式获取 PPPOE:pppoe方式获取]
data.ipv4.staticIp	object[]	IPv4地址列表
data.ipv4.staticIp.ipaddress	object	IPv4地址 格式:ip-range
data.ipv4.staticIp.ipaddress.start	string	ip范围的起始位置 格式:ip-address
data.ipv4.staticIp.ipaddress.end	string	ip范围的终止位置 格式:ip-address
data.ipv4.staticIp.ipaddress.bits	uint8	ip掩码简写
data.ipv4.staticIp.ipaddress.mask	string	ip掩码 格式:ip-address
data.ipv4.staticIp.isSync	bool	是否不同步到备机的选项
data.ipv4.dhcp	object	dhcp4 client配置
data.ipv4.dhcp.gateway	bool	是否设置dhcp gateway
data.ipv4.pppoeGroupName	string	pppoe名称
data.ipv6	object	接口IPv6相关配置

data.ipv6.ipv6Mode	string	IPv6地址获取类型，路由模式有效 枚举值:[STATIC:static IP DHCP6:dhcp]
data.ipv6.staticIp	object[]	IPv6地址列表
data.ipv6.staticIp.start	string	ip范围的起始位置 格式:ip-address
data.ipv6.staticIp.end	string	ip范围的终止位置 格式:ip-address
data.ipv6.staticIp.bits	uint8	ip掩码简写
data.ipv6.staticIp.mask	string	ip掩码 格式:ip-address
data.wanEnable	bool	是否开启接口的wan模式
data.bandSwitch	object	接口接收的带宽限制
data.bandSwitch.ingressbandSwitch	object	接口接收的带宽限制
data.bandSwitch.ingressbandSwitch.value	uint64	数值大小 最小值:0 最大值:1099511627776
data.bandSwitch.egressbandSwitch	object	接口发送的带宽限制
data.bandSwitch.egressbandSwitch.value	uint64	数值大小 最小值:0 最大值:1099511627776
data.manage	object	本地策略控制，接口上控制是否放行对应类型的报文
data.manage.ssh	bool	是否启用ssh
data.manage.snmp	bool	是否启用snmp
data.manage.https	bool	是否启用https
data.manage.ping	bool	是否启用ping
data.manage.update	bool	是否启用系统升级维护
data.vlan	object	vlan配置
data.vlan.vlanMode	string	vlan模式 枚举值:[ACCESS:access模式 TRUNK:trunk模式]
data.vlan.accessId	uint32	access vlan id 最小值:0 最大值:4094
data.vlan.trunkConfig	object	trunk配置
data.vlan.trunkConfig.nativeId	int32	trunk native vlanid 最小值:0 最大值:4094

data.vlan.trunkConfig.trunkRange	object[]	trunk vlan 范围
data.vlan.trunkConfig.trunkRange.start	uint32	start vlan 最小值:1 最大值:4094
data.vlan.trunkConfig.trunkRange.end	uint32	start vlan 最小值:1 最大值:4094
data.vlanId	uint32	标识子接口和vlan口的vlanid 最小值:1 最大值:4094
data.physicalif	object	物理接口相关配置
data.physicalif.speedDuplex	object	速率双工配置
data.physicalif.speedDuplex.autoneg	bool	接口速率工作模式自动协商, 为true时speed和duplex不再生效
data.physicalif.speedDuplex.speed	uint32	接口速率, autoneg为false时speed配置生效 枚举值:[10:选择10Mbps 100:选择100Mbps 1000:选择1000Mbps 10000:选择10000Mbps]
data.physicalif.speedDuplex.duplex	string	双工模式, autoneg为false时duplex配置生效 枚举值:[half:半双工模式 full:全双工模式]
data.physicalif.bypassMirror	object	旁路镜像模式有效
data.physicalif.bypassMirror.statsEnable	bool	是否启用流量统计
data.physicalif.bypassMirror.ipGroup	string[]	关联的网络对象
data.subif	object	子接口相关配置
data.subif.fatherInterface	string	接口的父接口, 子接口时表示父接口
data.channelif	object	汇聚口相关配置
data.channelif.channelMode	string	汇聚口的工作模式配置 枚举值:[STATIC:静态 LACP:动态]
data.channelif.channelStatic	object	汇聚口静态模式下的相关配置

data.channelif.channelStatic.staticMode	string	模式选择 枚举值:[FAILOVER:故障转移 LOADBALANCE:负载均衡]
data.channelif.channelStatic.failover	object	故障转移
data.channelif.channelStatic.failover.preempt	bool	主备模式下的抢占或非抢占
data.channelif.channelStatic.failover.channelPrimary	string	聚合口主备模式的主接口，状态字段，配置无效
data.channelif.channelStatic.loadbalance	string	负载均衡的方法 枚举值:[ROUNDROBIN:平衡轮询策略 L3HASH:hash负载分担平衡策略]
data.channelif.channelLacp	object	动态模式下的相关配置
data.channelif.channelLacp.hashMode	string	哈希策略 枚举值:[MAC:源与目的IP 2TUPLE:源与目的IP、mac 3TUPLE:源与目的IP、mac、port]
data.channelif.channelLacp.negotiateMode	string	协商模式 枚举值:[INITIATIVE:源与目的IP PASSIVE:源与目的IP、mac]
data.channelif.ethSelect	string[]	选择的物理口列表

请求成功样例:

```
{
  "data": {
    "ifType": "SUBIF",
    "mtu": 1500,
    "vlanId": 1,
    "subif": {
      "fatherInterface": "eth0"
    },
    "manage": {
      "ping": true,
      "ssh": false,
      "snmp": false,
      "https": true
    },
    "ipv4": {
      "staticIp": [],
      "ipv4Mode": "STATIC"
    },
    "uuid": "00000000000000000000000006000000001",
    "name": "eth0.1",
    "description": ""
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3.2.5 增量修改指定接口信息

功能说明：增量修改指定接口信息

1) API请求数据

PATCH https://{host}/api/v1/namespaces/@namespace/interfaces/@name

2) 输入参数

• URL参数

参数	类型	是否必选	描述
_arrayop	string	否	对象属性是数组的操作控制 枚举值:[add:添加对应的数组元素 remove:删除对应的数组元素]

• Body参数

参数	类型	是否必选	描述
obj	object	是	接口信息
obj.name	string	是	接口名称
obj.description	string	否	接口描述 最小长度:0 最大长度:95
obj.mtu	int32	否	最大传输单元 最小值:68 最大值:1796 默认值:true
obj.ifType	string	是	表示接口类型 枚举值:[PHYSICALIF:物理口 SUBIF:子接口 CHANNELIF:汇聚口 VLANIF:vlan对应的3层接口]
obj.mac	string	否	接口的mac地址
obj.shutdown	bool	否	接口的启用或禁用 默认值:true

obj.ifMode	string	否	接口的使用类型 默认值:true 枚举值:[ROUTE:路由类型, 接口作为3层接口, 提供IP地址配置, 路由配置转发等功能 SWITCH:交换类型, 接口作为2层接口, 提供二层转发、vlan配置等功能 VIRTUALLINE:虚拟线类型, 接口作为虚拟线的一端, 透明传输报文 BYPASSMIRROR:旁路镜像类型, 该模式进来的报文, 只处理不转发]
obj.ipv4	object	否	接口IPv4相关配置
obj.ipv4.ipv4Mode	string	否	IPv4地址获取类型 枚举值:[STATIC:静态配置 DHCP:dhcp方式获取 PPPOE:pppoe方式获取]
obj.ipv4.staticIp	object[]	否	IPv4地址列表
obj.ipv4.staticIp.ipaddresses	object	是	IPv4地址 格式:ip-range
obj.ipv4.staticIp.ipaddresses.start	string	是	ip范围的起始位置 格式:ip-address
obj.ipv4.staticIp.ipaddresses.end	string	否	ip范围的终止位置 格式:ip-address
obj.ipv4.staticIp.ipaddresses.bits	uint8	否	ip掩码简写
obj.ipv4.staticIp.ipaddresses.mask	string	否	ip掩码 格式:ip-address
obj.ipv4.staticIp.isSync	bool	否	是否不同步到备机的选项
obj.ipv4.dhcp	object	否	dhcp4 client配置
obj.ipv4.dhcp.gateway	bool	否	是否设置dhcp gateway
obj.ipv4.pppoeGroupName	string	否	pppoe名称
obj.ipv6	object	否	接口IPv6相关配置
obj.ipv6.ipv6Mode	string	否	IPv6地址获取类型, 路由模式有效 枚举值:[STATIC:static IP DHCP6:dhcp]
obj.ipv6.staticIp	object[]	否	IPv6地址列表
obj.ipv6.staticIp.start	string	是	ip范围的起始位置 格式:ip-address
obj.ipv6.staticIp.end	string	否	ip范围的终止位置 格式:ip-address
obj.ipv6.staticIp.bits	uint8	否	ip掩码简写

obj.ipv6.staticIp.mask	string	否	ip掩码 格式:ip-address
obj.wanEnable	bool	否	是否开启接口的wan模式
obj.bandSwitch	object	否	接口接收的带宽限制
obj.bandSwitch.ingress bandSwitch	object	否	接口接收的带宽限制
obj.bandSwitch.ingress bandSwitch.value	uint64	否	数值大小 最小值:0 最大值:1099511627776 默认值:true
obj.bandSwitch.egressb andSwitch	object	否	接口发送的带宽限制
obj.bandSwitch.egressb andSwitch.value	uint64	否	数值大小 最小值:0 最大值:1099511627776 默认值:true
obj.manage	object	否	本地策略控制，接口上控制是否放行对应类型的报文
obj.manage.ssh	bool	否	是否启用ssh
obj.manage.snmp	bool	否	是否启用snmp
obj.manage.https	bool	是	是否启用https
obj.manage.ping	bool	否	是否启用ping 默认值:true
obj.manage.update	bool	否	是否启用系统升级维护
obj.vlan	object	否	vlan配置
obj.vlan.vlanMode	string	否	vlan模式 枚举值:[ACCESS:access模式 TRUNK:trunk模式]
obj.vlan.accessId	uint32	否	access vlan id 最小值:0 最大值:4094
obj.vlan.trunkConfig	object	否	trunk配置
obj.vlan.trunkConfig.nati vId	int32	否	trunk native vlanid 最小值:0 最大值:4094
obj.vlan.trunkConfig.tru nkRange	object[]	否	trunk vlan 范围
obj.vlan.trunkConfig.tru nkRange.start	uint32	是	start vlan 最小值:1 最大值:4094
obj.vlan.trunkConfig.tru nkRange.end	uint32	否	start vlan 最小值:1 最大值:4094

obj.vlanId	uint32	否	标识子接口和vlan口的vlanid 最小值:1 最大值:4094
obj.physicalif	object	否	物理接口相关配置
obj.physicalif.speedDuplex	object	否	速率双工配置
obj.physicalif.speedDuplex.autoneg	bool	否	接口速率工作模式自动协商，为true时speed和duplex不再生效 默认值:true
obj.physicalif.speedDuplex.speed	uint32	否	接口速率，autoneg为false时speed配置生效 默认值:true 枚举值:[10:选择10Mbps 100:选择100Mbps 1000:选择1000Mbps 10000:选择10000Mbps]
obj.physicalif.speedDuplex.duplex	string	否	双工模式，autoneg为false时duplex配置生效 默认值:true 枚举值:[half:半双工模式 full:全双工模式]
obj.physicalif.bypassMirror	object	否	旁路镜像模式有效
obj.physicalif.bypassMirror.statsEnable	bool	否	是否启用流量统计
obj.physicalif.bypassMirror.ipGroup	string[]	否	关联的网络对象
obj.subif	object	否	子接口相关配置
obj.subif.fatherInterface	string	是	接口的父接口，子接口时表示父接口
obj.channelif	object	否	汇聚口相关配置
obj.channelif.channelMode	string	否	汇聚口的工作模式配置 枚举值:[STATIC:静态 LACP:动态]
obj.channelif.channelStatic	object	否	汇聚口静态模式下的相关配置
obj.channelif.channelStatic.staticMode	string	否	模式选择 枚举值:[FAILOVER:故障转移 LOADBALANCE:负载均衡]
obj.channelif.channelStatic.failover	object	否	故障转移

obj.channelif.channelStatic.failover.preempt	bool	是	主备模式下的抢占或非抢占
obj.channelif.channelStatic.failover.channelPrimary	string	否	聚合口主备模式的主接口，状态字段，配置无效
obj.channelif.channelStatic.loadbalance	string	否	负载均衡的方法 枚举值:[ROUNDROBIN:平衡轮询策略 L3HASH:hash负载分担平衡策略]
obj.channelif.channelLacp	object	否	动态模式下的相关配置
obj.channelif.channelLacp.hashMode	string	否	哈希策略 枚举值:[MAC:源与目的IP 2TUPLE:源与目的IP、mac 3TUPLE:源与目的IP、mac、port]
obj.channelif.channelLacp.negotiateMode	string	否	协商模式 枚举值:[INITIATIVE:源与目的IP PASSIVE:源与目的IP、mac]
obj.channelif.ethSelect	string[]	否	选择的物理口列表

请求样例:

修改单个接口，修改名为eth1的物理接口为禁用，其余不改

```
PATCH https://192.168.1.1/api/v1/namespaces/public/interfaces/eth1
{
  "shutdown": true
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	接口信息
data.uuid	string	唯一标识id
data.name	string	接口名称
data.description	string	接口描述 最小长度:0 最大长度:95
data.mtu	int32	最大传输单元 最小值:68 最大值:1796

data.ifType	string	表示接口类型 枚举值:[PHYSICALIF:物理口 SUBIF:子接口 CHANNELIF:汇聚口 VLANIF:vlan对应的3层接口]
data.mac	string	接口的mac地址
data.shutdown	bool	接口的启用或禁用
data.ifMode	string	接口的使用类型 枚举值:[ROUTE:路由类型, 接口作为3层接口, 提供IP地址配置, 路由配置转发等功能 SWITCH:交换类型, 接口作为2层接口, 提供二层转发、vlan配置等功能 VIRTUALLINE:虚拟线类型, 接口作为虚拟线的一端, 透明传输报文 BYPASSMIRROR:旁路镜像类型, 该模式进来的报文, 只处理不转发]
data.ipv4	object	接口IPv4相关配置
data.ipv4.ipv4Mode	string	IPv4地址获取类型 枚举值:[STATIC:静态配置 DHCP:dhcp方式获取 PPPOE:pppoe方式获取]
data.ipv4.staticIp	object[]	IPv4地址列表
data.ipv4.staticIp.ipaddress	object	IPv4地址 格式:ip-range
data.ipv4.staticIp.ipaddress.start	string	ip范围的起始位置 格式:ip-address
data.ipv4.staticIp.ipaddress.end	string	ip范围的终止位置 格式:ip-address
data.ipv4.staticIp.ipaddress.bits	uint8	ip掩码简写
data.ipv4.staticIp.ipaddress.mask	string	ip掩码 格式:ip-address
data.ipv4.staticIp.isSync	bool	是否不同步到备机的选项
data.ipv4.dhcp	object	dhcp4 client配置
data.ipv4.dhcp.gateway	bool	是否设置dhcp gateway
data.ipv4.pppoeGroupName	string	pppoe名称
data.ipv6	object	接口IPv6相关配置

data.ipv6.ipv6Mode	string	IPv6地址获取类型，路由模式有效 枚举值:[STATIC:static IP DHCP6:dhcp]
data.ipv6.staticIp	object[]	IPv6地址列表
data.ipv6.staticIp.start	string	ip范围的起始位置 格式:ip-address
data.ipv6.staticIp.end	string	ip范围的终止位置 格式:ip-address
data.ipv6.staticIp.bits	uint8	ip掩码简写
data.ipv6.staticIp.mask	string	ip掩码 格式:ip-address
data.wanEnable	bool	是否开启接口的wan模式
data.bandSwitch	object	接口接收的带宽限制
data.bandSwitch.ingressbandSwitch	object	接口接收的带宽限制
data.bandSwitch.ingressbandSwitch.value	uint64	数值大小 最小值:0 最大值:1099511627776
data.bandSwitch.egressbandSwitch	object	接口发送的带宽限制
data.bandSwitch.egressbandSwitch.value	uint64	数值大小 最小值:0 最大值:1099511627776
data.manage	object	本地策略控制，接口上控制是否放行对应类型的报文
data.manage.ssh	bool	是否启用ssh
data.manage.snmp	bool	是否启用snmp
data.manage.https	bool	是否启用https
data.manage.ping	bool	是否启用ping
data.manage.update	bool	是否启用系统升级维护
data.vlan	object	vlan配置
data.vlan.vlanMode	string	vlan模式 枚举值:[ACCESS:access模式 TRUNK:trunk模式]
data.vlan.accessId	uint32	access vlan id 最小值:0 最大值:4094
data.vlan.trunkConfig	object	trunk配置
data.vlan.trunkConfig.nativeId	int32	trunk native vlanid 最小值:0 最大值:4094

data.vlan.trunkConfig.trunkRange	object[]	trunk vlan 范围
data.vlan.trunkConfig.trunkRange.start	uint32	start vlan 最小值:1 最大值:4094
data.vlan.trunkConfig.trunkRange.end	uint32	start vlan 最小值:1 最大值:4094
data.vlanId	uint32	标识子接口和vlan口的vlanid 最小值:1 最大值:4094
data.physicalif	object	物理接口相关配置
data.physicalif.speedDuplex	object	速率双工配置
data.physicalif.speedDuplex.autoneg	bool	接口速率工作模式自动协商, 为true时speed和duplex不再生效
data.physicalif.speedDuplex.speed	uint32	接口速率, autoneg为false时speed配置生效 枚举值:[10:选择10Mbps 100:选择100Mbps 1000:选择1000Mbps 10000:选择10000Mbps]
data.physicalif.speedDuplex.duplex	string	双工模式, autoneg为false时duplex配置生效 枚举值:[half:半双工模式 full:全双工模式]
data.physicalif.bypassMirror	object	旁路镜像模式有效
data.physicalif.bypassMirror.statsEnable	bool	是否启用流量统计
data.physicalif.bypassMirror.ipGroup	string[]	关联的网络对象
data.subif	object	子接口相关配置
data.subif.fatherInterface	string	接口的父接口, 子接口时表示父接口
data.channelif	object	汇聚口相关配置
data.channelif.channelMode	string	汇聚口的工作模式配置 枚举值:[STATIC:静态 LACP:动态]
data.channelif.channelStatic	object	汇聚口静态模式下的相关配置

data.channelif.channelStatic.staticMode	string	模式选择 枚举值:[FAILOVER:故障转移 LOADBALANCE:负载均衡]
data.channelif.channelStatic.failover	object	故障转移
data.channelif.channelStatic.failover.preempt	bool	主备模式下的抢占或非抢占
data.channelif.channelStatic.failover.channelPrimary	string	聚合口主备模式的主接口，状态字段，配置无效
data.channelif.channelStatic.loadbalance	string	负载均衡的方法 枚举值:[ROUNDROBIN:平衡轮询策略 L3HASH:hash负载分担平衡策略]
data.channelif.channelLacp	object	动态模式下的相关配置
data.channelif.channelLacp.hashMode	string	哈希策略 枚举值:[MAC:源与目的IP 2TUPLE:源与目的IP、mac 3TUPLE:源与目的IP、mac、port]
data.channelif.channelLacp.negotiateMode	string	协商模式 枚举值:[INITIATIVE:源与目的IP PASSIVE:源与目的IP、mac]
data.channelif.ethSelect	string[]	选择的物理口列表

请求成功样例:

```
{
  "data": {
    "ifType": "PHYSICALIF",
    "wanEnable": false,
    "mtu": 1500,
    "bandSwitch": {
      "ingressbandSwitch": {
        "value": 1073741824
      },
      "egressbandSwitch": {
        "value": 1073741824
      }
    },
    "uuid": "0000000000000000000000000500000001",
    "physicalif": {
      "speedDuplex": {
        "duplex": "full",
        "autoneg": true,
        "speed": 1000
      }
    },
    "manage": {
      "ping": true,
      "snmp": false,
      "https": true,
      "ssh": false
    }
  }
}
```

```
    },
    "ipv4": {
      "staticIp": [],
      "ipv4Mode": "STATIC"
    },
    "mac": "FE:FC:FE:0B:D1:BB",
    "ipv6": {
      "ipv6Mode": "STATIC",
      "staticIp": []
    },
    "ifMode": "ROUTE",
    "name": "eth1",
    "shutdown": true,
    "description": ""
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3.2.6 全量修改指定接口信息

功能说明: 全量修改指定接口信息

1) API请求数据

PUT

https://{host}/api/v1/namespaces/@namespace/interfaces/@name

2) 输入参数

• Body参数

参数	类型	是否必选	描述
obj	object	是	接口信息
obj.name	string	是	接口名称
obj.description	string	否	接口描述 最小长度:0 最大长度:95
obj.mtu	int32	否	最大传输单元 最小值:68 最大值:1796 默认值:true
obj.ifType	string	是	表示接口类型 枚举值:[PHYSICALIF:物理口 SUBIF:子接口 CHANNELIF:汇聚口 VLANIF:vlan对应的3层接口]
obj.mac	string	否	接口的mac地址
obj.shutdown	bool	否	接口的启用或禁用 默认值:true

obj.ifMode	string	否	接口的使用类型 默认值:true 枚举值:[ROUTE:路由类型, 接口作为3层接口, 提供IP地址配置, 路由配置转发等功能 SWITCH:交换类型, 接口作为2层接口, 提供二层转发、vlan配置等功能 VIRTUALLINE:虚拟线类型, 接口作为虚拟线的一端, 透明传输报文 BYPASSMIRROR:旁路镜像类型, 该模式进来的报文, 只处理不转发]
obj.ipv4	object	否	接口IPv4相关配置
obj.ipv4.ipv4Mode	string	否	IPv4地址获取类型 枚举值:[STATIC:静态配置 DHCP:dhcp方式获取 PPPOE:pppoe方式获取]
obj.ipv4.staticIp	object[]	否	IPv4地址列表
obj.ipv4.staticIp.ipaddresses	object	是	IPv4地址 格式:ip-range
obj.ipv4.staticIp.ipaddresses.start	string	是	ip范围的起始位置 格式:ip-address
obj.ipv4.staticIp.ipaddresses.end	string	否	ip范围的终止位置 格式:ip-address
obj.ipv4.staticIp.ipaddresses.bits	uint8	否	ip掩码简写
obj.ipv4.staticIp.ipaddresses.mask	string	否	ip掩码 格式:ip-address
obj.ipv4.staticIp.isSync	bool	否	是否不同步到备机的选项
obj.ipv4.dhcp	object	否	dhcp4 client配置
obj.ipv4.dhcp.gateway	bool	否	是否设置dhcp gateway
obj.ipv4.pppoeGroupName	string	否	pppoe名称
obj.ipv6	object	否	接口IPv6相关配置
obj.ipv6.ipv6Mode	string	否	IPv6地址获取类型, 路由模式有效 枚举值:[STATIC:static IP DHCP6:dhcp]
obj.ipv6.staticIp	object[]	否	IPv6地址列表
obj.ipv6.staticIp.start	string	是	ip范围的起始位置 格式:ip-address
obj.ipv6.staticIp.end	string	否	ip范围的终止位置 格式:ip-address
obj.ipv6.staticIp.bits	uint8	否	ip掩码简写

obj.ipv6.staticIp.mask	string	否	ip掩码 格式:ip-address
obj.wanEnable	bool	否	是否开启接口的wan模式
obj.bandSwitch	object	否	接口接收的带宽限制
obj.bandSwitch.ingress bandSwitch	object	否	接口接收的带宽限制
obj.bandSwitch.ingress bandSwitch.value	uint64	否	数值大小 最小值:0 最大值:1099511627776 默认值:true
obj.bandSwitch.egressb andSwitch	object	否	接口发送的带宽限制
obj.bandSwitch.egressb andSwitch.value	uint64	否	数值大小 最小值:0 最大值:1099511627776 默认值:true
obj.manage	object	否	本地策略控制，接口上控制是否放行对应类型的报文
obj.manage.ssh	bool	否	是否启用ssh
obj.manage.snmp	bool	否	是否启用snmp
obj.manage.https	bool	是	是否启用https
obj.manage.ping	bool	否	是否启用ping 默认值:true
obj.manage.update	bool	否	是否启用系统升级维护
obj.vlan	object	否	vlan配置
obj.vlan.vlanMode	string	否	vlan模式 枚举值:[ACCESS:access模式 TRUNK:trunk模式]
obj.vlan.accessId	uint32	否	access vlan id 最小值:0 最大值:4094
obj.vlan.trunkConfig	object	否	trunk配置
obj.vlan.trunkConfig.nati vId	int32	否	trunk native vlanid 最小值:0 最大值:4094
obj.vlan.trunkConfig.tru nkRange	object[]	否	trunk vlan 范围
obj.vlan.trunkConfig.tru nkRange.start	uint32	是	start vlan 最小值:1 最大值:4094
obj.vlan.trunkConfig.tru nkRange.end	uint32	否	start vlan 最小值:1 最大值:4094

obj.vlanId	uint32	否	标识子接口和vlan口的vlanid 最小值:1 最大值:4094
obj.physicalif	object	否	物理接口相关配置
obj.physicalif.speedDuplex	object	否	速率双工配置
obj.physicalif.speedDuplex.autoneg	bool	否	接口速率工作模式自动协商，为true时speed和duplex不再生效 默认值:true
obj.physicalif.speedDuplex.speed	uint32	否	接口速率，autoneg为false时speed配置生效 默认值:true 枚举值:[10:选择10Mbps 100:选择100Mbps 1000:选择1000Mbps 10000:选择10000Mbps]
obj.physicalif.speedDuplex.duplex	string	否	双工模式，autoneg为false时duplex配置生效 默认值:true 枚举值:[half:半双工模式 full:全双工模式]
obj.physicalif.bypassMirror	object	否	旁路镜像模式有效
obj.physicalif.bypassMirror.statsEnable	bool	否	是否启用流量统计
obj.physicalif.bypassMirror.ipGroup	string[]	否	关联的网络对象
obj.subif	object	否	子接口相关配置
obj.subif.fatherInterface	string	是	接口的父接口，子接口时表示父接口
obj.channelif	object	否	汇聚口相关配置
obj.channelif.channelMode	string	否	汇聚口的工作模式配置 枚举值:[STATIC:静态 LACP:动态]
obj.channelif.channelStatic	object	否	汇聚口静态模式下的相关配置
obj.channelif.channelStatic.staticMode	string	否	模式选择 枚举值:[FAILOVER:故障转移 LOADBALANCE:负载均衡]
obj.channelif.channelStatic.failover	object	否	故障转移

obj.channelif.channelStatic.failover.preempt	bool	是	主备模式下的抢占或非抢占
obj.channelif.channelStatic.failover.channelPrimary	string	否	聚合口主备模式的主接口，状态字段，配置无效
obj.channelif.channelStatic.loadbalance	string	否	负载均衡的方法 枚举值:[ROUNDROBIN:平衡轮询策略 L3HASH:hash负载分担平衡策略]
obj.channelif.channelLacp	object	否	动态模式下的相关配置
obj.channelif.channelLacp.hashMode	string	否	哈希策略 枚举值:[MAC:源与目的IP 2TUPLE:源与目的IP、mac 3TUPLE:源与目的IP、mac、port]
obj.channelif.channelLacp.negotiateMode	string	否	协商模式 枚举值:[INITIATIVE:源与目的IP PASSIVE:源与目的IP、mac]
obj.channelif.ethSelect	string[]	否	选择的物理口列表

请求样例:

全量更新一个物理接口，更新名为eth1的物理接口，修改描述为put eth1，修改ipv4连接方式为dhcp，允许ssh连接，修改双工速率为全双工1000M

PUT https://192.168.1.1/api/v1/namespaces/public/interfaces/eth1

```
{
  "ifType": "PHYSICALIF",
  "wanEnable": false,
  "mtu": 1500,
  "bandSwitch": {
    "ingressbandSwitch": {
      "value": 1073741824
    },
    "egressbandSwitch": {
      "value": 1073741824
    }
  },
  "manage": {
    "ping": true,
    "https": true,
    "ssh": true,
    "snmp": false
  },
  "description": "put eth1",
  "shutdown": false,
  "mac": "FE:FC:FE:0B:D1:BB",
  "ipv6": {
    "ipv6Mode": "STATIC",
    "staticIp": []
  },
  "ipv4": {
    "ipv4Mode": "DHCP",
    "staticIp": [],
    "dhcp": {
      "gateway": true
    }
  },
  "ifMode": "ROUTE",
```

```

    "name": "eth1",
    "physicalif": {
      "speedDuplex": {
        "duplex": "full",
        "autoneg": false,
        "speed": 1000
      }
    }
  }
}

```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	接口信息
data.uuid	string	唯一标识id
data.name	string	接口名称
data.description	string	接口描述 最小长度:0 最大长度:95
data.mtu	int32	最大传输单元 最小值:68 最大值:1796
data.ifType	string	表示接口类型 枚举值:[PHYSICALIF:物理口 SUBIF:子接口 CHANNELIF:汇聚口 VLANIF:vlan对应的3层接口]
data.mac	string	接口的mac地址
data.shutdown	bool	接口的启用或禁用
data.ifMode	string	接口的使用类型 枚举值:[ROUTE:路由类型, 接口作为3层接口, 提供IP地址配置, 路由配置转发等功能 SWITCH:交换类型, 接口作为2层接口, 提供二层转发、vlan配置等功能 VIRTUALLINE:虚拟线类型, 接口作为虚拟线的一端, 透明传输报文 BYPASSMIRROR:旁路镜像类型, 该模式进来的报文, 只处理不转发]
data.ipv4	object	接口IPv4相关配置
data.ipv4.ipv4Mode	string	IPv4地址获取类型 枚举值:[STATIC:静态配置 DHCP:dhcp方式获取 PPPOE:pppoe方式获取]

data.ipv4.staticIp	object[]	IPv4地址列表
data.ipv4.staticIp.ipaddress	object	IPv4地址 格式:ip-range
data.ipv4.staticIp.ipaddress.start	string	ip范围的起始位置 格式:ip-address
data.ipv4.staticIp.ipaddress.end	string	ip范围的终止位置 格式:ip-address
data.ipv4.staticIp.ipaddress.bits	uint8	ip掩码简写
data.ipv4.staticIp.ipaddress.mask	string	ip掩码 格式:ip-address
data.ipv4.staticIp.isSync	bool	是否不同步到备机的选项
data.ipv4.dhcp	object	dhcp4 client配置
data.ipv4.dhcp.gateway	bool	是否设置dhcp gateway
data.ipv4.pppoeGroupName	string	pppoe名称
data.ipv6	object	接口IPv6相关配置
data.ipv6.ipv6Mode	string	IPv6地址获取类型, 路由模式有效 枚举值:[STATIC:static IP DHCP6:dhcp]
data.ipv6.staticIp	object[]	IPv6地址列表
data.ipv6.staticIp.start	string	ip范围的起始位置 格式:ip-address
data.ipv6.staticIp.end	string	ip范围的终止位置 格式:ip-address
data.ipv6.staticIp.bits	uint8	ip掩码简写
data.ipv6.staticIp.mask	string	ip掩码 格式:ip-address
data.wanEnable	bool	是否开启接口的wan模式
data.bandSwitch	object	接口接收的带宽限制
data.bandSwitch.ingressbandSwitch	object	接口接收的带宽限制
data.bandSwitch.ingressbandSwitch.value	uint64	数值大小 最小值:0 最大值:1099511627776
data.bandSwitch.egressbandSwitch	object	接口发送的带宽限制
data.bandSwitch.egressbandSwitch.value	uint64	数值大小 最小值:0 最大值:1099511627776
data.manage	object	本地策略控制, 接口上控制是否放行对应类型的报文
data.manage.ssh	bool	是否启用ssh

data.manage.snmp	bool	是否启用snmp
data.manage.https	bool	是否启用https
data.manage.ping	bool	是否启用ping
data.manage.update	bool	是否启用系统升级维护
data.vlan	object	vlan配置
data.vlan.vlanMode	string	vlan模式 枚举值:[ACCESS:access模式 TRUNK:trunk模式]
data.vlan.accessId	uint32	access vlan id 最小值:0 最大值:4094
data.vlan.trunkConfig	object	trunk配置
data.vlan.trunkConfig.nativeId	int32	trunk native vlanid 最小值:0 最大值:4094
data.vlan.trunkConfig.trunkRange	object[]	trunk vlan 范围
data.vlan.trunkConfig.trunkRange.start	uint32	start vlan 最小值:1 最大值:4094
data.vlan.trunkConfig.trunkRange.end	uint32	start vlan 最小值:1 最大值:4094
data.vlanId	uint32	标识子接口和vlan口的vlanid 最小值:1 最大值:4094
data.physicalif	object	物理接口相关配置
data.physicalif.speedDuplex	object	速率双工配置
data.physicalif.speedDuplex.autoneg	bool	接口速率工作模式自动协商, 为true时speed和duplex不再生效
data.physicalif.speedDuplex.speed	uint32	接口速率, autoneg为false时speed配置生效 枚举值:[10:选择10Mbps 100:选择100Mbps 1000:选择1000Mbps 10000:选择10000Mbps]
data.physicalif.speedDuplex.duplex	string	双工模式, autoneg为false时duplex配置生效 枚举值:[half:半双工模式 full:全双工模式]
data.physicalif.bypassMirror	object	旁路镜像模式有效

data.physicalif.bypassMirror.statsEnable	bool	是否启用流量统计
data.physicalif.bypassMirror.ipGroup	string[]	关联的网络对象
data.subif	object	子接口相关配置
data.subif.fatherInterface	string	接口的父接口，子接口时表示父接口
data.channelif	object	汇聚口相关配置
data.channelif.channelMode	string	汇聚口的工作模式配置 枚举值:[STATIC:静态 LACP:动态]
data.channelif.channelStatic	object	汇聚口静态模式下的相关配置
data.channelif.channelStatic.staticMode	string	模式选择 枚举值:[FAILOVER:故障转移 LOADBALANCE:负载均衡]
data.channelif.channelStatic.failover	object	故障转移
data.channelif.channelStatic.failover.preempt	bool	主备模式下的抢占或非抢占
data.channelif.channelStatic.failover.channelPrimary	string	聚合口主备模式的主接口，状态字段，配置无效
data.channelif.channelStatic.loadbalance	string	负载均衡的方法 枚举值:[ROUNDROBIN:平衡轮询策略 L3HASH:hash负载分担平衡策略]
data.channelif.channelLacp	object	动态模式下的相关配置
data.channelif.channelLacp.hashMode	string	哈希策略 枚举值:[MAC:源与目的IP 2TUPLE:源与目的IP、mac 3TUPLE:源与目的IP、mac、port]
data.channelif.channelLacp.negotiateMode	string	协商模式 枚举值:[INITIATIVE:源与目的IP PASSIVE:源与目的IP、mac]
data.channelif.ethSelect	string[]	选择的物理口列表

请求成功样例：

```
{
  "data": {
    "ifType": "PHYSICALIF",
    "wanEnable": false,
    "mtu": 1500,
    "bandSwitch": {
      "ingressbandSwitch": {
        "value": 1073741824
      },
      "egressbandSwitch": {
        "value": 1073741824
      }
    },
    "uuid": "0000000000000000000000005000000001",
    "manage": {
      "ping": true,
      "https": true,
      "ssh": true,
      "snmp": false
    },
    "ipv4": {
      "ipv4Mode": "DHCP",
      "staticIp": [],
      "dhcp": {
        "gateway": true
      }
    },
    "description": "put eth1",
    "mac": "FE:FC:FE:0B:D1:BB",
    "ipv6": {
      "ipv6Mode": "STATIC",
      "staticIp": []
    },
    "ifMode": "ROUTE",
    "name": "eth1",
    "shutdown": false,
    "physicalif": {
      "speedDuplex": {
        "duplex": "full",
        "autoneg": false,
        "speed": 1000
      }
    }
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3.2.7 批量删除接口

功能说明：批量删除接口

1) API请求数据

POST

```
https://{host}/api/batch/v1/namespaces/@namespace/interfaces?_method=delete
```

2) 输入参数

- Body参数

参数	类型	是否必选	描述
obj	object[]	是	接口列表

obj.name	string	是	接口名称
obj.description	string	否	接口描述 最小长度:0 最大长度:95
obj.mtu	int32	否	最大传输单元 最小值:68 最大值:1796 默认值:true
obj.ifType	string	是	表示接口类型 枚举值:[PHYSICALIF:物理口 SUBIF:子接口 CHANNELIF:汇聚口 VLANIF:vlan对应的3层接口]
obj.mac	string	否	接口的mac地址
obj.shutdown	bool	否	接口的启用或禁用 默认值:true
obj.ifMode	string	否	接口的使用类型 默认值:true 枚举值:[ROUTE:路由类型, 接口作为3层接口, 提供IP地址配置, 路由配置转发等功能 SWITCH:交换类型, 接口作为2层接口, 提供二层转发、vlan配置等功能 VIRTUALLINE:虚拟线类型, 接口作为虚拟线的一端, 透明传输报文 BYPASSMIRROR:旁路镜像类型, 该模式进来的报文, 只处理不转发]
obj.ipv4	object	否	接口IPv4相关配置
obj.ipv4.ipv4Mode	string	否	IPv4地址获取类型 枚举值:[STATIC:静态配置 DHCP:dhcp方式获取 PPPOE:pppoe方式获取]
obj.ipv4.staticIp	object[]	否	IPv4地址列表
obj.ipv4.staticIp.ipaddress	object	是	IPv4地址 格式:ip-range
obj.ipv4.staticIp.ipaddress.start	string	是	ip范围的起始位置 格式:ip-address
obj.ipv4.staticIp.ipaddress.end	string	否	ip范围的终止位置 格式:ip-address
obj.ipv4.staticIp.ipaddress.bits	uint8	否	ip掩码简写
obj.ipv4.staticIp.ipaddress.mask	string	否	ip掩码 格式:ip-address
obj.ipv4.staticIp.isSync	bool	否	是否不同步到备机的选项

obj.ipv4.dhcp	object	否	dhcp4 client配置
obj.ipv4.dhcp.gateway	bool	否	是否设置dhcp gateway
obj.ipv4.pppoeGroupName	string	否	pppoe名称
obj.ipv6	object	否	接口IPv6相关配置
obj.ipv6.ipv6Mode	string	否	IPv6地址获取类型，路由模式有效枚举值:[STATIC:static IP DHCP6:dhcp]
obj.ipv6.staticIp	object[]	否	IPv6地址列表
obj.ipv6.staticIp.start	string	是	ip范围的起始位置 格式:ip-address
obj.ipv6.staticIp.end	string	否	ip范围的终止位置 格式:ip-address
obj.ipv6.staticIp.bits	uint8	否	ip掩码简写
obj.ipv6.staticIp.mask	string	否	ip掩码 格式:ip-address
obj.wanEnable	bool	否	是否开启接口的wan模式
obj.bandSwitch	object	否	接口接收的带宽限制
obj.bandSwitch.ingressbandSwitch	object	否	接口接收的带宽限制
obj.bandSwitch.ingressbandSwitch.value	uint64	否	数值大小 最小值:0 最大值:1099511627776 默认值:true
obj.bandSwitch.egressbandSwitch	object	否	接口发送的带宽限制
obj.bandSwitch.egressbandSwitch.value	uint64	否	数值大小 最小值:0 最大值:1099511627776 默认值:true
obj.manage	object	否	本地策略控制，接口上控制是否放行对应类型的报文
obj.manage.ssh	bool	否	是否启用ssh
obj.manage.snmp	bool	否	是否启用snmp
obj.manage.https	bool	是	是否启用https
obj.manage.ping	bool	否	是否启用ping 默认值:true
obj.manage.update	bool	否	是否启用系统升级维护
obj.vlan	object	否	vlan配置

obj.vlan.vlanMode	string	否	vlan模式 枚举值:[ACCESS:access模式 TRUNK:trunk模式]
obj.vlan.accessId	uint32	否	access vlan id 最小值:0 最大值:4094
obj.vlan.trunkConfig	object	否	trunk配置
obj.vlan.trunkConfig.nativeVid	int32	否	trunk native vlanid 最小值:0 最大值:4094
obj.vlan.trunkConfig.trunkRange	object[]	否	trunk vlan 范围
obj.vlan.trunkConfig.trunkRange.start	uint32	是	start vlan 最小值:1 最大值:4094
obj.vlan.trunkConfig.trunkRange.end	uint32	否	start vlan 最小值:1 最大值:4094
obj.vlanId	uint32	否	标识子接口和vlan口的vlanid 最小值:1 最大值:4094
obj.physicalif	object	否	物理接口相关配置
obj.physicalif.speedDuplex	object	否	速率双工配置
obj.physicalif.speedDuplex.autoneg	bool	否	接口速率工作模式自动协商, 为true时speed和duplex不再生效 默认值:true
obj.physicalif.speedDuplex.speed	uint32	否	接口速率, autoneg为false时speed配置生效 默认值:true 枚举值:[10:选择10Mbps 100:选择100Mbps 1000:选择1000Mbps 10000:选择10000Mbps]
obj.physicalif.speedDuplex.duplex	string	否	双工模式, autoneg为false时duplex配置生效 默认值:true 枚举值:[half:半双工模式 full:全双工模式]
obj.physicalif.bypassMirror	object	否	旁路镜像模式有效
obj.physicalif.bypassMirror.statsEnable	bool	否	是否启用流量统计

obj.physicalif.bypassMirror.ipGroup	string[]	否	关联的网络对象
obj.subif	object	否	子接口相关配置
obj.subif.fatherInterface	string	是	接口的父接口，子接口时表示父接口
obj.channelif	object	否	汇聚口相关配置
obj.channelif.channelMode	string	否	汇聚口的工作模式配置 枚举值:[STATIC:静态 LACP:动态]
obj.channelif.channelStatic	object	否	汇聚口静态模式下的相关配置
obj.channelif.channelStatic.staticMode	string	否	模式选择 枚举值:[FAILOVER:故障转移 LOADBALANCE:负载均衡]
obj.channelif.channelStatic.failover	object	否	故障转移
obj.channelif.channelStatic.failover.preempt	bool	是	主备模式下的抢占或非抢占
obj.channelif.channelStatic.failover.channelPrimary	string	否	聚合口主备模式的主接口，状态字段，配置无效
obj.channelif.channelStatic.loadbalance	string	否	负载均衡的方法 枚举值:[ROUNDROBIN:平衡轮询策略 L3HASH:hash负载分担平衡策略]
obj.channelif.channelLacp	object	否	动态模式下的相关配置
obj.channelif.channelLacp.hashMode	string	否	哈希策略 枚举值:[MAC:源与目的IP 2TUPLE:源与目的IP、mac 3TUPLE:源与目的IP、mac、port]
obj.channelif.channelLacp.negotiateMode	string	否	协商模式 枚举值:[INITIATIVE:源与目的IP PASSIVE:源与目的IP、mac]
obj.channelif.ethSelect	string[]	否	选择的物理口列表

请求样例:

批量删除多个接口，修改名为eth0.1，eth0.2的子接口

```
POST https://192.168.1.1/api/batch/v1/namespaces/public/interfaces?_method=DELETE
[
{
```

```

    "name": "eth0.1"
  },
  {
    "name": "eth0.2"
  }
]

```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	接口列表
data.uuid	string	唯一标识id
data.name	string	接口名称
data.description	string	接口描述 最小长度:0 最大长度:95
data.mtu	int32	最大传输单元 最小值:68 最大值:1796
data.ifType	string	表示接口类型 枚举值:[PHYSICALIF:物理口 SUBIF:子接口 CHANNELIF:汇聚口 VLANIF:vlan对应的3层接口]
data.mac	string	接口的mac地址
data.shutdown	bool	接口的启用或禁用
data.ifMode	string	接口的使用类型 枚举值:[ROUTE:路由类型，接口作为3层接口，提供IP地址配置，路由配置转发等功能 SWITCH:交换类型，接口作为2层接口，提供二层转发、vlan配置等功能 VIRTUALLINE:虚拟线类型，接口作为虚拟线的一端，透明传输报文 BYPASSMIRROR:旁路镜像类型，该模式进来的报文，只处理不转发]
data.ipv4	object	接口IPv4相关配置
data.ipv4.ipv4Mode	string	IPv4地址获取类型 枚举值:[STATIC:静态配置 DHCP:dhcp方式获取 PPPOE:pppoe方式获取]
data.ipv4.staticIp	object[]	IPv4地址列表

data.ipv4.staticIp.ipaddress	object	IPv4地址 格式:ip-range
data.ipv4.staticIp.ipaddress.start	string	ip范围的起始位置 格式:ip-address
data.ipv4.staticIp.ipaddress.end	string	ip范围的终止位置 格式:ip-address
data.ipv4.staticIp.ipaddress.bits	uint8	ip掩码简写
data.ipv4.staticIp.ipaddress.mask	string	ip掩码 格式:ip-address
data.ipv4.staticIp.isSync	bool	是否不同步到备机的选项
data.ipv4.dhcp	object	dhcp4 client配置
data.ipv4.dhcp.gateway	bool	是否设置dhcp gateway
data.ipv4.pppoeGroupName	string	pppoe名称
data.ipv6	object	接口IPv6相关配置
data.ipv6.ipv6Mode	string	IPv6地址获取类型，路由模式有效 枚举值:[STATIC:static IP DHCP6:dhcp]
data.ipv6.staticIp	object[]	IPv6地址列表
data.ipv6.staticIp.start	string	ip范围的起始位置 格式:ip-address
data.ipv6.staticIp.end	string	ip范围的终止位置 格式:ip-address
data.ipv6.staticIp.bits	uint8	ip掩码简写
data.ipv6.staticIp.mask	string	ip掩码 格式:ip-address
data.wanEnable	bool	是否开启接口的wan模式
data.bandSwitch	object	接口接收的带宽限制
data.bandSwitch.ingressbandSwitch	object	接口接收的带宽限制
data.bandSwitch.ingressbandSwitch.value	uint64	数值大小 最小值:0 最大值:1099511627776
data.bandSwitch.egressbandSwitch	object	接口发送的带宽限制
data.bandSwitch.egressbandSwitch.value	uint64	数值大小 最小值:0 最大值:1099511627776
data.manage	object	本地策略控制，接口上控制是否放行对应类型的报文
data.manage.ssh	bool	是否启用ssh
data.manage.snmp	bool	是否启用snmp

data.manage.https	bool	是否启用https
data.manage.ping	bool	是否启用ping
data.manage.update	bool	是否启用系统升级维护
data.vlan	object	vlan配置
data.vlan.vlanMode	string	vlan模式 枚举值:[ACCESS:access模式 TRUNK:trunk模式]
data.vlan.accessId	uint32	access vlan id 最小值:0 最大值:4094
data.vlan.trunkConfig	object	trunk配置
data.vlan.trunkConfig.nativeVid	int32	trunk native vlanid 最小值:0 最大值:4094
data.vlan.trunkConfig.trunkRange	object[]	trunk vlan 范围
data.vlan.trunkConfig.trunkRange.start	uint32	start vlan 最小值:1 最大值:4094
data.vlan.trunkConfig.trunkRange.end	uint32	start vlan 最小值:1 最大值:4094
data.vlanId	uint32	标识子接口和vlan口的vlanid 最小值:1 最大值:4094
data.physicalif	object	物理接口相关配置
data.physicalif.speedDuplex	object	速率双工配置
data.physicalif.speedDuplex.autoneg	bool	接口速率工作模式自动协商, 为true时speed和duplex不再生效
data.physicalif.speedDuplex.speed	uint32	接口速率, autoneg为false时speed配置生效 枚举值:[10:选择10Mbps 100:选择100Mbps 1000:选择1000Mbps 10000:选择10000Mbps]
data.physicalif.speedDuplex.duplex	string	双工模式, autoneg为false时duplex配置生效 枚举值:[half:半双工模式 full:全双工模式]
data.physicalif.bypassMirror	object	旁路镜像模式有效

data.physicalif.bypassMirror.statsEnable	bool	是否启用流量统计
data.physicalif.bypassMirror.ipGroup	string[]	关联的网络对象
data.subif	object	子接口相关配置
data.subif.fatherInterface	string	接口的父接口，子接口时表示父接口
data.channelif	object	汇聚口相关配置
data.channelif.channelMode	string	汇聚口的工作模式配置 枚举值:[STATIC:静态 LACP:动态]
data.channelif.channelStatic	object	汇聚口静态模式下的相关配置
data.channelif.channelStatic.staticMode	string	模式选择 枚举值:[FAILOVER:故障转移 LOADBALANCE:负载均衡]
data.channelif.channelStatic.failover	object	故障转移
data.channelif.channelStatic.failover.preempt	bool	主备模式下的抢占或非抢占
data.channelif.channelStatic.failover.channelPrimary	string	聚合口主备模式的主接口，状态字段，配置无效
data.channelif.channelStatic.loadbalance	string	负载均衡的方法 枚举值:[ROUNDROBIN:平衡轮询策略 L3HASH:hash负载分担平衡策略]
data.channelif.channelLacp	object	动态模式下的相关配置
data.channelif.channelLacp.hashMode	string	哈希策略 枚举值:[MAC:源与目的IP 2TUPLE:源与目的IP、mac 3TUPLE:源与目的IP、mac、port]
data.channelif.channelLacp.negotiateMode	string	协商模式 枚举值:[INITIATIVE:源与目的IP PASSIVE:源与目的IP、mac]
data.channelif.ethSelect	string[]	选择的物理口列表

请求成功样例：


```
{
  "data": [
    {
      "ifType": "SUBIF",
      "mtu": 1500,
      "vlanId": 1,
      "uuid": "0000000000000000000000000600000001",
      "subif": {
        "fatherInterface": "eth0"
      },
      "ipv4": {
        "pppoeGroupName": null,
        "ipv4Mode": "STATIC",
        "staticIp": [],
        "dhcp": null
      },
      "manage": {
        "ping": true,
        "snmp": false,
        "https": true,
        "ssh": false
      },
      "name": "eth0.1",
      "description": ""
    },
    {
      "ifType": "SUBIF",
      "mtu": 1500,
      "vlanId": 2,
      "uuid": "0000000000000000000000000600000002",
      "subif": {
        "fatherInterface": "eth0"
      },
      "ipv4": {
        "pppoeGroupName": null,
        "ipv4Mode": "STATIC",
        "staticIp": [],
        "dhcp": null
      },
      "manage": {
        "ping": true,
        "snmp": false,
        "https": true,
        "ssh": false
      },
      "name": "eth0.2",
      "description": ""
    }
  ],
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3.2.8 批量修改接口信息

功能说明: 批量修改接口信息

1) API请求数据

PATCH <https://{host}/api/batch/v1/namespaces/@namespace/interfaces>

2) 输入参数

• Body参数

参数	类型	是否必选	描述
obj	object[]	是	接口列表
obj.name	string	是	接口名称
obj.description	string	否	接口描述 最小长度:0 最大长度:95
obj.mtu	int32	否	最大传输单元 最小值:68 最大值:1796 默认值:true
obj.ifType	string	是	表示接口类型 枚举值:[PHYSICALIF:物理口 SUBIF:子接口 CHANNELIF:汇聚口 VLANIF:vlan对应的3层接口]
obj.mac	string	否	接口的mac地址
obj.shutdown	bool	否	接口的启用或禁用 默认值:true
obj.ifMode	string	否	接口的使用类型 默认值:true 枚举值:[ROUTE:路由类型, 接口作为3层接口, 提供IP地址配置, 路由配置转发等功能 SWITCH:交换类型, 接口作为2层接口, 提供二层转发、vlan配置等功能 VIRTUALLINE:虚拟线类型, 接口作为虚拟线的一端, 透明传输报文 BYPASSMIRROR:旁路镜像类型, 该模式进来的报文, 只处理不转发]
obj.ipv4	object	否	接口IPv4相关配置
obj.ipv4.ipv4Mode	string	否	IPv4地址获取类型 枚举值:[STATIC:静态配置 DHCP:dhcp方式获取 PPPOE:pppoe方式获取]
obj.ipv4.staticIp	object[]	否	IPv4地址列表
obj.ipv4.staticIp.ipaddress	object	是	IPv4地址 格式:ip-range
obj.ipv4.staticIp.ipaddress.start	string	是	ip范围的起始位置 格式:ip-address
obj.ipv4.staticIp.ipaddress.end	string	否	ip范围的终止位置 格式:ip-address

obj.ipv4.staticIp.address.bits	uint8	否	ip掩码简写
obj.ipv4.staticIp.address.mask	string	否	ip掩码 格式:ip-address
obj.ipv4.staticIp.isSync	bool	否	是否不同步到备机的选项
obj.ipv4.dhcp	object	否	dhcp4 client配置
obj.ipv4.dhcp.gateway	bool	否	是否设置dhcp gateway
obj.ipv4.pppoeGroupName	string	否	pppoe名称
obj.ipv6	object	否	接口IPv6相关配置
obj.ipv6.ipv6Mode	string	否	IPv6地址获取类型，路由模式有效 枚举值:[STATIC:static IP DHCP6:dhcp]
obj.ipv6.staticIp	object[]	否	IPv6地址列表
obj.ipv6.staticIp.start	string	是	ip范围的起始位置 格式:ip-address
obj.ipv6.staticIp.end	string	否	ip范围的终止位置 格式:ip-address
obj.ipv6.staticIp.bits	uint8	否	ip掩码简写
obj.ipv6.staticIp.mask	string	否	ip掩码 格式:ip-address
obj.wanEnable	bool	否	是否开启接口的wan模式
obj.bandSwitch	object	否	接口接收的带宽限制
obj.bandSwitch.ingressBandSwitch	object	否	接口接收的带宽限制
obj.bandSwitch.ingressBandSwitch.value	uint64	否	数值大小 最小值:0 最大值:1099511627776 默认值:true
obj.bandSwitch.egressBandSwitch	object	否	接口发送的带宽限制
obj.bandSwitch.egressBandSwitch.value	uint64	否	数值大小 最小值:0 最大值:1099511627776 默认值:true
obj.manage	object	否	本地策略控制，接口上控制是否放行对应类型的报文
obj.manage.ssh	bool	否	是否启用ssh
obj.manage.snmp	bool	否	是否启用snmp
obj.manage.https	bool	是	是否启用https
obj.manage.ping	bool	否	是否启用ping 默认值:true

obj.manage.update	bool	否	是否启用系统升级维护
obj.vlan	object	否	vlan配置
obj.vlan.vlanMode	string	否	vlan模式 枚举值:[ACCESS:access模式 TRUNK:trunk模式]
obj.vlan.accessId	uint32	否	access vlan id 最小值:0 最大值:4094
obj.vlan.trunkConfig	object	否	trunk配置
obj.vlan.trunkConfig.nativeVid	int32	否	trunk native vlanid 最小值:0 最大值:4094
obj.vlan.trunkConfig.trunkRange	object[]	否	trunk vlan 范围
obj.vlan.trunkConfig.trunkRange.start	uint32	是	start vlan 最小值:1 最大值:4094
obj.vlan.trunkConfig.trunkRange.end	uint32	否	start vlan 最小值:1 最大值:4094
obj.vlanId	uint32	否	标识子接口和vlan口的vlanid 最小值:1 最大值:4094
obj.physicalif	object	否	物理接口相关配置
obj.physicalif.speedDuplex	object	否	速率双工配置
obj.physicalif.speedDuplex.autoneg	bool	否	接口速率工作模式自动协商, 为true时speed和duplex不再生效 默认值:true
obj.physicalif.speedDuplex.speed	uint32	否	接口速率, autoneg为false时speed配置生效 默认值:true 枚举值:[10:选择10Mbps 100:选择100Mbps 1000:选择1000Mbps 10000:选择10000Mbps]
obj.physicalif.speedDuplex.duplex	string	否	双工模式, autoneg为false时duplex配置生效 默认值:true 枚举值:[half:半双工模式 full:全双工模式]
obj.physicalif.bypassMirror	object	否	旁路镜像模式有效

obj.physicalif.bypassMirror.statsEnable	bool	否	是否启用流量统计
obj.physicalif.bypassMirror.ipGroup	string[]	否	关联的网络对象
obj.subif	object	否	子接口相关配置
obj.subif.fatherInterface	string	是	接口的父接口，子接口时表示父接口
obj.channelif	object	否	汇聚口相关配置
obj.channelif.channelMode	string	否	汇聚口的工作模式配置 枚举值:[STATIC:静态 LACP:动态]
obj.channelif.channelStatic	object	否	汇聚口静态模式下的相关配置
obj.channelif.channelStatic.staticMode	string	否	模式选择 枚举值:[FAILOVER:故障转移 LOADBALANCE:负载均衡]
obj.channelif.channelStatic.failover	object	否	故障转移
obj.channelif.channelStatic.failover.preempt	bool	是	主备模式下的抢占或非抢占
obj.channelif.channelStatic.failover.channelPrimary	string	否	聚合口主备模式的主接口，状态字段，配置无效
obj.channelif.channelStatic.loadbalance	string	否	负载均衡的方法 枚举值:[ROUNDROBIN:平衡轮询策略 L3HASH:hash负载分担平衡策略]
obj.channelif.channelLacp	object	否	动态模式下的相关配置
obj.channelif.channelLacp.hashMode	string	否	哈希策略 枚举值:[MAC:源与目的IP 2TUPLE:源与目的IP、mac 3TUPLE:源与目的IP、mac、port]
obj.channelif.channelLacp.negotiateMode	string	否	协商模式 枚举值:[INITIATIVE:源与目的IP PASSIVE:源与目的IP、mac]
obj.channelif.ethSelect	string[]	否	选择的物理口列表

请求样例：

批量修改多个接口，修改名为eth2，eth3的物理接口为禁用，其余不改

```
PATCH https://192.168.1.1/api/batch/v1/namespaces/public/interfaces
[
  {
    "name": "eth2",
    "shutdown": true
  },
  {
    "name": "eth3",
    "shutdown": true
  }
]
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	接口列表
data.uuid	string	唯一标识id
data.name	string	接口名称
data.description	string	接口描述 最小长度:0 最大长度:95
data.mtu	int32	最大传输单元 最小值:68 最大值:1796
data.ifType	string	表示接口类型 枚举值:[PHYSICALIF:物理口 SUBIF:子接口 CHANNELIF:汇聚口 VLANIF:vlan对应的3层接口]
data.mac	string	接口的mac地址
data.shutdown	bool	接口的启用或禁用
data.ifMode	string	接口的使用类型 枚举值:[ROUTE:路由类型，接口作为3层接口，提供IP地址配置，路由配置转发等功能 SWITCH:交换类型，接口作为2层接口，提供二层转发、vlan配置等功能 VIRTUALLINE:虚拟线类型，接口作为虚拟线的一端，透明传输报文 BYPASSMIRROR:旁路镜像类型，该模式进来的报文，只处理不转发]
data.ipv4	object	接口IPv4相关配置

data.ipv4.ipv4Mode	string	IPv4地址获取类型 枚举值:[STATIC:静态配置 DHCP:dhcp方式获取 PPPOE:pppoe方式获取]
data.ipv4.staticIp	object[]	IPv4地址列表
data.ipv4.staticIp.ipaddress	object	IPv4地址 格式:ip-range
data.ipv4.staticIp.ipaddress.start	string	ip范围的起始位置 格式:ip-address
data.ipv4.staticIp.ipaddress.end	string	ip范围的终止位置 格式:ip-address
data.ipv4.staticIp.ipaddress.bits	uint8	ip掩码简写
data.ipv4.staticIp.ipaddress.mask	string	ip掩码 格式:ip-address
data.ipv4.staticIp.isSync	bool	是否不同步到备机的选项
data.ipv4.dhcp	object	dhcp4 client配置
data.ipv4.dhcp.gateway	bool	是否设置dhcp gateway
data.ipv4.pppoeGroupName	string	pppoe名称
data.ipv6	object	接口IPv6相关配置
data.ipv6.ipv6Mode	string	IPv6地址获取类型，路由模式有效 枚举值:[STATIC:static IP DHCP6:dhcp]
data.ipv6.staticIp	object[]	IPv6地址列表
data.ipv6.staticIp.start	string	ip范围的起始位置 格式:ip-address
data.ipv6.staticIp.end	string	ip范围的终止位置 格式:ip-address
data.ipv6.staticIp.bits	uint8	ip掩码简写
data.ipv6.staticIp.mask	string	ip掩码 格式:ip-address
data.wanEnable	bool	是否开启接口的wan模式
data.bandSwitch	object	接口接收的带宽限制
data.bandSwitch.ingressbandSwitch	object	接口接收的带宽限制
data.bandSwitch.ingressbandSwitch.value	uint64	数值大小 最小值:0 最大值:1099511627776
data.bandSwitch.egressbandSwitch	object	接口发送的带宽限制

data.bandSwitch.egressbandSwitch.value	uint64	数值大小 最小值:0 最大值:1099511627776
data.manage	object	本地策略控制，接口上控制是否放行对应类型的报文
data.manage.ssh	bool	是否启用ssh
data.manage.snmp	bool	是否启用snmp
data.manage.https	bool	是否启用https
data.manage.ping	bool	是否启用ping
data.manage.update	bool	是否启用系统升级维护
data.vlan	object	vlan配置
data.vlan.vlanMode	string	vlan模式 枚举值:[ACCESS:access模式 TRUNK:trunk模式]
data.vlan.accessId	uint32	access vlan id 最小值:0 最大值:4094
data.vlan.trunkConfig	object	trunk配置
data.vlan.trunkConfig.nativeVid	int32	trunk native vlanid 最小值:0 最大值:4094
data.vlan.trunkConfig.trunkRange	object[]	trunk vlan 范围
data.vlan.trunkConfig.trunkRange.start	uint32	start vlan 最小值:1 最大值:4094
data.vlan.trunkConfig.trunkRange.end	uint32	start vlan 最小值:1 最大值:4094
data.vlanId	uint32	标识子接口和vlan口的vlanid 最小值:1 最大值:4094
data.physicalif	object	物理接口相关配置
data.physicalif.speedDuplex	object	速率双工配置
data.physicalif.speedDuplex.autoneg	bool	接口速率工作模式自动协商，为true时speed和duplex不再生效
data.physicalif.speedDuplex.speed	uint32	接口速率，autoneg为false时speed配置生效 枚举值:[10:选择10Mbps 100:选择100Mbps 1000:选择1000Mbps 10000:选择10000Mbps]

data.physicalif.speedDuplex.duplex	string	双工模式，autoneg为false时duplex配置生效 枚举值:[half:半双工模式 full:全双工模式]
data.physicalif.bypassMirror	object	旁路镜像模式有效
data.physicalif.bypassMirror.statsEnable	bool	是否启用流量统计
data.physicalif.bypassMirror.ipGroup	string[]	关联的网络对象
data.subif	object	子接口相关配置
data.subif.fatherInterface	string	接口的父接口，子接口时表示父接口
data.channelif	object	汇聚口相关配置
data.channelif.channelMode	string	汇聚口的工作模式配置 枚举值:[STATIC:静态 LACP:动态]
data.channelif.channelStatic	object	汇聚口静态模式下的相关配置
data.channelif.channelStatic.staticMode	string	模式选择 枚举值:[FAILOVER:故障转移 LOADBALANCE:负载均衡]
data.channelif.channelStatic.failover	object	故障转移
data.channelif.channelStatic.failover.preempt	bool	主备模式下的抢占或非抢占
data.channelif.channelStatic.failover.channelPrimary	string	聚合口主备模式的主接口，状态字段，配置无效
data.channelif.channelStatic.loadbalance	string	负载均衡的方法 枚举值:[ROUNDROBIN:平衡轮询策略 L3HASH:hash负载分担平衡策略]
data.channelif.channelLacp	object	动态模式下的相关配置
data.channelif.channelLacp.hashMode	string	哈希策略 枚举值:[MAC:源与目的IP 2TUPLE:源与目的IP、mac 3TUPLE:源与目的IP、mac、port]

data.channelif.channelL acp.negotiateMode	string	协商模式 枚举值:[INITIATIVE:源与目的IP PASSIVE:源与目的IP、mac]
data.channelif.ethSelect	string[]	选择的物理口列表

请求成功样例:

```
{
  "data": [
    {
      "ifType": "PHYSICALIF",
      "wanEnable": false,
      "mtu": 1500,
      "uuid": "000000000000000000000000500000002",
      "mac": "78:88:22:33:44:55",
      "physicalif": {
        "speedDuplex": {
          "duplex": "full",
          "autoneg": true,
          "speed": 1000
        }
      },
      "ifMode": "VIRTUALLINE",
      "name": "eth2",
      "shutdown": true,
      "description": ""
    },
    {
      "ifType": "PHYSICALIF",
      "wanEnable": false,
      "mtu": 1500,
      "bandSwitch": {
        "ingressbandSwitch": {
          "value": 1073741824
        },
        "egressbandSwitch": {
          "value": 1073741824
        }
      },
      "uuid": "000000000000000000000000500000003",
      "physicalif": {
        "speedDuplex": {
          "duplex": "full",
          "autoneg": true,
          "speed": 1000
        }
      },
      "manage": {
        "ping": true,
        "snmp": false,
        "https": true,
        "ssh": false
      },
      "ipv4": {
        "ipv4Mode": "DHCP",
        "staticip": [],
        "dhcp": {
          "gateway": false
        }
      },
      "mac": "FE:FC:FE:AC:32:88",
      "ipv6": {
        "ipv6Mode": "STATIC",
        "staticip": []
      },
      "ifMode": "ROUTE",
      "name": "eth3",
      "shutdown": true,
      "description": ""
    }
  ],
  "code": 0,
```

```
    "message": "成功"
  }
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3.2.9 获取指定接口默认mac

功能说明: 获取指定接口默认mac

1) API请求数据

GET `https://{host}/api/v1/namespaces/@namespace/defaultmac/@name`

2) 输入参数

- URL 参数

参数	类型	是否必选	描述
_select	string	否	选择字段, 字段用逗号分隔 最大长度:1000

请求样例:

GET `https://192.168.1.1/api/v1/namespaces/public/defaultmac/eth1`

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	单个接口的默认mac
data.name	string	接口名称 最小长度:1 最大长度:95
data.defaultMac	string	接口默认mac值
data.driver	string	网卡驱动类型 最大长度:95

请求成功样例:

```
{
  "data": {
    "defaultMac": "FE:FC:FE:8D:67:C6",
    "name": "eth1",
    "driver": "e1000"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,

```

```
"message": "无效参数"
}
```

3.2.10 获取所有接口默认mac

功能说明：获取所有物理口默认mac

1) API请求数据

GET https://{host}/api/v1/namespaces/@namespace/defaultmac

2) 输入参数

- URL 参数

参数	类型	是否必选	描述
_select	string	否	选择字段, 字段用逗号分隔 最大长度:1000

请求样例：

获取多个物理接口的默认MAC及网卡驱动信息

GET https://192.168.1.1/api/v1/namespaces/public/defaultmac

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	分页数据
data.totalItems	int32	总共多少项目
data.totalPages	int32	总共多少页
data.pageNumber	int32	当前页码, 从 1 开始
data.pageSize	int32	每页多大
data.itemsOffset	int32	当前条目偏移, 从 0 开始
data.itemLength	int32	数据列表长度
data.items	object[]	有效数据列表
data.items.name	string	接口名称 最小长度:1 最大长度:95
data.items.defaultMac	string	接口默认mac值
data.items.driver	string	网卡驱动类型 最大长度:95

请求成功样例：

```
{
  "data": {
    "itemsOffset": 0,
    "items": [
      {
        "defaultMac": "FE:FC:FE:43:B6:4C",
        "name": "eth0",

```

```
        "driver": "e1000"
      },
      {
        "defaultMac": "FE:FC:FE:8D:67:C6",
        "name": "eth1",
        "driver": "e1000"
      },
      {
        "defaultMac": "FE:FC:FE:A1:0F:63",
        "name": "eth2",
        "driver": "e1000"
      },
      {
        "defaultMac": "FE:FC:FE:AC:32:88",
        "name": "eth3",
        "driver": "e1000"
      }
    ],
    "totalItems": 4,
    "pageNumber": 1,
    "itemLength": 4,
    "totalPages": 1,
    "pageSize": 100
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3.3 PPPOE接口

3.3.1 获取全部PPPOE接口配置信息

功能说明: 获取全部PPPOE接口配置信息

1) API请求数据

GET

https://{host}/api/v1/namespaces/@namespace/pppoe

2) 输入参数

- URL 参数

参数	类型	是否必选	描述
_start	int32	否	起始位置, 从 0 开始 最小值:0 最大值:2000000
_search	string	否	模糊搜索关键字 最大长度:95
_select	string	否	选择字段, 字段用逗号分隔 最大长度:1000
_order	string	否	指定排序方式 (正序/倒序) 默认值:true 枚举值:[asc:按升序排列 desc:按降序排列]

_length	int32	否	获取最大长度，限制最大值，如果数据不够可能返回小于 length 最小值:1 最大值:200 默认值:true
_sortby	string	否	指定排序字段 最大长度:100

请求样例:

获取多个pppoe配置信息，查询条件：起始位置为0，最多展示100条

GET https://192.168.1.1/api/v1/namespaces/public/pppoe?_start=0&_length=100

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	分页数据
data.totalItems	int32	总共多少项目
data.totalPages	int32	总共多少页
data.pageNumber	int32	当前页码，从 1 开始
data.pageSize	int32	每页多大
data.itemsOffset	int32	当前条目偏移，从 0 开始
data.itemLength	int32	数据列表长度
data.items	object[]	有效数据列表
data.items.name	string	名字，命名方式用接口名作前缀 例：eth1
data.items.uuid	string	唯一标识id
data.items.user	string	用户名账号 最小长度:0 最大长度:95
data.items.password	string	密码，明文密码长度最大为95字符 最小长度:0 最大长度:512
data.items.lcpEchoTime out	uint32	探测超时/握手时间 最小值:5 最大值:60
data.items.idleTimeout	uint32	空闲超时时间/超时时间 最小值:20 最大值:240
data.items.lcpEchoRetr y	uint32	重试次数 最小值:1 最大值:10
data.items.autoConnect	bool	自动重连/自动拨号
data.items.setRoute	bool	添加默认路由

data.items.setDns	bool	作为系统首选DNS
-------------------	------	-----------

请求成功样例:

```
{
  "data": {
    "itemsOffset": 0,
    "items": [
      {
        "setDns": true,
        "autoConnect": true,
        "lcpEchoRetry": 3,
        "uuid": "00000000000000000000000001100000000",
        "password": "*****",
        "lcpEchoTimeout": 20,
        "idleTimeout": 80,
        "setRoute": true,
        "name": "eth0",
        "user": ""
      },
      {
        "setDns": true,
        "autoConnect": true,
        "lcpEchoRetry": 3,
        "uuid": "000000000000000000000000011000000001",
        "password": "*****",
        "lcpEchoTimeout": 20,
        "idleTimeout": 80,
        "setRoute": true,
        "name": "eth1",
        "user": ""
      },
      {
        "setDns": true,
        "autoConnect": true,
        "lcpEchoRetry": 3,
        "uuid": "000000000000000000000000011000000002",
        "password": "*****",
        "lcpEchoTimeout": 20,
        "idleTimeout": 80,
        "setRoute": true,
        "name": "eth2",
        "user": ""
      },
      {
        "setDns": true,
        "autoConnect": true,
        "lcpEchoRetry": 3,
        "uuid": "000000000000000000000000011000000003",
        "password": "*****",
        "lcpEchoTimeout": 20,
        "idleTimeout": 80,
        "setRoute": true,
        "name": "eth3",
        "user": ""
      }
    ],
    "totalItems": 4,
    "pageNumber": 1,
    "itemLength": 4,
    "totalPages": 1,
    "pageSize": 100
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3.3.2 获取单个PPPOE接口配置信息

功能说明：获取单个PPPOE接口配置信息

1) API请求数据

GET `https://{host}/api/v1/namespaces/@namespace/pppoe/@name`

2) 输入参数

- URL 参数

参数	类型	是否必选	描述
_select	string	否	选择字段, 字段用逗号分隔 最大长度:1000

请求样例：

获取单个pppoe的配置信息，查询条件：查看名为eth1的pppoe

GET `https://192.168.1.1/api/v1/namespaces/public/pppoe/eth1`

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	pppoe配置
data.name	string	名字，命名方式用接口名作前缀 例：eth1
data.uuid	string	唯一标识id
data.user	string	用户名账号 最小长度:0 最大长度:95
data.password	string	密码，明文密码长度最大为95字符 最小长度:0 最大长度:512
data.lcpEchoTimeout	uint32	探测超时/握手时间 最小值:5 最大值:60
data.idleTimeout	uint32	空闲超时时间/超时时间 最小值:20 最大值:240
data.lcpEchoRetry	uint32	重试次数 最小值:1 最大值:10
data.autoConnect	bool	自动重连/自动拨号
data.setRoute	bool	添加默认路由
data.setDns	bool	作为系统首选DNS

请求成功样例：


```
{
  "data": {
    "setDns": true,
    "autoConnect": true,
    "lcpEchoRetry": 3,
    "uuid": "00000000000000000000000000000001",
    "password": "*****",
    "lcpEchoTimeout": 20,
    "idleTimeout": 80,
    "setRoute": true,
    "name": "eth1",
    "user": ""
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3.3.3 添加PPPOE接口配置信息

功能说明: 添加PPPOE接口配置信息

1) API请求数据

POST <https://{host}/api/v1/namespaces/@namespace/pppoe>

2) 输入参数

- Body参数

参数	类型	是否必选	描述
obj	object	是	pppoe配置
obj.name	string	是	名字, 命名方式用接口名作前缀 例: eth1
obj.user	string	否	用户名账号 最小长度:0 最大长度:95
obj.password	string	否	密码, 明文密码长度最大为95字符 最小长度:0 最大长度:512
obj.lcpEchoTimeout	uint32	否	探测超时/握手时间 最小值:5 最大值:60 默认值:true
obj.idleTimeout	uint32	否	空闲超时时间/超时时间 最小值:20 最大值:240 默认值:true
obj.lcpEchoRetry	uint32	否	重试次数 最小值:1 最大值:10 默认值:true
obj.autoConnect	bool	否	自动重连/自动拨号 默认值:true

obj.setRoute	bool	否	添加默认路由 默认值:true
obj.setDns	bool	否	作为系统首选DNS 默认值:true

请求样例:

新增单个pppoe配置信息

POST https://192.168.1.1/api/v1/namespaces/public/pppoe

```
{
  "user": "test1234",
  "name": "eth1.1",
  "password": "test1234"
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	pppoe配置
data.name	string	名字, 命名方式用接口名作前缀 例: eth1
data.uuid	string	唯一标识id
data.user	string	用户名账号 最小长度:0 最大长度:95
data.password	string	密码, 明文密码长度最大为95字符 最小长度:0 最大长度:512
data.lcpEchoTimeout	uint32	探测超时/握手时间 最小值:5 最大值:60
data.idleTimeout	uint32	空闲超时时间/超时时间 最小值:20 最大值:240
data.lcpEchoRetry	uint32	重试次数 最小值:1 最大值:10
data.autoConnect	bool	自动重连/自动拨号
data.setRoute	bool	添加默认路由
data.setDns	bool	作为系统首选DNS

请求成功样例:

```
{
  "data": {
    "setDns": true,
    "autoConnect": true,
    "lcpEchoRetry": 3,
    "uuid": "7657DACC18A8409494F82FD5C92E5FD6",
    "password": "+-{}~!4333231347375647",
    "lcpEchoTimeout": 20,
```

```
{
  "idleTimeout": 80,
  "setRoute": true,
  "name": "eth1.1",
  "user": "test1234"
},
{
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3.3.4 删除PPPOE单个接口配置信息

功能说明: 删除PPPOE单个接口配置信息

1) API请求数据

DELETE https://{host}/api/v1/namespaces/@namespace/pppoe/@name

2) 输入参数

请求样例:

删除一个pppoe, 删除名为eth1.1的pppoe

DELETE https://192.168.1.1/api/v1/namespaces/public/pppoe/eth1.1

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	pppoe配置
data.name	string	名字, 命名方式用接口名作前缀 例: eth1
data.uuid	string	唯一标识id
data.user	string	用户名账号 最小长度:0 最大长度:95
data.password	string	密码, 明文密码长度最大为95字符 最小长度:0 最大长度:512
data.lcpEchoTimeout	uint32	探测超时/握手时间 最小值:5 最大值:60
data.idleTimeout	uint32	空闲超时时间/超时时间 最小值:20 最大值:240
data.lcpEchoRetry	uint32	重试次数 最小值:1 最大值:10

data.autoConnect	bool	自动重连/自动拨号
data.setRoute	bool	添加默认路由
data.setDns	bool	作为系统首选DNS

请求成功样例：

```
{
  "data": {
    "setDns": true,
    "autoConnect": true,
    "lcpEchoRetry": 5,
    "uuid": "7657DACC18A8409494F82FD5C92E5FD6",
    "password": "+-[]~!4627F67737371607475707",
    "lcpEchoTimeout": 25,
    "idleTimeout": 80,
    "setRoute": true,
    "name": "eth1.1",
    "user": "putuser"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3.3.5 批量删除pppoe接口配置

功能说明：批量删除pppoe接口配置

1) API请求数据

POST

https://{host}/api/batch/v1/namespaces/@namespace/pppoe?_method=delete

2) 输入参数

• Body参数

参数	类型	是否必选	描述
obj	object[]	是	PPPOE配置列表
obj.name	string	是	名字，命名方式用接口名作前缀 例：eth1
obj.user	string	否	用户名账号 最小长度:0 最大长度:95
obj.password	string	否	密码，明文密码长度最大为95字符 最小长度:0 最大长度:512
obj.lcpEchoTimeout	uint32	否	探测超时/握手时间 最小值:5 最大值:60 默认值:true

obj.idleTimeout	uint32	否	空闲超时时间/超时时间 最小值:20 最大值:240 默认值:true
obj.lcpEchoRetry	uint32	否	重试次数 最小值:1 最大值:10 默认值:true
obj.autoConnect	bool	否	自动重连/自动拨号 默认值:true
obj.setRoute	bool	否	添加默认路由 默认值:true
obj.setDns	bool	否	作为系统首选DNS 默认值:true

请求样例:

```
批量删除pppoe数据
POST https://192.168.1.1/api/batch/v1/namespaces/public/pppoe?_method=DELETE
[
  {
    "name": "eth1.2"
  },
  {
    "name": "eth1.3"
  }
]
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	PPPOE配置列表
data.name	string	名字，命名方式用接口名作前缀 例：eth1
data.uuid	string	唯一标识id
data.user	string	用户名账号 最小长度:0 最大长度:95
data.password	string	密码，明文密码长度最大为95字符 最小长度:0 最大长度:512
data.lcpEchoTimeout	uint32	探测超时/握手时间 最小值:5 最大值:60
data.idleTimeout	uint32	空闲超时时间/超时时间 最小值:20 最大值:240

data.lcpEchoRetry	uint32	重试次数 最小值:1 最大值:10
data.autoConnect	bool	自动重连/自动拨号
data.setRoute	bool	添加默认路由
data.setDns	bool	作为系统首选DNS

请求成功样例:

```
{
  "data": [
    {
      "setDns": true,
      "autoConnect": true,
      "lcpEchoRetry": 3,
      "uuid": "2672F52DD93D4D00B1D5758D15DD3CDC",
      "password": "+-{}~!4333231347375647",
      "lcpEchoTimeout": 20,
      "idleTimeout": 80,
      "setRoute": true,
      "name": "eth1.2",
      "user": "test1234"
    },
    {
      "setDns": true,
      "autoConnect": true,
      "lcpEchoRetry": 3,
      "uuid": "6D5C2BE41A8645DFABD8CEE51AC5C850",
      "password": "+-{}~!4333231347375647",
      "lcpEchoTimeout": 20,
      "idleTimeout": 80,
      "setRoute": true,
      "name": "eth1.3",
      "user": "test1234"
    }
  ],
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3.3.6 增量修改PPPOE单个接口配置信息

功能说明: 增量修改PPPOE单个接口配置信息

1) API请求数据

PATCH <https://{host}/api/v1/namespaces/@namespace/pppoe/@name>

2) 输入参数

- Body参数

参数	类型	是否必选	描述
obj	object	是	pppoe配置
obj.name	string	是	名字, 命名方式用接口名作前缀 例: eth1

obj.user	string	否	用户名账号 最小长度:0 最大长度:95
obj.password	string	否	密码, 明文密码长度最大为95字符 最小长度:0 最大长度:512
obj.lcpEchoTimeout	uint32	否	探测超时/握手时间 最小值:5 最大值:60 默认值:true
obj.idleTimeout	uint32	否	空闲超时时间/超时时间 最小值:20 最大值:240 默认值:true
obj.lcpEchoRetry	uint32	否	重试次数 最小值:1 最大值:10 默认值:true
obj.autoConnect	bool	否	自动重连/自动拨号 默认值:true
obj.setRoute	bool	否	添加默认路由 默认值:true
obj.setDns	bool	否	作为系统首选DNS 默认值:true

请求样例:

增量修改单个pppoe, 修改重试次数为5

```
PATCH https://192.168.1.1/api/v1/namespaces/public/pppoe/eth1.1
{
  "lcpEchoRetry": 5
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	pppoe配置
data.name	string	名字, 命名方式用接口名作前缀 例: eth1
data.uuid	string	唯一标识id
data.user	string	用户名账号 最小长度:0 最大长度:95
data.password	string	密码, 明文密码长度最大为95字符 最小长度:0 最大长度:512

data.lcpEchoTimeout	uint32	探测超时/握手时间 最小值:5 最大值:60
data.idleTimeout	uint32	空闲超时时间/超时时间 最小值:20 最大值:240
data.lcpEchoRetry	uint32	重试次数 最小值:1 最大值:10
data.autoConnect	bool	自动重连/自动拨号
data.setRoute	bool	添加默认路由
data.setDns	bool	作为系统首选DNS

请求成功样例:

```
{
  "data": {
    "setDns": true,
    "autoConnect": true,
    "lcpEchoRetry": 5,
    "uuid": "7657DACC18A8409494F82FD5C92E5FD6",
    "password": "+-{}~!4627F67737371607475707",
    "lcpEchoTimeout": 25,
    "idleTimeout": 80,
    "setRoute": true,
    "name": "eth1.1",
    "user": "putuser"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3.3.7 全量修改PPPOE单个接口配置信息

功能说明: 全量修改PPPOE单个接口配置信息

1) API请求数据

PUT <https://{host}/api/v1/namespaces/@namespace/pppoe/@name>

2) 输入参数

• Body参数

参数	类型	是否必选	描述
obj	object	是	pppoe配置
obj.name	string	是	名字, 命名方式用接口名作前缀 例: eth1
obj.user	string	否	用户名账号 最小长度:0 最大长度:95

obj.password	string	否	密码，明文密码长度最大为95字符 最小长度:0 最大长度:512
obj.lcpEchoTimeout	uint32	否	探测超时/握手时间 最小值:5 最大值:60 默认值:true
obj.idleTimeout	uint32	否	空闲超时时间/超时时间 最小值:20 最大值:240 默认值:true
obj.lcpEchoRetry	uint32	否	重试次数 最小值:1 最大值:10 默认值:true
obj.autoConnect	bool	否	自动重连/自动拨号 默认值:true
obj.setRoute	bool	否	添加默认路由 默认值:true
obj.setDns	bool	否	作为系统首选DNS 默认值:true

请求样例:

全量更新一个pppoe配置，修改账号为putuser,修改密码为putpassword，握手时间为25，其余为默认值

```
PUT https://192.168.1.1/api/v1/namespaces/public/pppoe/eth1.1
{
  "user": "putuser",
  "lcpEchoTimeout": 25,
  "name": "eth1.1",
  "password": "putpassword"
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	pppoe配置
data.name	string	名字，命名方式用接口名作前缀 例：eth1
data.uuid	string	唯一标识id
data.user	string	用户名账号 最小长度:0 最大长度:95
data.password	string	密码，明文密码长度最大为95字符 最小长度:0 最大长度:512
data.lcpEchoTimeout	uint32	探测超时/握手时间 最小值:5 最大值:60

data.idleTimeout	uint32	空闲超时时间/超时时间 最小值:20 最大值:240
data.lcpEchoRetry	uint32	重试次数 最小值:1 最大值:10
data.autoConnect	bool	自动重连/自动拨号
data.setRoute	bool	添加默认路由
data.setDns	bool	作为系统首选DNS

请求成功样例:

```
{
  "data": {
    "setDns": true,
    "autoConnect": true,
    "lcpEchoRetry": 3,
    "uuid": "7657DACC18A8409494F82FD5C92E5FD6",
    "password": "+-{}~!4627F67737371607475707",
    "lcpEchoTimeout": 25,
    "idleTimeout": 80,
    "setRoute": true,
    "name": "eth1.1",
    "user": "putuser"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3.3.8 获取PPPOE接口状态信息

功能说明: 获取PPPOE接口状态信息

1) API请求数据

GET https://{host}/api/v1/namespaces/@namespace/pppoestatus/@name

2) 输入参数

- URL参数

参数	类型	是否必选	描述
_search	string	否	模糊搜索关键字 最大长度:95

请求样例:

GET https://192.168.1.1/api/v1/namespaces/public/pppoestatus/admin

3) 输出参数

- API返回数据

参数	类型	描述
----	----	----

code	int	错误码
message	string	错误信息
data	object	pppoe接口状态信息
data.name	string	名字，命名方式用接口名作前缀 例：eth1 最小长度:1 最大长度:95
data.status	string	pppoe接口拨号状态 最大长度:32
data.status_code	uint32	状态码，与status对应，主要是给前端显示时判断用
data.sendBytes	uint32	pppoe接口自拨号成功发送字节数(单位KB)
data.recvBytes	uint32	pppoe接口自拨号成功接收字节数(单位KB)
data.durationTime	string	pppoe接口自拨号成功持续时间 最大长度:32
data.logInfo	string	pppoe接口拨号日志对应文件

请求成功样例：

```
{
  "message": "",
  "data": {
    "recvBytes": 0,
    "name": "admin",
    "status_code": 0,
    "status": "test",
    "durationTime": "test",
    "sendBytes": 0,
    "logInfo": "test"
  },
  "code": 0
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3.3.9 PPPOE接口连接操作

功能说明：PPPOE接口连接操作

1) API请求数据

POST

https://{host}/api/v1/namespaces/@namespace/pppoeconnect/@name

2) 输入参数

- Body参数

参数	类型	是否必选	描述
obj	object	是	pppoe接口连接信息
obj.name	string	否	名字，命名方式用接口名作前缀 例：eth1 最大长度:32

obj.oprtype	string	否	pppoe接口连接操作类型 默认值:true 枚举值:[LINKON:拨号连接 LINKOFF:拨号断开连接 ONSTATUS:拨号连接状态]
obj.status_code	uint32	否	状态码

请求样例:

```
POST https://192.168.1.1/api/v1/namespaces/public/pppoeconnect/admin
{
  "oprtype": "ONSTATUS",
  "name": "test",
  "status_code": 0
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	pppoe接口连接信息
data.name	string	名字, 命名方式用接口名作前缀 例: eth1 最大长度:32
data.oprtype	string	pppoe接口连接操作类型 枚举值:[LINKON:拨号连接 LINKOFF:拨号断开连接 ONSTATUS:拨号连接状态]
data.status_code	uint32	状态码

请求成功样例:

```
{
  "message": "",
  "data": {
    "oprtype": "ONSTATUS",
    "name": "test",
    "status_code": 0
  },
  "code": 0
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3.4 虚拟网线组

3.4.1 获取全部虚拟网线组配置信息

功能说明：获取全部虚拟网线组配置信息

1) API请求数据

GET

https://{host}/api/v1/namespaces/@namespace/virlines

2) 输入参数

- URL参数

参数	类型	是否必选	描述
itf	string	否	过滤参数：虚拟网线组的一个接口名称

请求样例：

获取多条虚拟网线配置信息，查询条件：起始位置为0，最多展示100条

GET https://192.168.1.1/api/v1/namespaces/public/virlines?_start=0&_length=100

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	分页数据
data.totalItems	int32	总共多少项目
data.totalPages	int32	总共多少页
data.pageNumber	int32	当前页码，从 1 开始
data.pageSize	int32	每页多大
data.itemsOffset	int32	当前条目偏移, 从 0 开始
data.itemLength	int32	数据列表长度
data.items	object[]	有效数据列表
data.items.name	string	虚拟网线的名字 最小长度:1 最大长度:95
data.items.uuid	string	唯一标识id
data.items.line1	string	虚拟网线组的第一个网口
data.items.line2	string	虚拟网线组的第二个网口
data.items.description	string	虚拟网线组的描述 最小长度:0 最大长度:95

请求成功样例：

```
{  "data": {
```

```
    "itemsOffset": 0,
    "items": [
      {
        "uuid": "95095CC739E74E6294D4B080C8E9FE19",
        "line1": "eth1",
        "name": "eth1-eth2",
        "line2": "eth2"
      }
    ],
    "totalItems": 1,
    "pageNumber": 1,
    "itemLength": 1,
    "totalPages": 1,
    "pageSize": 100
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3.4.2 获取单个虚拟网线组配置信息

功能说明：获取单个虚拟网线组配置信息

1) API请求数据

GET

https://{host}/api/v1/namespaces/@namespace/virlines/@name

2) 输入参数

请求样例：

获取单条虚拟网线，查询条件：查看名为eth1-eth2的虚拟网线

GET https://192.168.1.1/api/v1/namespaces/public/virlines/eth1-eth2

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	虚拟网线组配置列表
data.name	string	虚拟网线的名字 最小长度:1 最大长度:95
data.uuid	string	唯一标识id
data.line1	string	虚拟网线组的第一个网口
data.line2	string	虚拟网线组的第二个网口
data.description	string	虚拟网线组的描述 最小长度:0 最大长度:95

请求成功样例：

```
{
  "data": {
    "uuid": "95095CC739E74E6294D4B080C8E9FE19",
    "line1": "eth1",
    "name": "eth1-eth2",
    "line2": "eth2"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3.4.3 添加虚拟网线组配置信息

功能说明: 添加虚拟网线组配置信息

1) API请求数据

POST <https://{host}/api/v1/namespaces/@namespace/virlines>

2) 输入参数

• Body参数

参数	类型	是否必选	描述
obj	object	是	虚拟网线组配置列表
obj.name	string	是	虚拟网线的名字 最小长度:1 最大长度:95
obj.line1	string	否	虚拟网线组的第一个网口
obj.line2	string	否	虚拟网线组的第二个网口
obj.description	string	否	虚拟网线组的描述 最小长度:0 最大长度:95

请求样例:

新增单个pppoe配置信息

POST <https://192.168.1.1/api/v1/namespaces/public/virlines>

```
{
  "line1": "eth1",
  "name": "eth1-eth2",
  "line2": "eth2"
}
```

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	虚拟网线组配置列表

data.name	string	虚拟网线的名字 最小长度:1 最大长度:95
data.uuid	string	唯一标识id
data.line1	string	虚拟网线组的第一个网口
data.line2	string	虚拟网线组的第二个网口
data.description	string	虚拟网线组的描述 最小长度:0 最大长度:95

请求成功样例:

```
{
  "data": {
    "uuid": "95095CC739E74E6294D4B080C8E9FE19",
    "line1": "eth1",
    "name": "eth1-eth2",
    "line2": "eth2"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3.4.4 删除单个虚拟网线组配置信息

功能说明: 删除单个虚拟网线组配置信息

1) API请求数据

DELETE https://{host}/api/v1/namespaces/@namespace/virlines/@name

2) 输入参数

请求样例:

删除一个虚拟网线, 删除名为eth1-eth2的虚拟网线

DELETE https://192.168.1.1/api/v1/namespaces/public/virlines/eth1-eth2

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	虚拟网线组配置列表
data.name	string	虚拟网线的名字 最小长度:1 最大长度:95
data.uuid	string	唯一标识id

data.line1	string	虚拟网线组的第一个网口
data.line2	string	虚拟网线组的第二个网口
data.description	string	虚拟网线组的描述 最小长度:0 最大长度:95

请求成功样例:

```
{
  "data": {
    "description": "patchtest",
    "name": "eth1-eth2",
    "line1": "eth1",
    "uuid": "95095CC739E74E6294D4B080C8E9FE19",
    "line2": "eth2"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3.4.5 批量删除虚拟网线组

功能说明: 批量删除虚拟网线组

1) API请求数据

POST

https://{host}/api/batch/v1/namespaces/@namespace/virlines?_method=delete

2) 输入参数

• Body参数

参数	类型	是否必选	描述
obj	object[]	是	虚拟网线组配置列表
obj.name	string	是	虚拟网线的名字 最小长度:1 最大长度:95
obj.line1	string	否	虚拟网线组的第一个网口
obj.line2	string	否	虚拟网线组的第二个网口
obj.description	string	否	虚拟网线组的描述 最小长度:0 最大长度:95

请求样例:

批量删除虚拟网线数据

POST https://192.168.1.1/api/batch/v1/namespaces/public/virlines?_method=DELETE

```
[
  {
    "name": "eth1-eth2"
  },
  {

```

```
      "name": "eth3-eth4"
    }
  ]
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	虚拟网线组配置列表
data.name	string	虚拟网线的名字 最小长度:1 最大长度:95
data.uuid	string	唯一标识id
data.line1	string	虚拟网线组的第一个网口
data.line2	string	虚拟网线组的第二个网口
data.description	string	虚拟网线组的描述 最小长度:0 最大长度:95

请求成功样例：

```
{
  "data": [
    {
      "uuid": "2672F52DD93D4D00B1D5758D15DD3CDC",
      "line1": "eth1",
      "name": "eth1-eth2",
      "line2": "eth2"
    },
    {
      "uuid": "6D5C2BE41A8645DFABD8CEE51AC5C850",
      "line1": "eth3",
      "name": "eth3-eth4",
      "line2": "eth4"
    }
  ],
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3.4.6 增量修改单个虚拟网线组配置信息

功能说明：增量修改单个虚拟网线组配置信息

1) API请求数据

PATCH

https://{host}/api/v1/namespaces/@namespace/virlines/@name

2) 输入参数

- Body参数

参数	类型	是否必选	描述
obj	object	是	虚拟网线组配置列表
obj.name	string	是	虚拟网线的名字 最小长度:1 最大长度:95
obj.line1	string	否	虚拟网线组的第一个网口
obj.line2	string	否	虚拟网线组的第二个网口
obj.description	string	否	虚拟网线组的描述 最小长度:0 最大长度:95

请求样例:

增量修改单个虚拟网线, 修改描述为patchtest

```
PATCH https://192.168.1.1/api/v1/namespaces/public/virlines/eth1-eth2
{
  "description": "patchtest"
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	虚拟网线组配置列表
data.name	string	虚拟网线的名字 最小长度:1 最大长度:95
data.uuid	string	唯一标识id
data.line1	string	虚拟网线组的第一个网口
data.line2	string	虚拟网线组的第二个网口
data.description	string	虚拟网线组的描述 最小长度:0 最大长度:95

请求成功样例:

```
{
  "data": {
    "description": "patchtest",
    "name": "eth1-eth2",
    "line1": "eth1",
    "uuid": "95095CC739E74E6294D4B080C8E9FE19",
    "line2": "eth2"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3.4.7 全量修改单个虚拟网线组配置信息

功能说明：全量修改单个虚拟网线组配置信息

1) API请求数据

PUT https://{host}/api/v1/namespaces/@namespace/virlines/@name

2) 输入参数

- Body参数

参数	类型	是否必选	描述
obj	object	是	虚拟网线组配置列表
obj.name	string	是	虚拟网线的名字 最小长度:1 最大长度:95
obj.line1	string	否	虚拟网线组的第一个网口
obj.line2	string	否	虚拟网线组的第二个网口
obj.description	string	否	虚拟网线组的描述 最小长度:0 最大长度:95

请求样例：

全量更新一个虚拟网线配置，修改描述为puttest,其余不变

```
PUT https://192.168.1.1/api/v1/namespaces/public/virlines/eth1-eth2
{
  "description": "puttest",
  "line1": "eth1",
  "name": "eth1-eth2",
  "line2": "eth2"
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	虚拟网线组配置列表
data.name	string	虚拟网线的名字 最小长度:1 最大长度:95
data.uuid	string	唯一标识id
data.line1	string	虚拟网线组的第一个网口
data.line2	string	虚拟网线组的第二个网口

data.description	string	虚拟网线组的描述 最小长度:0 最大长度:95
------------------	--------	-------------------------------

请求成功样例:

```
{
  "data": {
    "description": "puttest",
    "name": "eth1-eth2",
    "line1": "eth1",
    "uuid": "95095CC739E74E6294D4B080C8E9FE19",
    "line2": "eth2"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3.5 路由

3.5.1 静态路由IPV4

3.5.1.1 获取单条IPv4静态路由信息

功能说明: 通过此接口, 可以获取指定单条静态路由表项

1) API请求数据

GET

https://{host}/api/v1/namespaces/@namespace/staticroutes/ipv4/@uuid

2) 输入参数

请求样例:

获取单条ipv4静态路由的配置

```
GET https://192.168.1.1/api/v1/namespaces/public/staticroutes/ipv4/4770AB8D9C724277ADA046371463C13F
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	IPv4静态路由表项
data.uuid	string	静态路由唯一资源ID
data.enable	bool	静态路由启禁用开关
data.description	string	描述信息 最小长度:0 最大长度:95

data.prefix	string	目的ip地址/掩码
data.gateway	string	下一跳ip地址
data.ifname	string	出接口名称 最小长度:1 最大长度:48
data.metric	uint32	路由度量值 最小值:0 最大值:255
data.linkdtEnable	bool	静态路由关联接口链路探测启禁用开关，默认false，不可修改
data.linkdt	string	静态路由关联链路探测，当前版本不可修改
data.tag	uint32	zebra路由标记，默认为0，当前版本不可修改
data.distance	uint8	路由管理距离，默认为1，当前版本不可修改 最小值:1 最大值:255
data.tableId	uint32	路由表Id，默认为0，当前版本不可修改 最小值:0 最大值:256
data.weight	uint8	IPv4静态路由权重，默认为1，当前版本不可修改 最小值:1 最大值:255
data.status	uint8	状态信息，状态字段不可设置或修改

请求成功样例：

```
{
  "data": {
    "enable": true,
    "metric": 0,
    "tag": 0,
    "uuid": "4770AB8D9C724277ADA046371463C13F",
    "ifname": "auto",
    "tableId": 0,
    "weight": 1,
    "linkdtEnable": false,
    "gateway": "10.155.255.254",
    "distance": 1,
    "prefix": "0.0.0.0/0",
    "description": ""
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3.5.1.2 获取IPv4静态路由列表

功能说明：通过此接口，可以获取配置中心所有静态路由表项

1) API请求数据

GET `https://{host}/api/v1/namespaces/@namespace/staticroutes/ipv4`

2) 输入参数

- URL 参数

参数	类型	是否必选	描述
_start	int32	否	起始位置, 从 0 开始 最小值:0 最大值:2000000
metric	uint32	否	静态路由度量值 最小值:0 最大值:255
prefix	string	否	目的ip地址/掩码
ifname	string	否	出接口名称 最小长度:1 最大长度:48
_length	int32	否	获取最大长度, 限制最大值, 如果数据不够可能返回小于 length 最小值:1 最大值:200 默认值:true
_search	string	否	模糊搜索关键字 最大长度:95
_select	string	否	选择字段, 字段用逗号分隔 最大长度:1000
gateway	string	否	下一跳ip地址
_sortby	string	否	指定排序字段 最大长度:100
_order	string	否	指定排序方式 (正序/倒序) 默认值:true 枚举值:[asc:按升序排列 desc:按降序排列]

请求样例:

获取所有ipv4静态路由配置, 查询条件: 起始位置为0, 最多展示100条

GET `https://192.168.1.1/api/v1/namespaces/public/staticroutes/ipv4?_start=0&_length=100`

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	分页数据

data.totalItems	int32	总共多少项目
data.totalPages	int32	总共多少页
data.pageNumber	int32	当前页码, 从 1 开始
data.pageSize	int32	每页多大
data.itemsOffset	int32	当前条目偏移, 从 0 开始
data.itemLength	int32	数据列表长度
data.items	object[]	有效数据列表
data.items.uuid	string	静态路由唯一资源ID
data.items.enable	bool	静态路由启禁用开关
data.items.description	string	描述信息 最小长度:0 最大长度:95
data.items.prefix	string	目的ip地址/掩码
data.items.gateway	string	下一跳ip地址
data.items.ifname	string	出接口名称 最小长度:1 最大长度:48
data.items.metric	uint32	路由度量值 最小值:0 最大值:255
data.items.linkdtEnable	bool	静态路由关联接口链路探测启禁用开关, 默认false, 不可修改
data.items.linkdt	string	静态路由关联链路探测, 当前版本不可修改
data.items.tag	uint32	zebra路由标记, 默认为0, 当前版本不可修改
data.items.distance	uint8	路由管理距离, 默认为1, 当前版本不可修改 最小值:1 最大值:255
data.items.tableId	uint32	路由表Id, 默认为0, 当前版本不可修改 最小值:0 最大值:256
data.items.weight	uint8	IPv4静态路由权重, 默认为1, 当前版本不可修改 最小值:1 最大值:255
data.items.status	uint8	状态信息, 状态字段不可设置或修改

请求成功样例:

```
{
  "data": {
    "itemsOffset": 0,
    "items": [
      {
        "enable": true,
        "metric": 0,
        "tag": 0,
        "status": 1,
        "uuid": "4770AB8D9C724277ADA046371463C13F",
        "ifname": "eth0",
        "tableId": 0,
        "weight": 1,
        "linkdtEnable": false,

```



```
        "gateway": "10.155.255.254",
        "distance": 1,
        "prefix": "0.0.0.0/0",
        "description": ""
      },
      "totalItems": 1,
      "pageNumber": 1,
      "itemLength": 1,
      "totalPages": 1,
      "pageSize": 100
    },
    "code": 0,
    "message": "成功"
  }
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3.5.1.3 添加一条IPv4静态路由信息

功能说明：通过此接口，可以添加一条静态路由表项

1) API请求数据

POST

https://{host}/api/v1/namespaces/@namespace/staticroutes/ipv4

2) 输入参数

• Body参数

参数	类型	是否必选	描述
obj	object	是	IPv4静态路由表项
obj.enable	bool	否	静态路由启禁用开关 默认值:true
obj.description	string	否	描述信息 最小长度:0 最大长度:95
obj.prefix	string	是	目的ip地址/掩码
obj.gateway	string	是	下一跳ip地址
obj.ifname	string	否	出接口名称 最小长度:1 最大长度:48 默认值:true
obj.metric	uint32	否	路由度量值 最小值:0 最大值:255
obj.linkdtEnable	bool	否	静态路由关联接口链路探测启禁用开关，默 认false，不可修改
obj.linkdt	string	否	静态路由关联链路探测，当前版本不可修改
obj.tag	uint32	否	zebra路由标记，默认为0，当前版本不可 修改

obj.distance	uint8	否	路由管理距离，默认为1，当前版本不可修改 最小值:1 最大值:255 默认值:true
obj.tableId	uint32	否	路由表Id，默认为0，当前版本不可修改 最小值:0 最大值:256
obj.weight	uint8	否	IPv4静态路由权重，默认为1，当前版本不可修改 最小值:1 最大值:255 默认值:true

请求样例:

添加一条ipv4静态路由

```
POST https://192.168.1.1/api/v1/namespaces/public/staticroutes/ipv4
{
  "enable": true,
  "metric": 0,
  "linkdtEnable": false,
  "ifname": "auto",
  "gateway": "10.155.255.254",
  "prefix": "0.0.0.0/0",
  "description": "default staticroute"
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	IPv4静态路由表项
data.uuid	string	静态路由唯一资源ID
data.enable	bool	静态路由启禁用开关
data.description	string	描述信息 最小长度:0 最大长度:95
data.prefix	string	目的ip地址/掩码
data.gateway	string	下一跳ip地址
data.ifname	string	出接口名称 最小长度:1 最大长度:48
data.metric	uint32	路由度量值 最小值:0 最大值:255
data.linkdtEnable	bool	静态路由关联接口链路探测启禁用开关，默认false，不可修改
data.linkdt	string	静态路由关联链路探测，当前版本不可修改

data.tag	uint32	zebra路由标记，默认为0，当前版本不可修改
data.distance	uint8	路由管理距离，默认为1，当前版本不可修改 最小值:1 最大值:255
data.tableId	uint32	路由表Id，默认为0，当前版本不可修改 最小值:0 最大值:256
data.weight	uint8	IPv4静态路由权重，默认为1，当前版本不可修改 最小值:1 最大值:255
data.status	uint8	状态信息，状态字段不可设置或修改

请求成功样例：

```
{
  "data": {
    "enable": true,
    "metric": 0,
    "tag": 0,
    "uuid": "9F3885CF7FBB499D8E30D73DD3386A86",
    "ifname": "auto",
    "tableId": 0,
    "weight": 1,
    "linkdtEnable": false,
    "gateway": "10.155.255.254",
    "distance": 1,
    "prefix": "0.0.0.0/0",
    "description": "default staticroute"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3.5.1.4 添加多条IPv4静态路由信息

功能说明：通过此接口，可以添加多条静态路由表项

1) API请求数据

POST

https://{host}/api/batch/v1/namespaces/@namespace/staticroutes/ipv4

2) 输入参数

• Body参数

参数	类型	是否必选	描述
objs	object[]	是	IPv4路由表项列表
objs.enable	bool	否	静态路由启禁用开关 默认值:true
objs.description	string	否	描述信息 最小长度:0 最大长度:95
objs.prefix	string	是	目的ip地址/掩码

objs.gateway	string	是	下一跳ip地址
objs.ifname	string	否	出接口名称 最小长度:1 最大长度:48 默认值:true
objs.metric	uint32	否	路由度量值 最小值:0 最大值:255
objs.linkdtEnable	bool	否	静态路由关联接口链路探测启禁用开关，默认false，不可修改
objs.linkdt	string	否	静态路由关联链路探测，当前版本不可修改
objs.tag	uint32	否	zebra路由标记，默认为0，当前版本不可修改
objs.distance	uint8	否	路由管理距离，默认为1，当前版本不可修改 最小值:1 最大值:255 默认值:true
objs.tableId	uint32	否	路由表Id，默认为0，当前版本不可修改 最小值:0 最大值:256
objs.weight	uint8	否	IPv4静态路由权重，默认为1，当前版本不可修改 最小值:1 最大值:255 默认值:true

请求样例:

添加多条ipv4静态路由

```
POST https://192.168.1.1/api/batch/v1/namespaces/public/staticroutes/ipv4
[
  {
    "metric": 1,
    "linkdtEnable": false,
    "gateway": "10.155.255.254",
    "prefix": "0.0.0.0/0",
    "ifname": "eth0"
  },
  {
    "metric": 2,
    "linkdtEnable": false,
    "gateway": "10.155.255.254",
    "prefix": "0.0.0.0/0",
    "ifname": "auto"
  }
]
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	IPv4路由表项列表

data.uuid	string	静态路由唯一资源ID
data.enable	bool	静态路由启禁用开关
data.description	string	描述信息 最小长度:0 最大长度:95
data.prefix	string	目的ip地址/掩码
data.gateway	string	下一跳ip地址
data.ifname	string	出接口名称 最小长度:1 最大长度:48
data.metric	uint32	路由度量值 最小值:0 最大值:255
data.linkdtEnable	bool	静态路由关联接口链路探测启禁用开关，默认false，不可修改
data.linkdt	string	静态路由关联链路探测，当前版本不可修改
data.tag	uint32	zebra路由标记，默认为0，当前版本不可修改
data.distance	uint8	路由管理距离，默认为1，当前版本不可修改 最小值:1 最大值:255
data.tableId	uint32	路由表Id，默认为0，当前版本不可修改 最小值:0 最大值:256
data.weight	uint8	IPv4静态路由权重，默认为1，当前版本不可修改 最小值:1 最大值:255
data.status	uint8	状态信息，状态字段不可设置或修改

请求成功样例：

```
{
  "data": [
    {
      "enable": true,
      "metric": 1,
      "tag": 0,
      "uuid": "6A3FBCFDCD644ADFB90F9E087C22172C",
      "ifname": "eth0",
      "tableId": 0,
      "linkdtEnable": false,
      "weight": 1,
      "distance": 1,
      "prefix": "0.0.0.0/0",
      "gateway": "10.155.255.254"
    },
    {
      "enable": true,
      "metric": 2,
      "tag": 0,
      "uuid": "48AAB0EE6F6944C88A3F1A5D97102D4F",
      "ifname": "auto",
      "tableId": 0,
      "linkdtEnable": false,
      "weight": 1,
      "distance": 1,
      "prefix": "0.0.0.0/0",
      "gateway": "10.155.255.254"
    }
  ]
}
```

```
    ],
    "code": 0,
    "message": "成功"
  }
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3.5.1.5 删除指定的IPv4静态路由信息

功能说明: 通过此接口, 可以删除一条静态路由表项

1) API请求数据

DELETE https://{host}/api/v1/namespaces/@namespace/staticroutes/ipv4/@uuid

2) 输入参数

请求样例:

删除UUID为48AAB0EE6F6944C88A3F1A5D97102D4F的ipv4静态路由

DELETE https://192.168.1.1/api/v1/namespaces/public/staticroutes/ipv4/48AAB0EE6F6944C88A3F1A5D97102D4F

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	IPv4静态路由表项
data.uuid	string	静态路由唯一资源ID
data.enable	bool	静态路由启禁用开关
data.description	string	描述信息 最小长度:0 最大长度:95
data.prefix	string	目的ip地址/掩码
data.gateway	string	下一跳ip地址
data.ifname	string	出接口名称 最小长度:1 最大长度:48
data.metric	uint32	路由度量值 最小值:0 最大值:255
data.linkdtEnable	bool	静态路由关联接口链路探测启禁用开关, 默认false, 不可修改
data.linkdt	string	静态路由关联链路探测, 当前版本不可修改
data.tag	uint32	zebra路由标记, 默认为0, 当前版本不可修改

data.distance	uint8	路由管理距离，默认为1，当前版本不可修改 最小值:1 最大值:255
data.tableId	uint32	路由表Id，默认为0，当前版本不可修改 最小值:0 最大值:256
data.weight	uint8	IPv4静态路由权重，默认为1，当前版本不可修改 最小值:1 最大值:255
data.status	uint8	状态信息，状态字段不可设置或修改

请求成功样例：

```
{
  "data": {
    "enable": true,
    "metric": 2,
    "tag": 0,
    "uuid": "48AAB0EE6F6944C88A3F1A5D97102D4F",
    "ifname": "auto",
    "tableId": 0,
    "linkdtEnable": false,
    "weight": 1,
    "distance": 1,
    "prefix": "0.0.0.0/0",
    "gateway": "10.155.255.254"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3.5.1.6 通过参数删除指定的IPv4静态路由信息

功能说明：通过此接口，可以传入关键属性来删除一条静态路由表项

1) API请求数据

POST	https://{host}/api/v1/namespaces/@namespace/staticroutes/ipv4?_method=delete
------	--

2) 输入参数

• Body参数

参数	类型	是否必选	描述
obj	object	是	IPv4静态路由表项
obj.enable	bool	否	静态路由启禁用开关 默认值:true
obj.description	string	否	描述信息 最小长度:0 最大长度:95
obj.prefix	string	是	目的ip地址/掩码
obj.gateway	string	是	下一跳ip地址

obj.ifname	string	否	出接口名称 最小长度:1 最大长度:48 默认值:true
obj.metric	uint32	否	路由度量值 最小值:0 最大值:255
obj.linkdtEnable	bool	否	静态路由关联接口链路探测启禁用开关，默认false，不可修改
obj.linkdt	string	否	静态路由关联链路探测，当前版本不可修改
obj.tag	uint32	否	zebra路由标记，默认为0，当前版本不可修改
obj.distance	uint8	否	路由管理距离，默认为1，当前版本不可修改 最小值:1 最大值:255 默认值:true
obj.tableId	uint32	否	路由表Id，默认为0，当前版本不可修改 最小值:0 最大值:256
obj.weight	uint8	否	IPv4静态路由权重，默认为1，当前版本不可修改 最小值:1 最大值:255 默认值:true

请求样例：

通过参数删除指定的ipv4静态路由信息,删除目的地址为0.0.0.0/0，下一跳10.155.255.254，接口为eth0，度量为1的静态路由

```
POST https://192.168.1.1/api/v1/namespaces/public/staticroutes/ipv4?_method=DELETE
{
  "metric": 1,
  "gateway": "10.155.255.254",
  "prefix": "0.0.0.0/0",
  "ifname": "eth0"
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	IPv4静态路由表项
data.uuid	string	静态路由唯一资源ID
data.enable	bool	静态路由启禁用开关
data.description	string	描述信息 最小长度:0 最大长度:95
data.prefix	string	目的ip地址/掩码

data.gateway	string	下一跳ip地址
data.ifname	string	出接口名称 最小长度:1 最大长度:48
data.metric	uint32	路由度量值 最小值:0 最大值:255
data.linkdtEnable	bool	静态路由关联接口链路探测启禁用开关，默认false，不可修改
data.linkdt	string	静态路由关联链路探测，当前版本不可修改
data.tag	uint32	zebra路由标记，默认为0，当前版本不可修改
data.distance	uint8	路由管理距离，默认为1，当前版本不可修改 最小值:1 最大值:255
data.tableId	uint32	路由表Id，默认为0，当前版本不可修改 最小值:0 最大值:256
data.weight	uint8	IPv4静态路由权重，默认为1，当前版本不可修改 最小值:1 最大值:255
data.status	uint8	状态信息，状态字段不可设置或修改

请求成功样例：

```
{
  "data": {
    "enable": true,
    "metric": 1,
    "tag": 0,
    "uuid": "6A3FBCFDCD644ADFB90F9E087C22172C",
    "ifname": "eth0",
    "tableId": 0,
    "linkdtEnable": false,
    "weight": 1,
    "distance": 1,
    "prefix": "0.0.0.0/0",
    "gateway": "10.155.255.254"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3.5.1.7 删除多条IPv4静态路由信息

功能说明：通过此接口，可以删除多条静态路由表项

1) API请求数据

POST

https://{host}/api/batch/v1/namespaces/@namespace/staticroutes/ipv4?_method=delete

2) 输入参数

• Body参数

参数	类型	是否必选	描述
objs	object[]	是	IPv4路由表项列表
objs.enable	bool	否	静态路由启禁用开关 默认值:true
objs.description	string	否	描述信息 最小长度:0 最大长度:95
objs.prefix	string	是	目的ip地址/掩码
objs.gateway	string	是	下一跳ip地址
objs.ifname	string	否	出接口名称 最小长度:1 最大长度:48 默认值:true
objs.metric	uint32	否	路由度量值 最小值:0 最大值:255
objs.linkdtEnable	bool	否	静态路由关联接口链路探测启禁用开关，默认false，不可修改
objs.linkdt	string	否	静态路由关联链路探测，当前版本不可修改
objs.tag	uint32	否	zebra路由标记，默认为0，当前版本不可修改
objs.distance	uint8	否	路由管理距离，默认为1，当前版本不可修改 最小值:1 最大值:255 默认值:true
objs.tableId	uint32	否	路由表Id，默认为0，当前版本不可修改 最小值:0 最大值:256
objs.weight	uint8	否	IPv4静态路由权重，默认为1，当前版本不可修改 最小值:1 最大值:255 默认值:true

请求样例:

删除多条ipv4静态路由信息

```
POST https://192.168.1.1/api/batch/v1/namespaces/public/staticroutes/ipv4?_method=DELETE
[
  {
    "uuid": "6A3FBCFDCD644ADFB90F9E087C22172C"
  },
  {
    "uuid": "48AAB0EE6F6944C88A3F1A5D97102D4F"
  }
]
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	IPv4路由表项列表
data.uuid	string	静态路由唯一资源ID
data.enable	bool	静态路由启禁用开关
data.description	string	描述信息 最小长度:0 最大长度:95
data.prefix	string	目的ip地址/掩码
data.gateway	string	下一跳ip地址
data.ifname	string	出接口名称 最小长度:1 最大长度:48
data.metric	uint32	路由度量值 最小值:0 最大值:255
data.linkdtEnable	bool	静态路由关联接口链路探测启禁用开关，默认false，不可修改
data.linkdt	string	静态路由关联链路探测，当前版本不可修改
data.tag	uint32	zebra路由标记，默认为0，当前版本不可修改
data.distance	uint8	路由管理距离，默认为1，当前版本不可修改 最小值:1 最大值:255
data.tableId	uint32	路由表Id，默认为0，当前版本不可修改 最小值:0 最大值:256
data.weight	uint8	IPv4静态路由权重，默认为1，当前版本不可修改 最小值:1 最大值:255
data.status	uint8	状态信息，状态字段不可设置或修改

请求成功样例：

```
{
  "data": [
    {
      "enable": true,
      "metric": 1,
      "tag": 0,
      "uuid": "6A3FBCFDCD644ADFB90F9E087C22172C",
      "ifname": "eth0",
      "tableId": 0,
      "linkdtEnable": false,
      "weight": 1,
      "distance": 1,
      "prefix": "0.0.0.0/0",
      "gateway": "10.155.255.254"
    }
  ],
}
```

```
{
  "enable": true,
  "metric": 2,
  "tag": 0,
  "uuid": "48AAB0EE6F6944C88A3F1A5D97102D4F",
  "ifname": "auto",
  "tableId": 0,
  "linkdtEnable": false,
  "weight": 1,
  "distance": 1,
  "prefix": "0.0.0.0/0",
  "gateway": "10.155.255.254"
},
"code": 0,
"message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3.5.1.8 全量修改指定的IPv4静态路由信息

功能说明: 通过此接口, 可以全量修改一条静态路由表项

1) API请求数据

PUT <https://{host}/api/v1/namespaces/@namespace/staticroutes/ipv4/@uuid>

2) 输入参数

- Body参数

参数	类型	是否必选	描述
obj	object	是	IPv4静态路由表项
obj.enable	bool	否	静态路由启禁用开关 默认值:true
obj.description	string	否	描述信息 最小长度:0 最大长度:95
obj.prefix	string	是	目的ip地址/掩码
obj.gateway	string	是	下一跳ip地址
obj.ifname	string	否	出接口名称 最小长度:1 最大长度:48 默认值:true
obj.metric	uint32	否	路由度量值 最小值:0 最大值:255
obj.linkdtEnable	bool	否	静态路由由关联接口链路探测启禁用开关, 默认false, 不可修改
obj.linkdt	string	否	静态路由由关联链路探测, 当前版本不可修改
obj.tag	uint32	否	zebra路由标记, 默认为0, 当前版本不可修改

obj.distance	uint8	否	路由管理距离，默认为1，当前版本不可修改 最小值:1 最大值:255 默认值:true
obj.tableId	uint32	否	路由表Id，默认为0，当前版本不可修改 最小值:0 最大值:256
obj.weight	uint8	否	IPv4静态路由权重，默认为1，当前版本不可修改 最小值:1 最大值:255 默认值:true

请求样例:

全量更新UUID为6A3FBCFDCD644ADFB90F9E087C22172C的ipv4静态路由配置,修改下一跳为10.90.255.254, 目的地址0.0.0.0/0, 其他不填自动为默认值

```
PUT https://192.168.1.1/api/v1/namespaces/public/staticroutes/ipv4/6A3FBCFDCD644ADFB90F9E087C22172C
{
  "prefix": "0.0.0.0/0",
  "gateway": "10.90.255.254"
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	IPv4静态路由表项
data.uuid	string	静态路由唯一资源ID
data.enable	bool	静态路由启禁用开关
data.description	string	描述信息 最小长度:0 最大长度:95
data.prefix	string	目的ip地址/掩码
data.gateway	string	下一跳ip地址
data.ifname	string	出接口名称 最小长度:1 最大长度:48
data.metric	uint32	路由度量值 最小值:0 最大值:255
data.linkdtEnable	bool	静态路由关联接口链路探测启禁用开关，默认false，不可修改
data.linkdt	string	静态路由关联链路探测，当前版本不可修改
data.tag	uint32	zebra路由标记，默认为0，当前版本不可修改

data.distance	uint8	路由管理距离，默认为1，当前版本不可修改 最小值:1 最大值:255
data.tableId	uint32	路由表Id，默认为0，当前版本不可修改 最小值:0 最大值:256
data.weight	uint8	IPv4静态路由权重，默认为1，当前版本不可修改 最小值:1 最大值:255
data.status	uint8	状态信息，状态字段不可设置或修改

请求成功样例：

```
{
  "data": {
    "enable": true,
    "metric": 0,
    "tag": 0,
    "uuid": "6A3FBCFDCD644ADFB90F9E087C22172C",
    "ifname": "auto",
    "tableId": 0,
    "linkdtEnable": false,
    "weight": 1,
    "distance": 1,
    "prefix": "0.0.0.0/0",
    "gateway": "10.90.255.254"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3.5.1.9 增量修改指定IPv4静态路由信息

功能说明：通过此接口，可以增量修改一条静态路由表项

1) API请求数据

PATCH https://{host}/api/v1/namespaces/@namespace/staticroutes/ipv4/@uuid

2) 输入参数

- Body参数

参数	类型	是否必选	描述
obj	object	是	IPv4静态路由表项
obj.enable	bool	否	静态路由启禁用开关 默认值:true
obj.description	string	否	描述信息 最小长度:0 最大长度:95
obj.prefix	string	是	目的ip地址/掩码
obj.gateway	string	是	下一跳ip地址

obj.ifname	string	否	出接口名称 最小长度:1 最大长度:48 默认值:true
obj.metric	uint32	否	路由度量值 最小值:0 最大值:255
obj.linkdtEnable	bool	否	静态路由关联接口链路探测启禁用开关，默认false，不可修改
obj.linkdt	string	否	静态路由关联链路探测，当前版本不可修改
obj.tag	uint32	否	zebra路由标记，默认为0，当前版本不可修改
obj.distance	uint8	否	路由管理距离，默认为1，当前版本不可修改 最小值:1 最大值:255 默认值:true
obj.tableId	uint32	否	路由表Id，默认为0，当前版本不可修改 最小值:0 最大值:256
obj.weight	uint8	否	IPv4静态路由权重，默认为1，当前版本不可修改 最小值:1 最大值:255 默认值:true

请求样例:

增量更新UUID为6A3FBCFDCD644ADFB90F9E087C22172C的ipv4静态路由配置,修改下一跳为200.200.255.254，度量值为10，禁用该路由，其他字段不修改

```
PATCH https://192.168.1.1/api/v1/namespaces/public/staticroutes/ipv4/6A3FBCFDCD644ADFB90F9E087C22172C
{
  "enable": false,
  "metric": 10,
  "gateway": "10.90.255.254"
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	IPv4静态路由表项
data.uuid	string	静态路由唯一资源ID
data.enable	bool	静态路由启禁用开关
data.description	string	描述信息 最小长度:0 最大长度:95
data.prefix	string	目的ip地址/掩码

data.gateway	string	下一跳ip地址
data.ifname	string	出接口名称 最小长度:1 最大长度:48
data.metric	uint32	路由度量值 最小值:0 最大值:255
data.linkdtEnable	bool	静态路由关联接口链路探测启禁用开关，默认false，不可修改
data.linkdt	string	静态路由关联链路探测，当前版本不可修改
data.tag	uint32	zebra路由标记，默认为0，当前版本不可修改
data.distance	uint8	路由管理距离，默认为1，当前版本不可修改 最小值:1 最大值:255
data.tableId	uint32	路由表Id，默认为0，当前版本不可修改 最小值:0 最大值:256
data.weight	uint8	IPv4静态路由权重，默认为1，当前版本不可修改 最小值:1 最大值:255
data.status	uint8	状态信息，状态字段不可设置或修改

请求成功样例：

```
{
  "data": {
    "enable": false,
    "metric": 10,
    "tag": 0,
    "uuid": "6A3FBCFDCD644ADFB90F9E087C22172C",
    "ifname": "auto",
    "tableId": 0,
    "linkdtEnable": false,
    "weight": 1,
    "distance": 1,
    "prefix": "0.0.0.0/0",
    "gateway": "200.200.255.254"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3.5.1.10 增量修改多条IPv4静态路由信息

功能说明：通过此接口，可以增量修改多条静态路由表项

1) API请求数据

PATCH <https://{host}/api/batch/v1/namespaces/@namespace/staticroutes/ipv4>

2) 输入参数

• Body参数

参数	类型	是否必选	描述
objs	object[]	是	IPv4路由表项列表
objs.enable	bool	否	静态路由启禁用开关 默认值:true
objs.description	string	否	描述信息 最小长度:0 最大长度:95
objs.prefix	string	是	目的ip地址/掩码
objs.gateway	string	是	下一跳ip地址
objs.ifname	string	否	出接口名称 最小长度:1 最大长度:48 默认值:true
objs.metric	uint32	否	路由度量值 最小值:0 最大值:255
objs.linkdtEnable	bool	否	静态路由关联接口链路探测启禁用开关，默认false，不可修改
objs.linkdt	string	否	静态路由关联链路探测，当前版本不可修改
objs.tag	uint32	否	zebra路由标记，默认为0，当前版本不可修改
objs.distance	uint8	否	路由管理距离，默认为1，当前版本不可修改 最小值:1 最大值:255 默认值:true
objs.tableId	uint32	否	路由表Id，默认为0，当前版本不可修改 最小值:0 最大值:256
objs.weight	uint8	否	IPv4静态路由权重，默认为1，当前版本不可修改 最小值:1 最大值:255 默认值:true

请求样例:

增量更新多条静态路由配置,修改其中一条静态路由的目的地址为200.200.0.0/16，度量值为0，修改其中另一条静态路由的度量值为0，其他字段不修改

```
PATCH https://192.168.1.1/api/batch/v1/namespaces/public/staticroutes/ipv4
[
  {
    "uuid": "48AAB0EE6F6944C88A3F1A5D97102D4F",
    "prefix": "200.200.0.0/16"
  },
  {
    "uuid": "6A3FBCFDCD644ADFB90F9E087C22172C",
    "metric": 0
  }
]
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	IPv4路由表项列表
data.uuid	string	静态路由唯一资源ID
data.enable	bool	静态路由启禁用开关
data.description	string	描述信息 最小长度:0 最大长度:95
data.prefix	string	目的ip地址/掩码
data.gateway	string	下一跳ip地址
data.ifname	string	出接口名称 最小长度:1 最大长度:48
data.metric	uint32	路由度量值 最小值:0 最大值:255
data.linkdtEnable	bool	静态路由关联接口链路探测启禁用开关，默认false，不可修改
data.linkdt	string	静态路由关联链路探测，当前版本不可修改
data.tag	uint32	zebra路由标记，默认为0，当前版本不可修改
data.distance	uint8	路由管理距离，默认为1，当前版本不可修改 最小值:1 最大值:255
data.tableId	uint32	路由表Id，默认为0，当前版本不可修改 最小值:0 最大值:256
data.weight	uint8	IPv4静态路由权重，默认为1，当前版本不可修改 最小值:1 最大值:255
data.status	uint8	状态信息，状态字段不可设置或修改

请求成功样例：

```
{
  "data": [
    {
      "enable": true,
      "metric": 2,
      "tag": 0,
      "uuid": "48AAB0EE6F6944C88A3F1A5D97102D4F",
      "ifname": "auto",
      "tableId": 0,
      "linkdtEnable": false,
      "weight": 1,
      "distance": 1,
      "prefix": "200.200.0.0/16",
      "gateway": "10.155.255.254"
    }
  ],
}
```

```
{
  "enable": false,
  "metric": 0,
  "tag": 0,
  "uuid": "6A3FBCFDCD644ADFB90F9E087C22172C",
  "ifname": "auto",
  "tableId": 0,
  "linkdtEnable": false,
  "weight": 1,
  "distance": 1,
  "prefix": "0.0.0.0/0",
  "gateway": "200.200.255.254"
},
{
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3.5.2 静态路由IPV6

3.5.2.1 获取单条IPv6静态路由信息

功能说明: 通过此接口, 可以获取指定单条静态路由表项

1) API请求数据

GET <https://{host}/api/v1/namespaces/@namespace/staticroutes/ipv6/@uuid>

2) 输入参数

请求样例:

获取UUID为B514650456124430ADE8E2B763B42158的ipv6静态路由的配置

GET <https://192.168.1.1/api/v1/namespaces/public/staticroutes/ipv6/B514650456124430ADE8E2B763B42158>

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	IPv6静态路由表项
data.uuid	string	静态路由唯一资源ID
data.enable	bool	静态路由启禁用开关
data.description	string	描述信息 最小长度:0 最大长度:95
data.prefix	string	目的IPv6地址/掩码
data.gateway	string	下一跳IPv6地址 格式:ip6-address

data.ifname	string	出接口名称 最小长度:1 最大长度:48
data.metric	uint32	路由度量值 最小值:256 最大值:1023
data.linkdtEnable	bool	静态路由关联接口链路探测启禁用开关，默认false，不可修改
data.linkdt	string	静态路由关联链路探测，当前版本不可修改
data.tag	uint32	zebra路由标记，默认为0，当前版本不可修改
data.distance	uint32	路由管理距离，默认为1，当前版本不可修改 最小值:1 最大值:255
data.weight	uint8	IPv6静态路由权重，默认为1，当前版本不可修改 最小值:1 最大值:255
data.status	uint8	状态信息，状态字段不可设置或修改

请求成功样例：

```
{
  "data": {
    "enable": true,
    "metric": 256,
    "tag": 0,
    "uuid": "B514650456124430ADE8E2B763B42158",
    "ifname": "auto",
    "weight": 1,
    "linkdtEnable": false,
    "distance": 1,
    "gateway": "3AC3:3A::1",
    "prefix": "4000::/64",
    "description": ""
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3.5.2.2 获取IPv6静态路由列表

功能说明：通过此接口，可以获取配置中心所有静态路由表项

1) API请求数据

GET

https://{host}/api/v1/namespaces/@namespace/staticroutes/ipv6

2) 输入参数

- URL参数

参数	类型	是否必选	描述
----	----	------	----

_start	int32	否	起始位置, 从 0 开始 最小值:0 最大值:2000000
metric	uint32	否	静态路由度量值 最小值:256 最大值:1023
prefix	string	否	目的IPv6地址/掩码
ifname	string	否	出接口名称 最小长度:1 最大长度:48
_length	int32	否	获取最大长度, 限制最大值, 如果数据不够可能返回小于 length 最小值:1 最大值:200 默认值:true
_search	string	否	模糊搜索关键字 最大长度:95
_select	string	否	选择字段, 字段用逗号分隔 最大长度:1000
gateway	string	否	下一跳IPv6地址 格式:ip6-address
_sortby	string	否	指定排序字段 最大长度:100
_order	string	否	指定排序方式 (正序/倒序) 默认值:true 枚举值:[asc:按升序排列 desc:按降序排列]

请求样例:

获取所有ipv6静态路由配置, 查询条件: 起始位置为0, 最多展示100条

GET https://192.168.1.1/api/v1/namespaces/public/staticroutes/ipv6?_start=0&_length=100

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	分页数据
data.totalItems	int32	总共多少项目
data.totalPages	int32	总共多少页
data.pageNumber	int32	当前页码, 从 1 开始
data.pageSize	int32	每页多大
data.itemsOffset	int32	当前条目偏移, 从 0 开始

data.itemLength	int32	数据列表长度
data.items	object[]	有效数据列表
data.items.uuid	string	静态路由唯一资源ID
data.items.enable	bool	静态路由启禁用开关
data.items.description	string	描述信息 最小长度:0 最大长度:95
data.items.prefix	string	目的IPv6地址/掩码
data.items.gateway	string	下一跳IPv6地址 格式:ip6-address
data.items.ifname	string	出接口名称 最小长度:1 最大长度:48
data.items.metric	uint32	路由度量值 最小值:256 最大值:1023
data.items.linkdtEnable	bool	静态路由关联接口链路探测启禁用开关，默认false，不可修改
data.items.linkdt	string	静态路由关联链路探测，当前版本不可修改
data.items.tag	uint32	zebra路由标记，默认为0，当前版本不可修改
data.items.distance	uint32	路由管理距离，默认为1，当前版本不可修改 最小值:1 最大值:255
data.items.weight	uint8	IPv6静态路由权重，默认为1，当前版本不可修改 最小值:1 最大值:255
data.items.status	uint8	状态信息，状态字段不可设置或修改

请求成功样例：

```
{
  "data": {
    "itemsOffset": 0,
    "items": [
      {
        "enable": true,
        "metric": 256,
        "tag": 0,
        "uuid": "B514650456124430ADE8E2B763B42158",
        "ifname": "auto",
        "weight": 1,
        "linkdtEnable": false,
        "distance": 1,
        "gateway": "3AC3:3A::1",
        "prefix": "4000::/64",
        "description": ""
      }
    ],
    "totalItems": 1,
    "pageNumber": 1,
    "itemLength": 1,
    "totalPages": 1,
    "pageSize": 100
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3.5.2.3 添加一条IPv6静态路由信息

功能说明：通过此接口，可以添加一条静态路由表项

1) API请求数据

POST <https://{host}/api/v1/namespaces/@namespace/staticroutes/ipv6>

2) 输入参数

- Body参数

参数	类型	是否必选	描述
obj	object	是	IPv6静态路由表项
obj.enable	bool	否	静态路由启禁用开关 默认值:true
obj.description	string	否	描述信息 最小长度:0 最大长度:95
obj.prefix	string	是	目的IPv6地址/掩码
obj.gateway	string	是	下一跳IPv6地址 格式:ip6-address
obj.ifname	string	否	出接口名称 最小长度:1 最大长度:48 默认值:true
obj.metric	uint32	否	路由度量值 最小值:256 最大值:1023 默认值:true
obj.linkdtEnable	bool	否	静态路由关联接口链路探测启禁用开关，默认false，不可修改
obj.linkdt	string	否	静态路由关联链路探测，当前版本不可修改
obj.tag	uint32	否	zebra路由标记，默认为0，当前版本不可修改
obj.distance	uint32	否	路由管理距离，默认为1，当前版本不可修改 最小值:1 最大值:255 默认值:true
obj.weight	uint8	否	IPv6静态路由权重，默认为1，当前版本不可修改 最小值:1 最大值:255 默认值:true

请求样例:

```
添加一条ipv6静态路由

POST https://192.168.1.1/api/v1/namespaces/public/staticroutes/ipv6
{
  "enable": true,
  "metric": 256,
  "ifname": "auto",
  "gateway": "3ac3:3a::1",
  "prefix": "4000::/64",
  "description": ""
}
```

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	IPv6静态路由表项
data.uuid	string	静态路由唯一资源ID
data.enable	bool	静态路由启禁用开关
data.description	string	描述信息 最小长度:0 最大长度:95
data.prefix	string	目的IPv6地址/掩码
data.gateway	string	下一跳IPv6地址 格式:ip6-address
data.ifname	string	出接口名称 最小长度:1 最大长度:48
data.metric	uint32	路由度量值 最小值:256 最大值:1023
data.linkdtEnable	bool	静态路由关联接口链路探测启禁用开关，默认false，不可修改
data.linkdt	string	静态路由关联链路探测，当前版本不可修改
data.tag	uint32	zebra路由标记，默认为0，当前版本不可修改
data.distance	uint32	路由管理距离，默认为1，当前版本不可修改 最小值:1 最大值:255
data.weight	uint8	IPv6静态路由权重，默认为1，当前版本不可修改 最小值:1 最大值:255
data.status	uint8	状态信息，状态字段不可设置或修改

请求成功样例:

```
{
  "data": {
    "enable": true,
    "metric": 256,
```



```

    "tag": 0,
    "uuid": "B514650456124430ADE8E2B763B42158",
    "ifname": "auto",
    "weight": 1,
    "linkdtEnable": false,
    "distance": 1,
    "gateway": "3AC3:3A::1",
    "prefix": "4000::/64",
    "description": ""
  },
  "code": 0,
  "message": "成功"
}

```

请求失败样例：

```

{
  "data": "",
  "code": 22,
  "message": "无效参数"
}

```

3.5.2.4 添加多条IPv6静态路由信息

功能说明：通过此接口，可以添加多条静态路由表项

1) API请求数据

POST <https://{host}/api/batch/v1/namespaces/@namespace/staticroutes/ipv6>

2) 输入参数

- Body参数

参数	类型	是否必选	描述
objs	object[]	是	IPv6静态路由表项列表
objs.enable	bool	否	静态路由启禁用开关 默认值:true
objs.description	string	否	描述信息 最小长度:0 最大长度:95
objs.prefix	string	是	目的IPv6地址/掩码
objs.gateway	string	是	下一跳IPv6地址 格式:ip6-address
objs.ifname	string	否	出接口名称 最小长度:1 最大长度:48 默认值:true
objs.metric	uint32	否	路由度量值 最小值:256 最大值:1023 默认值:true
objs.linkdtEnable	bool	否	静态路由关联接口链路探测启禁用开关，默认false，不可修改
objs.linkdt	string	否	静态路由关联链路探测，当前版本不可修改
objs.tag	uint32	否	zebra路由标记，默认为0，当前版本不可修改

objs.distance	uint32	否	路由管理距离，默认为1，当前版本不可修改 最小值:1 最大值:255 默认值:true
objs.weight	uint8	否	IPv6静态路由权重，默认为1，当前版本不可修改 最小值:1 最大值:255 默认值:true

请求样例:

添加多条ipv6静态路由

POST https://192.168.1.1/api/batch/v1/namespaces/public/staticroutes/ipv6

```
[
  {
    "metric": 270,
    "gateway": "3AC3:3A::1",
    "prefix": "4000::/64",
    "ifname": "auto"
  },
  {
    "metric": 280,
    "gateway": "2000::1",
    "prefix": "3000::/64",
    "ifname": "eth0"
  }
]
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	IPv6静态路由表项列表
data.uuid	string	静态路由唯一资源ID
data.enable	bool	静态路由启禁用开关
data.description	string	描述信息 最小长度:0 最大长度:95
data.prefix	string	目的IPv6地址/掩码
data.gateway	string	下一跳IPv6地址 格式:ip6-address
data.ifname	string	出接口名称 最小长度:1 最大长度:48
data.metric	uint32	路由度量值 最小值:256 最大值:1023
data.linkdtEnable	bool	静态路由关联接口链路探测启禁用开关，默认false，不可修改

data.linkdt	string	静态路由关联链路探测，当前版本不可修改
data.tag	uint32	zebra路由标记，默认为0，当前版本不可修改
data.distance	uint32	路由管理距离，默认为1，当前版本不可修改 最小值:1 最大值:255
data.weight	uint8	IPv6静态路由权重，默认为1，当前版本不可修改 最小值:1 最大值:255
data.status	uint8	状态信息，状态字段不可设置或修改

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3.5.2.5 删除指定的IPv6静态路由信息

功能说明：通过此接口，可以删除一条静态路由表项

1) API请求数据

DELETE

https://{host}/api/v1/namespaces/@namespace/staticroutes/ipv6/@uuid

2) 输入参数

请求样例：

```
删除UUID为ABC7B4A171A74BC98E5599037410B4B3的ipv6静态路由

DELETE https://192.168.1.1/api/v1/namespaces/public/staticroutes/ipv6/ABC7B4A171A74BC98E5599037410B4B3
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	IPv6静态路由表项
data.uuid	string	静态路由唯一资源ID
data.enable	bool	静态路由启禁用开关
data.description	string	描述信息 最小长度:0 最大长度:95
data.prefix	string	目的IPv6地址/掩码
data.gateway	string	下一跳IPv6地址 格式:ip6-address
data.ifname	string	出接口名称 最小长度:1 最大长度:48

data.metric	uint32	路由度量值 最小值:256 最大值:1023
data.linkdtEnable	bool	静态路由关联接口链路探测启禁用开关，默认false，不可修改
data.linkdt	string	静态路由关联链路探测，当前版本不可修改
data.tag	uint32	zebra路由标记，默认为0，当前版本不可修改
data.distance	uint32	路由管理距离，默认为1，当前版本不可修改 最小值:1 最大值:255
data.weight	uint8	IPv6静态路由权重，默认为1，当前版本不可修改 最小值:1 最大值:255
data.status	uint8	状态信息，状态字段不可设置或修改

请求成功样例：

```
{
  "data": {
    "enable": true,
    "metric": 256,
    "tag": 0,
    "uuid": "ABC7B4A171A74BC98E5599037410B4B3",
    "ifname": "auto",
    "linkdtEnable": false,
    "weight": 1,
    "gateway": "3AC3:3A::1",
    "prefix": "6000::/64",
    "distance": 1
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3.5.2.6 通过参数删除指定的IPv6静态路由信息

功能说明：通过此接口，可以传入关键属性来删除一条静态路由表项

1) API请求数据

POST

https://{host}/api/v1/namespaces/@namespace/staticroutes/ipv6?_method=delete

2) 输入参数

- Body参数

参数	类型	是否必选	描述
obj	object	是	IPv6静态路由表项
obj.enable	bool	否	静态路由启禁用开关 默认值:true

obj.description	string	否	描述信息 最小长度:0 最大长度:95
obj.prefix	string	是	目的IPv6地址/掩码
obj.gateway	string	是	下一跳IPv6地址 格式:ip6-address
obj.ifname	string	否	出接口名称 最小长度:1 最大长度:48 默认值:true
obj.metric	uint32	否	路由度量值 最小值:256 最大值:1023 默认值:true
obj.linkdtEnable	bool	否	静态路由由关联接口链路探测启禁用开关，默认false，不可修改
obj.linkdt	string	否	静态路由由关联链路探测，当前版本不可修改
obj.tag	uint32	否	zebra路由标记，默认为0，当前版本不可修改
obj.distance	uint32	否	路由管理距离，默认为1，当前版本不可修改 最小值:1 最大值:255 默认值:true
obj.weight	uint8	否	IPv6静态路由权重，默认为1，当前版本不可修改 最小值:1 最大值:255 默认值:true

请求样例：

通过参数删除指定的ipv6静态路由信息，删除目的地址为3000::/64，下一跳2000::2，接口为eth0，度量值为280的静态路由

```
POST https://192.168.1.1/api/v1/namespaces/public/staticroutes/ipv6?_method=DELETE
{
  "metric": 280,
  "gateway": "2000::2",
  "prefix": "3000::/64",
  "ifname": "eth0"
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	IPv6静态路由表项
data.uuid	string	静态路由唯一资源ID
data.enable	bool	静态路由启禁用开关

data.description	string	描述信息 最小长度:0 最大长度:95
data.prefix	string	目的IPv6地址/掩码
data.gateway	string	下一跳IPv6地址 格式:ip6-address
data.ifname	string	出接口名称 最小长度:1 最大长度:48
data.metric	uint32	路由度量值 最小值:256 最大值:1023
data.linkdtEnable	bool	静态路由关联接口链路探测启禁用开关，默认false，不可修改
data.linkdt	string	静态路由关联链路探测，当前版本不可修改
data.tag	uint32	zebra路由标记，默认为0，当前版本不可修改
data.distance	uint32	路由管理距离，默认为1，当前版本不可修改 最小值:1 最大值:255
data.weight	uint8	IPv6静态路由权重，默认为1，当前版本不可修改 最小值:1 最大值:255
data.status	uint8	状态信息，状态字段不可设置或修改

请求成功样例：

```
{
  "data": {
    "enable": false,
    "metric": 280,
    "tag": 0,
    "uuid": "0C98777BF8354629AA9625FB3E8040DF",
    "ifname": "eth0",
    "linkdtEnable": false,
    "weight": 1,
    "gateway": "2000::2",
    "prefix": "3000::/64",
    "distance": 1
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3.5.2.7 删除多条IPv6静态路由信息

功能说明：通过此接口，可以删除多条静态路由表项

1) API请求数据

POST

https://{host}/api/batch/v1/namespaces/@namespace/staticroutes/ipv6?_method=delete

2) 输入参数

• Body参数

参数	类型	是否必选	描述
objs	object[]	是	IPv6静态路由表项列表
objs.enable	bool	否	静态路由启禁用开关 默认值:true
objs.description	string	否	描述信息 最小长度:0 最大长度:95
objs.prefix	string	是	目的IPv6地址/掩码
objs.gateway	string	是	下一跳IPv6地址 格式:ip6-address
objs.ifname	string	否	出接口名称 最小长度:1 最大长度:48 默认值:true
objs.metric	uint32	否	路由度量值 最小值:256 最大值:1023 默认值:true
objs.linkdtEnable	bool	否	静态路由关联接口链路探测启禁用开关, 默认false, 不可修改
objs.linkdt	string	否	静态路由关联链路探测, 当前版本不可修改
objs.tag	uint32	否	zebra路由标记, 默认为0, 当前版本不可修改
objs.distance	uint32	否	路由管理距离, 默认为1, 当前版本不可修改 最小值:1 最大值:255 默认值:true
objs.weight	uint8	否	IPv6静态路由权重, 默认为1, 当前版本不可修改 最小值:1 最大值:255 默认值:true

请求样例:

删除多条ipv6静态路由信息

```
POST https://192.168.1.1/api/batch/v1/namespaces/public/staticroutes/ipv6?_method=DELETE
[
  {
    "uuid": "ABC7B4A171A74BC98E5599037410B4B3"
  },
  {
    "uuid": "0C98777BF8354629AA9625FB3E8040DF"
  }
]
```

]

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	IPv6静态路由表项列表
data.uuid	string	静态路由唯一资源ID
data.enable	bool	静态路由启禁用开关
data.description	string	描述信息 最小长度:0 最大长度:95
data.prefix	string	目的IPv6地址/掩码
data.gateway	string	下一跳IPv6地址 格式:ip6-address
data.ifname	string	出接口名称 最小长度:1 最大长度:48
data.metric	uint32	路由度量值 最小值:256 最大值:1023
data.linkdtEnable	bool	静态路由关联接口链路探测启禁用开关，默认false，不可修改
data.linkdt	string	静态路由关联链路探测，当前版本不可修改
data.tag	uint32	zebra路由标记，默认为0，当前版本不可修改
data.distance	uint32	路由管理距离，默认为1，当前版本不可修改 最小值:1 最大值:255
data.weight	uint8	IPv6静态路由权重，默认为1，当前版本不可修改 最小值:1 最大值:255
data.status	uint8	状态信息，状态字段不可设置或修改

请求成功样例：

```
{
  "data": [
    {
      "enable": true,
      "metric": 256,
      "tag": 0,
      "uuid": "ABC7B4A171A74BC98E5599037410B4B3",
      "ifname": "auto",
      "linkdtEnable": false,
      "weight": 1,
      "gateway": "3AC3:3A::1",
      "prefix": "6000::/64",
      "distance": 1
    }
  ],
  "enable": false,
```



```

        "metric": 280,
        "tag": 0,
        "uuid": "0C98777BF8354629AA9625FB3E8040DF",
        "ifname": "eth0",
        "linkdtEnable": false,
        "weight": 1,
        "gateway": "2000::2",
        "prefix": "3000::/64",
        "distance": 1
    },
    "code": 0,
    "message": "成功"
}

```

请求失败样例:

```

{
  "data": "",
  "code": 22,
  "message": "无效参数"
}

```

3.5.2.8 全量修改指定的IPv6静态路由信息

功能说明: 通过此接口, 可以全量修改一条静态路由表项

1) API请求数据

PUT <https://{host}/api/v1/namespaces/@namespace/staticroutes/ipv6/@uuid>

2) 输入参数

- Body参数

参数	类型	是否必选	描述
obj	object	是	IPv6静态路由表项
obj.enable	bool	否	静态路由启禁用开关 默认值:true
obj.description	string	否	描述信息 最小长度:0 最大长度:95
obj.prefix	string	是	目的IPv6地址/掩码
obj.gateway	string	是	下一跳IPv6地址 格式:ip6-address
obj.ifname	string	否	出接口名称 最小长度:1 最大长度:48 默认值:true
obj.metric	uint32	否	路由度量值 最小值:256 最大值:1023 默认值:true
obj.linkdtEnable	bool	否	静态路由由关联接口链路探测启禁用开关, 默认false, 不可修改
obj.linkdt	string	否	静态路由由关联链路探测, 当前版本不可修改
obj.tag	uint32	否	zebra路由标记, 默认为0, 当前版本不可修改

obj.distance	uint32	否	路由管理距离，默认为1，当前版本不可修改 最小值:1 最大值:255 默认值:true
obj.weight	uint8	否	IPv6静态路由权重，默认为1，当前版本不可修改 最小值:1 最大值:255 默认值:true

请求样例:

全量更新UUID为ABC7B4A171A74BC98E5599037410B4B3的ipv6静态路由配置,修改下一跳为3AC3:3A::2, 目的地址5000::/64, 其他不填自动为默认值

```
PUT https://192.168.1.1/api/v1/namespaces/public/staticroutes/ipv6/ABC7B4A171A74BC98E5599037410B4B3
{
  "prefix": "5000::/64",
  "gateway": "3AC3:3A::2"
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	IPv6静态路由表项
data.uuid	string	静态路由唯一资源ID
data.enable	bool	静态路由启禁用开关
data.description	string	描述信息 最小长度:0 最大长度:95
data.prefix	string	目的IPv6地址/掩码
data.gateway	string	下一跳IPv6地址 格式:ip6-address
data.ifname	string	出接口名称 最小长度:1 最大长度:48
data.metric	uint32	路由度量值 最小值:256 最大值:1023
data.linkdtEnable	bool	静态路由关联接口链路探测启禁用开关，默认false，不可修改
data.linkdt	string	静态路由关联链路探测，当前版本不可修改
data.tag	uint32	zebra路由标记，默认为0，当前版本不可修改
data.distance	uint32	路由管理距离，默认为1，当前版本不可修改 最小值:1 最大值:255

data.weight	uint8	IPv6静态路由权重，默认为1，当前版本不可修改 最小值:1 最大值:255
data.status	uint8	状态信息，状态字段不可设置或修改

请求成功样例：

```
{
  "data": {
    "enable": true,
    "metric": 256,
    "tag": 0,
    "uuid": "ABC7B4A171A74BC98E5599037410B4B3",
    "ifname": "auto",
    "weight": 1,
    "linkdtEnable": false,
    "distance": 1,
    "gateway": "3AC3:3A::2",
    "prefix": "5000::/64",
    "description": ""
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3.5.2.9 增量修改指定IPv6静态路由信息

功能说明：通过此接口，可以增量修改一条静态路由表项

1) API请求数据

PATCH

https://{host}/api/v1/namespaces/@namespace/staticroutes/ipv6/@uuid

2) 输入参数

• Body参数

参数	类型	是否必选	描述
obj	object	是	IPv6静态路由表项
obj.enable	bool	否	静态路由启禁用开关 默认值:true
obj.description	string	否	描述信息 最小长度:0 最大长度:95
obj.prefix	string	是	目的IPv6地址/掩码
obj.gateway	string	是	下一跳IPv6地址 格式:ip6-address
obj.ifname	string	否	出接口名称 最小长度:1 最大长度:48 默认值:true

obj.metric	uint32	否	路由度量值 最小值:256 最大值:1023 默认值:true
obj.linkdtEnable	bool	否	静态路由关联接口链路探测启禁用开关，默认false，不可修改
obj.linkdt	string	否	静态路由关联链路探测，当前版本不可修改
obj.tag	uint32	否	zebra路由标记，默认为0，当前版本不可修改
obj.distance	uint32	否	路由管理距离，默认为1，当前版本不可修改 最小值:1 最大值:255 默认值:true
obj.weight	uint8	否	IPv6静态路由权重，默认为1，当前版本不可修改 最小值:1 最大值:255 默认值:true

请求样例：

增量更新UUID为0C98777BF8354629AA9625FB3E8040DF的ipv6静态路由配置,修改下一跳为2000::2，度量值为256，禁用该路由，其他字段不修改

```
PATCH https://192.168.1.1/api/v1/namespaces/public/staticroutes/ipv6/0C98777BF8354629AA9625FB3E8040DF
{
  "enable": false,
  "metric": 256,
  "gateway": "2000::2"
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	IPv6静态路由表项
data.uuid	string	静态路由唯一资源ID
data.enable	bool	静态路由启禁用开关
data.description	string	描述信息 最小长度:0 最大长度:95
data.prefix	string	目的IPv6地址/掩码
data.gateway	string	下一跳IPv6地址 格式:ip6-address
data.ifname	string	出接口名称 最小长度:1 最大长度:48

data.metric	uint32	路由度量值 最小值:256 最大值:1023
data.linkdtEnable	bool	静态路由关联接口链路探测启禁用开关，默认false，不可修改
data.linkdt	string	静态路由关联链路探测，当前版本不可修改
data.tag	uint32	zebra路由标记，默认为0，当前版本不可修改
data.distance	uint32	路由管理距离，默认为1，当前版本不可修改 最小值:1 最大值:255
data.weight	uint8	IPv6静态路由权重，默认为1，当前版本不可修改 最小值:1 最大值:255
data.status	uint8	状态信息，状态字段不可设置或修改

请求成功样例：

```
{
  "data": {
    "enable": false,
    "metric": 256,
    "tag": 0,
    "uuid": "0C98777BF8354629AA9625FB3E8040DF",
    "ifname": "eth0",
    "weight": 1,
    "linkdtEnable": false,
    "distance": 1,
    "gateway": "2000::2",
    "prefix": "3000::/64",
    "description": ""
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3.5.2.10 增量修改多条IPv6静态路由信息

功能说明：通过此接口，可以增量修改多条静态路由表项

1) API请求数据

PATCH

https://{host}/api/batch/v1/namespaces/@namespace/staticroutes/ipv6

2) 输入参数

- Body参数

参数	类型	是否必选	描述
objs	object[]	是	IPv6静态路由表项列表
objs.enable	bool	否	静态路由启禁用开关 默认值:true

objs.description	string	否	描述信息 最小长度:0 最大长度:95
objs.prefix	string	是	目的IPv6地址/掩码
objs.gateway	string	是	下一跳IPv6地址 格式:ip6-address
objs.ifname	string	否	出接口名称 最小长度:1 最大长度:48 默认值:true
objs.metric	uint32	否	路由度量值 最小值:256 最大值:1023 默认值:true
objs.linkdtEnable	bool	否	静态路由关联接口链路探测启禁用开关，默认false，不可修改
objs.linkdt	string	否	静态路由关联链路探测，当前版本不可修改
objs.tag	uint32	否	zebra路由标记，默认为0，当前版本不可修改
objs.distance	uint32	否	路由管理距离，默认为1，当前版本不可修改 最小值:1 最大值:255 默认值:true
objs.weight	uint8	否	IPv6静态路由权重，默认为1，当前版本不可修改 最小值:1 最大值:255 默认值:true

请求样例:

增量更新多条静态路由由配置,修改其中一条静态路由的目的地址为6000::/64，修改其中另一条静态路由的度量值为300，其他字段不修改

```
PATCH https://192.168.1.1/api/batch/v1/namespaces/public/staticroutes/ipv6
[
  {
    "uuid": "ABC7B4A171A74BC98E5599037410B4B3",
    "prefix": "6000::/64"
  },
  {
    "uuid": "0C98777BF8354629AA9625FB3E8040DF",
    "metric": 300
  }
]
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	IPv6静态路由表项列表

data.uuid	string	静态路由唯一资源ID
data.enable	bool	静态路由启禁用开关
data.description	string	描述信息 最小长度:0 最大长度:95
data.prefix	string	目的IPv6地址/掩码
data.gateway	string	下一跳IPv6地址 格式:ip6-address
data.ifname	string	出接口名称 最小长度:1 最大长度:48
data.metric	uint32	路由度量值 最小值:256 最大值:1023
data.linkdtEnable	bool	静态路由关联接口链路探测启禁用开关，默认false，不可修改
data.linkdt	string	静态路由关联链路探测，当前版本不可修改
data.tag	uint32	zebra路由标记，默认为0，当前版本不可修改
data.distance	uint32	路由管理距离，默认为1，当前版本不可修改 最小值:1 最大值:255
data.weight	uint8	IPv6静态路由权重，默认为1，当前版本不可修改 最小值:1 最大值:255
data.status	uint8	状态信息，状态字段不可设置或修改

请求成功样例：

```
{
  "data": [
    {
      "enable": true,
      "metric": 256,
      "tag": 0,
      "uuid": "ABC7B4A171A74BC98E5599037410B4B3",
      "ifname": "auto",
      "linkdtEnable": false,
      "weight": 1,
      "gateway": "3AC3:3A::1",
      "prefix": "6000::/64",
      "distance": 1
    },
    {
      "enable": false,
      "metric": 280,
      "tag": 0,
      "uuid": "0C98777BF8354629AA9625FB3E8040DF",
      "ifname": "eth0",
      "linkdtEnable": false,
      "weight": 1,
      "gateway": "2000::2",
      "prefix": "3000::/64",
      "distance": 1
    }
  ],
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3.5.3 策略路由IPV4

3.5.3.1 获取单条IPv4策略路由信息

功能说明：通过此接口，获取配置中心单条IPv4策略路由表项

1) API请求数据

GET

https://{host}/api/v1/namespaces/@namespace/pbrs/ipv4/@name

2) 输入参数

- URL参数

参数	类型	是否必选	描述
_select	string	否	选择字段, 字段用逗号分隔 最大长度:1000

请求样例：

获取单条ipv4策略路由的配置信息，查询条件：查看名为test的策略路由

GET https://192.168.1.1/api/v1/namespaces/public/pbrs/ipv4/test

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	策略路由属性模板
data.uuid	string	策略路由唯一资源ID
data.name	string	策略路由名称 最小长度:1 最大长度:95
data.enable	bool	启禁用开关
data.description	string	描述字段 最小长度:0 最大长度:95
data.schedule	string	时间计划名称
data.srcZones	string[]	关联区域列表
data.srcIpGroups	string[]	源地址引用IP组列表
data.dstIpGroups	string[]	目的地址引用IP组列表
data.dstIsp	string[]	目的地址关联ISP列表
data.dstCountryArea	string[]	目的地址关联国家/地区列表

data.services	string[]	服务对象列表
data.applications	string[]	关联应用识别列表
data.lbMethod	string	多线路模式下，接口选择策略 枚举值:[POLL:轮询 BANDWIDTH:带宽比例 WEIGHTED_MINIMUM_FLOW:加权最小流量 PRIORITY_FRONT_LINE:优先使用前面线路]
data.tableId	uint32	策略路由表ID，默认为0，当前版本不可修改
data.status	uint8	策略路由的状态值，状态字段不可设置或修改
data.position	int32	策略的位置，状态字段不可设置或修改
data.pbrType	string	策略路由类型 枚举值:[SRCADDRESS:源地址策略路由 MULT_LINE:多线路负载策略路由]
data.outif	object[]	策略路由出接口列表
data.outif.outifName	string	出接口名称
data.outif.gateway	string	下一跳ip地址
data.outif.linkdt	string	引用链路探测对象名称

请求成功样例：

```
{
  "data": {
    "enable": true,
    "applications": [
      "全部"
    ],
    "srcIpGroups": [
      "全部"
    ],
    "dstCountryArea": [],
    "dstIpGroups": [
      "全部"
    ],
    "uuid": "FBB3947A93714437AECB21B7ABA014DF",
    "outif": [
      {
        "outifName": "eth0",
        "gateway": "10.155.255.254"
      }
    ],
    "pbrType": "SRCADDRESS",
    "position": 0,
    "srcZones": [
      "manage"
    ],
    "dstIsp": [],
    "services": [
      "any"
    ],
    "name": "test",
    "schedule": "全天",
    "description": ""
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3.5.3.2 获取IPv4策略路由列表

功能说明：通过此接口，获取配置中心所有IPv4策略路由表项

1) API请求数据

GET <https://{host}/api/v1/namespaces/@namespace/pbrs/ipv4>

2) 输入参数

• URL参数

参数	类型	是否必选	描述
_start	int32	否	起始位置, 从 0 开始 最小值:0 最大值:2000000
position	int32	否	拿取指定位置的配置
_search	string	否	模糊搜索关键字 最大长度:95
_select	string	否	选择字段, 字段用逗号分隔 最大长度:1000
movesearch	string	否	移动搜索参数项 最小长度:1 最大长度:95
_order	string	否	指定排序方式（正序/倒序） 默认值:true 枚举值:[asc:按升序排列 desc:按降序排列]
_length	int32	否	获取最大长度，限制最大值，如果数据不够可能返回小于 length 最小值:1 最大值:200 默认值:true
_sortby	string	否	指定排序字段 最大长度:100

请求样例：

获取多条ipv4策略路由的配置信息，查询条件：起始位置为0，最多展示100条

GET https://192.168.1.1/api/v1/namespaces/public/pbrs/ipv4?_start=0&_length=100

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码

message	string	错误信息
data	object	分页数据
data.totalItems	int32	总共多少项目
data.totalPages	int32	总共多少页
data.pageNumber	int32	当前页码, 从 1 开始
data.pageSize	int32	每页多大
data.itemsOffset	int32	当前条目偏移, 从 0 开始
data.itemLength	int32	数据列表长度
data.items	object[]	有效数据列表
data.items.uuid	string	策略路由唯一资源ID
data.items.name	string	策略路由名称 最小长度:1 最大长度:95
data.items.enable	bool	启禁用开关
data.items.description	string	描述字段 最小长度:0 最大长度:95
data.items.schedule	string	时间计划名称
data.items.srcZones	string[]	关联区域列表
data.items.srcIpGroups	string[]	源地址引用IP组列表
data.items.dstIpGroups	string[]	目的地址引用IP组列表
data.items.dstIsp	string[]	目的地址关联ISP列表
data.items.dstCountryArea	string[]	目的地址关联国家/地区列表
data.items.services	string[]	服务对象列表
data.items.applications	string[]	关联应用识别列表
data.items.lbMethod	string	多线路模式下, 接口选择策略 枚举值:[POLL:轮询 BANDWIDTH:带宽比例 WEIGHTED_MINIMUM_FLOW:加权最小流量 PRIORITY_FRONT_LINE:优先使用前面线路]
data.items.tableId	uint32	策略路由表ID, 默认为0, 当前版本不可修改
data.items.status	uint8	策略路由的状态值, 状态字段不可设置或修改
data.items.position	int32	策略的位置, 状态字段不可设置或修改
data.items.pbrType	string	策略路由类型 枚举值:[SRCADDRESS:源地址策略路由 MULT_LINE:多线路负载策略路由]
data.items.outif	object[]	策略路由出接口列表

data.items.outif.outifName	string	出接口名称
data.items.outif.gateway	string	下一跳ip地址
data.items.outif.linkdt	string	引用链路探测对象名称

请求成功样例：

```
{
  "data": {
    "itemsOffset": 0,
    "items": [
      {
        "enable": true,
        "applications": [
          "全部"
        ],
        "srcIpGroups": [
          "全部"
        ],
        "dstCountryArea": [],
        "dstIpGroups": [
          "全部"
        ],
        "uuid": "FBB3947A93714437AECB21B7ABA014DF",
        "outif": [
          {
            "outifName": "eth0",
            "gateway": "10.155.255.254"
          }
        ],
        "pbrType": "SRCADDRESS",
        "position": 0,
        "srcZones": [
          "manage"
        ],
        "dstIsp": [],
        "services": [
          "any"
        ],
        "name": "test",
        "schedule": "全天",
        "description": ""
      }
    ],
    "totalItems": 1,
    "pageNumber": 1,
    "itemLength": 1,
    "totalPages": 1,
    "pageSize": 100
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3.5.3.3 添加一条IPv4策略路由信息

功能说明：通过此接口，添加单条IPv4策略路由表项

1) API请求数据

POST https://{host}/api/v1/namespaces/@namespace/pbrs/ipv4

2) 输入参数

• URL参数

参数	类型	是否必选	描述
_where	string	否	指定位置, 用于移动/插入 枚举值:[top:顶部 bottom:底部 before:在目标位置之前 after:在目标位置之后]
_dest	string	否	目标位置关键字对应的值 最大长度:100
_key	string	否	目标位置关键字 枚举值:[position:指定目标位置关键字为 position name:指定目标位置关键字为 name]

• Body参数

参数	类型	是否必选	描述
obj	object	是	策略路由属性模板
obj.name	string	是	策略路由名称 最小长度:1 最大长度:95
obj.enable	bool	否	启禁用开关 默认值:true
obj.description	string	否	描述字段 最小长度:0 最大长度:95
obj.schedule	string	否	时间计划名称
obj.srcZones	string[]	否	关联区域列表
obj.srcIpGroups	string[]	否	源地址引用IP组列表
obj.dstIpGroups	string[]	否	目的地址引用IP组列表
obj.dstIsp	string[]	否	目的地址关联ISP列表
obj.dstCountryArea	string[]	否	目的地址关联国家/地区列表
obj.services	string[]	否	服务对象列表
obj.applications	string[]	否	关联应用识别列表

obj.lbMethod	string	否	多线路模式下，接口选择策略 枚举值:[POLL:轮询 BANDWIDTH:带宽比例 WEIGHTED_MINIMUM_FLOW:加权最小流量 PRIORITY_FRONT_LINE:优先使用前面线路]
obj.tableId	uint32	否	策略路由表ID，默认为0，当前版本不可修改
obj.status	uint8	否	策略路由的状态值，状态字段不可设置或修改
obj.position	int32	否	策略的位置，状态字段不可设置或修改
obj.pbrType	string	是	策略路由类型 枚举值:[SRCADDRESS:源地址策略路由 MULT_LINE:多线路负载策略路由]
obj.outif	object[]	否	策略路由出接口列表
obj.outif.outifName	string	否	出接口名称
obj.outif.gateway	string	否	下一跳ip地址
obj.outif.linkdt	string	否	引用链路探测对象名称

请求样例：

新增单条ipv4策略路由

POST https://192.168.1.1/api/v1/namespaces/public/pbrs/ipv4

```
{
  "enable": true,
  "applications": [
    "全部"
  ],
  "srcIpGroups": [
    "全部"
  ],
  "outif": [
    {
      "outifName": "eth0",
      "gateway": "10.155.255.254"
    }
  ],
  "dstIpGroups": [
    "全部"
  ],
  "srcZones": [
    "manage"
  ],
  "pbrType": "SRCADDRESS",
  "services": [
    "any"
  ],
  "schedule": "全天",
  "name": "test",
  "description": "create test"
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	策略路由属性模板
data.uuid	string	策略路由唯一资源ID
data.name	string	策略路由名称 最小长度:1 最大长度:95
data.enable	bool	启禁用开关
data.description	string	描述字段 最小长度:0 最大长度:95
data.schedule	string	时间计划名称
data.srcZones	string[]	关联区域列表
data.srcIpGroups	string[]	源地址引用IP组列表
data.dstIpGroups	string[]	目的地址引用IP组列表
data.dstIsp	string[]	目的地址关联ISP列表
data.dstCountryArea	string[]	目的地址关联国家/地区列表
data.services	string[]	服务对象列表
data.applications	string[]	关联应用识别列表
data.lbMethod	string	多线路模式下，接口选择策略 枚举值:[POLL:轮询 BANDWIDTH:带宽比例 WEIGHTED_MINIMUM_FLOW:加权最小流量 PRIORITY_FRONT_LINE:优先使用前面线路]
data.tableId	uint32	策略路由表ID，默认为0，当前版本不可修改
data.status	uint8	策略路由的状态值，状态字段不可设置或修改
data.position	int32	策略的位置，状态字段不可设置或修改
data.pbrType	string	策略路由类型 枚举值:[SRCADDRESS:源地址策略路由 MULT_LINE:多线路负载策略路由]
data.outif	object[]	策略路由出接口列表
data.outif.outifName	string	出接口名称
data.outif.gateway	string	下一跳ip地址
data.outif.linkdt	string	引用链路探测对象名称

请求成功样例：

```
{
  "data": {
    "enable": true,
    "applications": [
      "全部"
    ],
    "srcIpsGroups": [
      "全部"
    ],
    "dstCountryArea": [],
    "dstIpsGroups": [
      "全部"
    ],
    "uuid": "FBB3947A93714437AECB21B7ABA014DF",
    "outif": [
      {
        "outifName": "eth0",
        "gateway": "10.155.255.254"
      }
    ],
    "pbrType": "SRCADDRESS",
    "srcZones": [
      "manage"
    ],
    "dstIps": [],
    "services": [
      "any"
    ],
    "name": "test",
    "schedule": "全天",
    "description": "create test"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3.5.3.4 添加多条IPv4策略路由信息

功能说明：通过此接口，添加多条IPv4策略路由表项

1) API请求数据

POST

https://{host}/api/batch/v1/namespaces/@namespace/pbrs/ipv4

2) 输入参数

• URL参数

参数	类型	是否必选	描述
_where	string	否	指定位置, 用于移动/插入 枚举值:[top:顶部 bottom:底部 before:在目标位置之前 after:在目标位置之后]
_dest	string	否	目标位置关键字对应的值 最大长度:100

_key	string	否	目标位置关键字 枚举值:[position:指定目标位置关键字为 position name:指定目标位置关键字为 name]
------	--------	---	---

• Body参数

参数	类型	是否必选	描述
obj	object[]	是	IPv4策略路由表项列表
obj.name	string	是	策略路由名称 最小长度:1 最大长度:95
obj.enable	bool	否	启禁用开关 默认值:true
obj.description	string	否	描述字段 最小长度:0 最大长度:95
obj.schedule	string	否	时间计划名称
obj.srcZones	string[]	否	关联区域列表
obj.srcIpGroups	string[]	否	源地址引用IP组列表
obj.dstIpGroups	string[]	否	目的地址引用IP组列表
obj.dstIsp	string[]	否	目的地址关联ISP列表
obj.dstCountryArea	string[]	否	目的地址关联国家/地区列表
obj.services	string[]	否	服务对象列表
obj.applications	string[]	否	关联应用识别列表
obj.lbMethod	string	否	多线路模式下，接口选择策略 枚举值:[POLL:轮询 BANDWIDTH:带宽比例 WEIGHTED_MINIMUM_FLOW:加权最小流量 PRIORITY_FRONT_LINE:优先使用前面线路]
obj.tableId	uint32	否	策略路由表ID，默认为0，当前版本不可修改
obj.status	uint8	否	策略路由的状态值，状态字段不可设置或修改
obj.position	int32	否	策略的位置，状态字段不可设置或修改
obj.pbrType	string	是	策略路由类型 枚举值:[SRCADDRESS:源地址策略路由 MULT_LINE:多线路负载策略路由]
obj.outif	object[]	否	策略路由出接口列表
obj.outif.outifName	string	否	出接口名称

obj.outif.gateway	string	否	下一跳ip地址
obj.outif.linkdt	string	否	引用链路探测对象名称

请求样例:

```
新增多条ipv4策略路由
POST https://192.168.1.1/api/batch/v1/namespaces/public/pbrs/ipv4
[
  {
    "enable": true,
    "applications": [
      "全部"
    ],
    "srcIpgroups": [
      "全部"
    ],
    "outif": [
      {
        "outifName": "eth0",
        "gateway": "10.155.255.254"
      }
    ],
    "dstIpgroups": [
      "全部"
    ],
    "srcZones": [
      "manage"
    ],
    "pbrType": "SRCADDRESS",
    "services": [
      "any"
    ],
    "schedule": "全天",
    "name": "test",
    "description": "create test"
  },
  {
    "enable": true,
    "applications": [
      "全部"
    ],
    "srcIpgroups": [
      "全部"
    ],
    "srcZones": [
      "manage"
    ],
    "pbrType": "SRCADDRESS",
    "services": [
      "any"
    ],
    "name": "testother",
    "dstIps": [
      "电信"
    ],
    "description": "create testother"
  }
]
```

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	IPv4策略路由表项列表
data.uuid	string	策略路由唯一资源ID

data.name	string	策略路由名称 最小长度:1 最大长度:95
data.enable	bool	启禁用开关
data.description	string	描述字段 最小长度:0 最大长度:95
data.schedule	string	时间计划名称
data.srcZones	string[]	关联区域列表
data.srcIpGroups	string[]	源地址引用IP组列表
data.dstIpGroups	string[]	目的地址引用IP组列表
data.dstIsp	string[]	目的地址关联ISP列表
data.dstCountryArea	string[]	目的地址关联国家/地区列表
data.services	string[]	服务对象列表
data.applications	string[]	关联应用识别列表
data.lbMethod	string	多线路模式下，接口选择策略 枚举值: POLL:轮询 BANDWIDTH:带宽比例 WEIGHTED_MINIMUM_FLOW:加权最小流量 PRIORITY_FRONT_LINE:优先使用前面线路]
data.tableId	uint32	策略路由表ID，默认为0，当前版本不可修改
data.status	uint8	策略路由的状态值，状态字段不可设置或修改
data.position	int32	策略的位置，状态字段不可设置或修改
data.pbrType	string	策略路由类型 枚举值: SRCADDRESS:源地址策略路由 MULT_LINE:多线路负载策略路由]
data.outif	object[]	策略路由出接口列表
data.outif.outifName	string	出接口名称
data.outif.gateway	string	下一跳ip地址
data.outif.linkdt	string	引用链路探测对象名称

请求成功样例：

```
{
  "data": [
    {
      "enable": true,
      "applications": [
        "全部"
      ],
      "srcIpGroups": [
        "全部"
      ],
      "outif": [
        {
          "outifName": "eth0",
          "gateway": "10.155.255.254"
        }
      ]
    }
  ]
}
```

```
    },
    "dstIpGroups": [
      "全部"
    ],
    "srcZones": [
      "manage"
    ],
    "pbrType": "SRCADDRESS",
    "schedule": "全天",
    "services": [
      "any"
    ],
    "uuid": "FBB3947A93714437AECB21B7ABA014DF",
    "name": "test",
    "description": "create test"
  },
  {
    "enable": true,
    "applications": [
      "全部"
    ],
    "srcIpGroups": [
      "全部"
    ],
    "srcZones": [
      "manage"
    ],
    "pbrType": "SRCADDRESS",
    "name": "testother",
    "services": [
      "any"
    ],
    "uuid": "1B853B8C70754BA6984AF24B14CAB9DB",
    "dstIp": [
      "电信"
    ],
    "description": "create testother"
  }
],
"code": 0,
"message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3.5.3.5 删除指定的IPv4策略路由信息

功能说明：通过此接口，删除单条IPv4策略路由表项

1) API请求数据

DELETE

https://{host}/api/v1/namespaces/@namespace/pbrs/ipv4/@name

2) 输入参数

请求样例：

删除一条ipv4策略路由，删除名为testother的策略路由

DELETE https://192.168.1.1/api/v1/namespaces/public/pbrs/ipv4/testother

3) 输出参数

• API返回数据

参数	类型	描述
----	----	----

code	int	错误码
message	string	错误信息
data	object	策略路由属性模板
data.uuid	string	策略路由唯一资源ID
data.name	string	策略路由名称 最小长度:1 最大长度:95
data.enable	bool	启禁用开关
data.description	string	描述字段 最小长度:0 最大长度:95
data.schedule	string	时间计划名称
data.srcZones	string[]	关联区域列表
data.srcIpGroups	string[]	源地址引用IP组列表
data.dstIpGroups	string[]	目的地址引用IP组列表
data.dstIsp	string[]	目的地址关联ISP列表
data.dstCountryArea	string[]	目的地址关联国家/地区列表
data.services	string[]	服务对象列表
data.applications	string[]	关联应用识别列表
data.lbMethod	string	多线路模式下，接口选择策略 枚举值: POLL:轮询 BANDWIDTH:带宽比例 WEIGHTED_MINIMUM_FLOW:加权最小流量 PRIORITY_FRONT_LINE:优先使用前面线路]
data.tableId	uint32	策略路由表ID，默认为0，当前版本不可修改
data.status	uint8	策略路由的状态值，状态字段不可设置或修改
data.position	int32	策略的位置，状态字段不可设置或修改
data.pbrType	string	策略路由类型 枚举值: SRCADDRESS:源地址策略路由 MULT_LINE:多线路负载策略路由]
data.outif	object[]	策略路由出接口列表
data.outif.outifName	string	出接口名称
data.outif.gateway	string	下一跳ip地址
data.outif.linkdt	string	引用链路探测对象名称

请求成功样例：

```
{
  "data": {
    "enable": true,
    "dstIsp": [
      "电信"
    ],
  },
}
```

```
    "pbrType": "SRCADDRESS",
    "srcZones": [
      "manage"
    ],
    "name": "testother",
    "uuid": "1B853B8C70754BA6984AF24B14CAB9DB",
    "schedule": "全天",
    "description": "put testother"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3.5.3.6 删除多条指定的IPv4策略路由信息

功能说明: 通过此接口, 删除多条IPv4策略路由表项

1) API请求数据

POST

https://{host}/api/batch/v1/namespaces/@namespace/pbrs/ipv4?_method=delete

2) 输入参数

• Body参数

参数	类型	是否必选	描述
objs	object[]	是	IPv4策略路由表项列表
objs.name	string	是	策略路由名称 最小长度:1 最大长度:95
objs.enable	bool	否	启禁用开关 默认值:true
objs.description	string	否	描述字段 最小长度:0 最大长度:95
objs.schedule	string	否	时间计划名称 默认值:true
objs.srcZones	string[]	否	关联区域列表
objs.srcIpGroups	string[]	否	源地址引用IP组列表
objs.dstIpGroups	string[]	否	目的地址引用IP组列表
objs.dstIsp	string[]	否	目的地址关联ISP列表
objs.dstCountryArea	string[]	否	目的地址关联国家/地区列表
objs.services	string[]	否	服务对象列表
objs.applications	string[]	否	关联应用识别列表

objs.lbMethod	string	否	多线路模式下，接口选择策略 枚举值:[POLL:轮询 BANDWIDTH:带宽比例 WEIGHTED_MINIMUM_FLOW:加权最小流量 PRIORITY_FRONT_LINE:优先使用前面线路]
objs.tableId	uint32	否	策略路由表ID，默认为0，当前版本不可修改
objs.status	uint8	否	策略路由的状态值，状态字段不可设置或修改
objs.position	int32	否	策略的位置，状态字段不可设置或修改
objs.pbrType	string	是	策略路由类型 枚举值:[SRCADDRESS:源地址策略路由 MULT_LINE:多线路负载策略路由]
objs.outif	object[]	否	策略路由出接口列表
objs.outif.outifName	string	否	出接口名称
objs.outif.gateway	string	否	下一跳ip地址
objs.outif.linkdt	string	否	引用链路探测对象名称

请求样例：

删除多条ipv4策略路由，删除名为test和testother的策略路由

```
POST https://192.168.1.1/api/batch/v1/namespaces/public/pbrs/ipv4?_method=DELETE
[
  {
    "name": "test"
  },
  {
    "name": "testother"
  }
]
```

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	IPv4策略路由表项列表
data.uuid	string	策略路由唯一资源ID
data.name	string	策略路由名称 最小长度:1 最大长度:95
data.enable	bool	启禁用开关

data.description	string	描述字段 最小长度:0 最大长度:95
data.schedule	string	时间计划名称
data.srcZones	string[]	关联区域列表
data.srcIpGroups	string[]	源地址引用IP组列表
data.dstIpGroups	string[]	目的地址引用IP组列表
data.dstIsp	string[]	目的地址关联ISP列表
data.dstCountryArea	string[]	目的地址关联国家/地区列表
data.services	string[]	服务对象列表
data.applications	string[]	关联应用识别列表
data.lbMethod	string	多线路模式下，接口选择策略 枚举值:[POLL:轮询 BANDWIDTH:带宽比例 WEIGHTED_MINIMUM_FLOW:加权最小流量 PRIORITY_FRONT_LINE:优先使用前面线路]
data.tableId	uint32	策略路由表ID，默认为0，当前版本不可修改
data.status	uint8	策略路由的状态值，状态字段不可设置或修改
data.position	int32	策略的位置，状态字段不可设置或修改
data.pbrType	string	策略路由类型 枚举值:[SRCADDRESS:源地址策略路由 MULT_LINE:多线路负载策略路由]
data.outif	object[]	策略路由出接口列表
data.outif.outifName	string	出接口名称
data.outif.gateway	string	下一跳ip地址
data.outif.linkdt	string	引用链路探测对象名称

请求成功样例：

```
{
  "data": [
    {
      "enable": false,
      "applications": [
        "全部"
      ],
      "srcIpGroups": [
        "全部"
      ],
      "outif": [
        {
          "outifName": "eth0",
          "gateway": "10.155.255.254"
        }
      ],
      "dstIpGroups": [
        "全部"
      ],
      "srcZones": [
        "manage"
      ]
    }
  ]
}
```



```
    ],
    "pbrType": "SRCADDRESS",
    "schedule": "全天",
    "services": [
      "any"
    ],
    },
    "uuid": "FBB3947A93714437AECB21B7ABA014DF",
    "name": "test",
    "description": "create test"
  },
  {
    "enable": true,
    "dstIsp": [
      "电信"
    ],
    "pbrType": "SRCADDRESS",
    "srcZones": [
      "manage"
    ],
    "name": "testother",
    "uuid": "1B853B8C70754BA6984AF24B14CAB9DB",
    "schedule": "全天",
    "description": "put testother"
  }
],
"code": 0,
"message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3.5.3.7 全量修改指定的IPv4策略路由信息

功能说明：通过此接口，全量修改单条IPv4策略路由表项

1) API请求数据

PUT

https://{host}/api/v1/namespaces/@namespace/pbrs/ipv4/@name

2) 输入参数

- Body参数

参数	类型	是否必选	描述
obj	object	是	策略路由属性模板
obj.name	string	是	策略路由名称 最小长度:1 最大长度:95
obj.enable	bool	否	启禁用开关 默认值:true
obj.description	string	否	描述字段 最小长度:0 最大长度:95
obj.schedule	string	否	时间计划名称
obj.srcZones	string[]	否	关联区域列表
obj.srcIpGroups	string[]	否	源地址引用IP组列表
obj.dstIpGroups	string[]	否	目的地址引用IP组列表

obj.dstIsp	string[]	否	目的地址关联ISP列表
obj.dstCountryArea	string[]	否	目的地址关联国家/地区列表
obj.services	string[]	否	服务对象列表
obj.applications	string[]	否	关联应用识别列表
obj.lbMethod	string	否	多线路模式下，接口选择策略 枚举值:[POLL:轮询 BANDWIDTH:带宽比例 WEIGHTED_MINIMUM_FLOW:加权最小流量 PRIORITY_FRONT_LINE:优先使用前面线路]
obj.tableId	uint32	否	策略路由表ID，默认为0，当前版本不可修改
obj.status	uint8	否	策略路由的状态值，状态字段不可设置或修改
obj.position	int32	否	策略的位置，状态字段不可设置或修改
obj.pbrType	string	是	策略路由类型 枚举值:[SRCADDRESS:源地址策略路由 MULT_LINE:多线路负载策略路由]
obj.outif	object[]	否	策略路由出接口列表
obj.outif.outifName	string	否	出接口名称
obj.outif.gateway	string	否	下一跳ip地址
obj.outif.linkdt	string	否	引用链路探测对象名称

请求样例：

全量更新一条ipv4策略路由，更新名为testother的策略路由，修改描述为put testother，禁用该策略，路由类型不变，时间计划引用全天，源区域引用manage，其余全部默认

```
PUT https://192.168.1.1/api/v1/namespaces/public/pbrs/ipv4/testother
{
  "enable": false,
  "pbrType": "SRCADDRESS",
  "srcZones": [
    "manage"
  ],
  "name": "testother",
  "schedule": "全天",
  "description": "put testother"
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	策略路由属性模板

data.uuid	string	策略路由唯一资源ID
data.name	string	策略路由名称 最小长度:1 最大长度:95
data.enable	bool	启禁用开关
data.description	string	描述字段 最小长度:0 最大长度:95
data.schedule	string	时间计划名称
data.srcZones	string[]	关联区域列表
data.srcIpGroups	string[]	源地址引用IP组列表
data.dstIpGroups	string[]	目的地址引用IP组列表
data.dstIsp	string[]	目的地址关联ISP列表
data.dstCountryArea	string[]	目的地址关联国家/地区列表
data.services	string[]	服务对象列表
data.applications	string[]	关联应用识别列表
data.lbMethod	string	多线路模式下，接口选择策略 枚举值: POLL:轮询 BANDWIDTH:带宽比例 WEIGHTED_MINIMUM_FLOW:加权最小流量 PRIORITY_FRONT_LINE:优先使用前面线路]
data.tableId	uint32	策略路由表ID，默认为0，当前版本不可修改
data.status	uint8	策略路由的状态值，状态字段不可设置或修改
data.position	int32	策略的位置，状态字段不可设置或修改
data.pbrType	string	策略路由类型 枚举值: SRCADDRESS:源地址策略路由 MULT_LINE:多线路负载策略路由]
data.outif	object[]	策略路由出接口列表
data.outif.outifName	string	出接口名称
data.outif.gateway	string	下一跳ip地址
data.outif.linkdt	string	引用链路探测对象名称

请求成功样例：

```
{
  "data": {
    "enable": false,
    "pbrType": "SRCADDRESS",
    "srcZones": [
      "manage"
    ],
    "name": "testother",
    "uuid": "1B853B8C70754BA6984AF24B14CAB9DB",
    "schedule": "全天",
    "description": "put testother"
  },
}
```

```

    "code": 0,
    "message": "成功"
  }

```

请求失败样例:

```

{
  "data": "",
  "code": 22,
  "message": "无效参数"
}

```

3.5.3.8 增量修改指定IPv4策略路由信息

功能说明: 通过此接口, 增量修改单条IPv4策略路由表项

1) API请求数据

PATCH <https://{host}/api/v1/namespaces/@namespace/pbrs/ipv4/@name>

2) 输入参数

• URL参数

参数	类型	是否必选	描述
_arrayop	string	否	对象属性是数组的操作控制 枚举值:[add:添加对应的数组元素 remove:删除对应的数组元素]
_key	string	否	目标位置关键字 枚举值:[position:指定目标位置关键字为 position name:指定目标位置关键字为 name]
_dest	string	否	目标位置关键字对应的值 最大长度:100
_where	string	否	指定位置, 用于移动/插入 枚举值:[top:顶部 bottom:底部 before:在目标位置之前 after:在目标位置之后]

• Body参数

参数	类型	是否必选	描述
obj	object	是	策略路由属性模板
obj.name	string	是	策略路由名称 最小长度:1 最大长度:95
obj.enable	bool	否	启禁用开关 默认值:true
obj.description	string	否	描述字段 最小长度:0 最大长度:95

obj.schedule	string	否	时间计划名称
obj.srcZones	string[]	否	关联区域列表
obj.srcIpGroups	string[]	否	源地址引用IP组列表
obj.dstIpGroups	string[]	否	目的地址引用IP组列表
obj.dstIsp	string[]	否	目的地址关联ISP列表
obj.dstCountryArea	string[]	否	目的地址关联国家/地区列表
obj.services	string[]	否	服务对象列表
obj.applications	string[]	否	关联应用识别列表
obj.lbMethod	string	否	多线路模式下，接口选择策略 枚举值:[POLL:轮询 BANDWIDTH:带宽比例 WEIGHTED_MINIMUM_FLOW:加权最小流量 PRIORITY_FRONT_LINE:优先使用前面线路]
obj.tableId	uint32	否	策略路由表ID，默认为0，当前版本不可修改
obj.status	uint8	否	策略路由的状态值，状态字段不可设置或修改
obj.position	int32	否	策略的位置，状态字段不可设置或修改
obj.pbrType	string	是	策略路由类型 枚举值:[SRCADDRESS:源地址策略路由 MULT_LINE:多线路负载策略路由]
obj.outif	object[]	否	策略路由出接口列表
obj.outif.outifName	string	否	出接口名称
obj.outif.gateway	string	否	下一跳ip地址
obj.outif.linkdt	string	否	引用链路探测对象名称

请求样例：

修改单条ipv4策略路由，修改名为testother的策略路由为启用，其余不改

```
PATCH https://192.168.1.1/api/v1/namespaces/public/pbrs/ipv4/testother
{
  "enable": true
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	策略路由属性模板

data.uuid	string	策略路由唯一资源ID
data.name	string	策略路由名称 最小长度:1 最大长度:95
data.enable	bool	启禁用开关
data.description	string	描述字段 最小长度:0 最大长度:95
data.schedule	string	时间计划名称
data.srcZones	string[]	关联区域列表
data.srcIpGroups	string[]	源地址引用IP组列表
data.dstIpGroups	string[]	目的地址引用IP组列表
data.dstIsp	string[]	目的地址关联ISP列表
data.dstCountryArea	string[]	目的地址关联国家/地区列表
data.services	string[]	服务对象列表
data.applications	string[]	关联应用识别列表
data.lbMethod	string	多线路模式下，接口选择策略 枚举值:[POLL:轮询 BANDWIDTH:带宽比例 WEIGHTED_MINIMUM_FLOW:加权最小流量 PRIORITY_FRONT_LINE:优先使用前面线路]
data.tableId	uint32	策略路由表ID，默认为0，当前版本不可修改
data.status	uint8	策略路由的状态值，状态字段不可设置或修改
data.position	int32	策略的位置，状态字段不可设置或修改
data.pbrType	string	策略路由类型 枚举值:[SRCADDRESS:源地址策略路由 MULT_LINE:多线路负载策略路由]
data.outif	object[]	策略路由出接口列表
data.outif.outifName	string	出接口名称
data.outif.gateway	string	下一跳ip地址
data.outif.linkdt	string	引用链路探测对象名称

请求成功样例：

```
{
  "data": {
    "enable": true,
    "pbrType": "SRCADDRESS",
    "srcZones": [
      "manage"
    ],
    "name": "testother",
    "uuid": "1B853B8C70754BA6984AF24B14CAB9DB",
    "schedule": "全天",
    "description": "put testother"
  },
}
```

```

    "code": 0,
    "message": "成功"
  }

```

请求失败样例:

```

{
  "data": "",
  "code": 22,
  "message": "无效参数"
}

```

3.5.3.9 增量修改多条IPv4策略路由信息

功能说明: 通过此接口, 增量修改多条IPv4策略路由表项

1) API请求数据

PATCH <https://{host}/api/batch/v1/namespaces/@namespace/pbrs/ipv4>

2) 输入参数

• URL参数

参数	类型	是否必选	描述
_arrayop	string	否	对象属性是数组的操作控制 枚举值:[add:添加对应的数组元素 remove:删除对应的数组元素]
_key	string	否	目标位置关键字 枚举值:[position:指定目标位置关键字为 position name:指定目标位置关键字为 name]
_dest	string	否	目标位置关键字对应的值 最大长度:100
_where	string	否	指定位置, 用于移动/插入 枚举值:[top:顶部 bottom:底部 before:在目标位置之前 after:在目标位置之后]

• Body参数

参数	类型	是否必选	描述
obj	object[]	是	IPv4策略路由表项列表
obj.name	string	是	策略路由名称 最小长度:1 最大长度:95
obj.enable	bool	否	启禁用开关 默认值:true
obj.description	string	否	描述字段 最小长度:0 最大长度:95

obj.schedule	string	否	时间计划名称
obj.srcZones	string[]	否	关联区域列表
obj.srcIpGroups	string[]	否	源地址引用IP组列表
obj.dstIpGroups	string[]	否	目的地址引用IP组列表
obj.dstIsp	string[]	否	目的地址关联ISP列表
obj.dstCountryArea	string[]	否	目的地址关联国家/地区列表
obj.services	string[]	否	服务对象列表
obj.applications	string[]	否	关联应用识别列表
obj.lbMethod	string	否	多线路模式下，接口选择策略 枚举值:[POLL:轮询 BANDWIDTH:带宽比例 WEIGHTED_MINIMUM_FLOW:加权最小流量 PRIORITY_FRONT_LINE:优先使用前面线路]
obj.tableId	uint32	否	策略路由表ID，默认为0，当前版本不可修改
obj.status	uint8	否	策略路由的状态值，状态字段不可设置或修改
obj.position	int32	否	策略的位置，状态字段不可设置或修改
obj.pbrType	string	是	策略路由类型 枚举值:[SRCADDRESS:源地址策略路由 MULT_LINE:多线路负载策略路由]
obj.outif	object[]	否	策略路由出接口列表
obj.outif.outifName	string	否	出接口名称
obj.outif.gateway	string	否	下一跳ip地址
obj.outif.linkdt	string	否	引用链路探测对象名称

请求样例:

```
PATCH https://192.168.1.1/api/batch/v1/namespaces/public/pbrs/ipv4
[
  {
    "name": "test",
    "enable": false
  },
  {
    "name": "testother",
    "dstIsp": [
      "电信"
    ]
  }
]
```


3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	IPv4策略路由表项列表
data.uuid	string	策略路由唯一资源ID
data.name	string	策略路由名称 最小长度:1 最大长度:95
data.enable	bool	启禁用开关
data.description	string	描述字段 最小长度:0 最大长度:95
data.schedule	string	时间计划名称
data.srcZones	string[]	关联区域列表
data.srcIpGroups	string[]	源地址引用IP组列表
data.dstIpGroups	string[]	目的地址引用IP组列表
data.dstIsp	string[]	目的地址关联ISP列表
data.dstCountryArea	string[]	目的地址关联国家/地区列表
data.services	string[]	服务对象列表
data.applications	string[]	关联应用识别列表
data.lbMethod	string	多线路模式下，接口选择策略 枚举值:[POLL:轮询 BANDWIDTH:带宽比例 WEIGHTED_MINIMUM_FLOW:加权最小流量 PRIORITY_FRONT_LINE:优先使用前面线路]
data.tableId	uint32	策略路由表ID，默认为0，当前版本不可修改
data.status	uint8	策略路由的状态值，状态字段不可设置或修改
data.position	int32	策略的位置，状态字段不可设置或修改
data.pbrType	string	策略路由类型 枚举值:[SRCADDRESS:源地址策略路由 MULT_LINE:多线路负载策略路由]
data.outif	object[]	策略路由出接口列表
data.outif.outifName	string	出接口名称
data.outif.gateway	string	下一跳ip地址
data.outif.linkdt	string	引用链路探测对象名称

请求成功样例：

```
{
  "data": [
    {
      "enable": false,
      "applications": [
        "全部"
      ],
      "srcIpGroups": [
        "全部"
      ],
      "outif": [
        {
          "outifName": "eth0",
          "gateway": "10.155.255.254"
        }
      ],
      "dstIpGroups": [
        "全部"
      ],
      "srcZones": [
        "manage"
      ],
      "pbrType": "SRCADDRESS",
      "schedule": "全天",
      "services": [
        "any"
      ],
      "uuid": "FBB3947A93714437AECB21B7ABA014DF",
      "name": "test",
      "description": "create test"
    },
    {
      "enable": true,
      "dstIpGroups": [
        "电信"
      ],
      "pbrType": "SRCADDRESS",
      "srcZones": [
        "manage"
      ],
      "name": "testother",
      "uuid": "1B853B8C70754BA6984AF24B14CAB9DB",
      "schedule": "全天",
      "description": "put testother"
    }
  ],
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3.5.4 策略路由IPv6

3.5.4.1 获取单条IPv6策略路由信息

功能说明: 通过此接口, 获取配置中心单条IPv6策略路由表项

1) API请求数据

GET <https://{host}/api/v1/namespaces/@namespace/pbrs/ipv6/@name>

2) 输入参数

- URL参数

参数	类型	是否必选	描述
_select	string	否	选择字段, 字段用逗号分隔 最大长度:1000

请求样例:

获取单条ipv6策略路由的配置信息, 查询条件: 查看名为test的策略路由

GET https://192.168.1.1/api/v1/namespaces/public/pbrs/ipv6/test

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	策略路由属性模板
data.uuid	string	策略路由唯一资源ID
data.name	string	策略路由名称 最小长度:1 最大长度:95
data.enable	bool	启禁用开关
data.description	string	描述字段 最小长度:0 最大长度:95
data.schedule	string	时间计划名称
data.srcZones	string[]	关联区域列表
data.srcIpGroups	string[]	源地址引用IP组列表
data.dstIpGroups	string[]	目的地址引用IP组列表
data.dstIsp	string[]	目的地址关联ISP列表
data.dstCountryArea	string[]	目的地址关联国家/地区列表
data.services	string[]	服务对象列表
data.applications	string[]	关联应用识别列表
data.lbMethod	string	多线路模式下, 接口选择策略 枚举值: POLL:轮询 BANDWIDTH:带宽比例 WEIGHTED_MINIMUM_FLOW:加权最小流量 PRIORITY_FRONT_LINE:优先使用前面线路]
data.tableId	uint32	策略路由表ID, 默认为0, 当前版本不可修改
data.status	uint8	策略路由的状态值, 状态字段不可设置或修改
data.position	int32	策略的位置, 状态字段不可设置或修改

data.pbrType	string	策略路由类型 枚举值:[SRCADDRESS:源地址策略路由]
data.outif	object[]	策略路由出接口列表
data.outif.outifName	string	出接口名称
data.outif.gateway	string	下一跳ip地址 格式:ip6-address
data.outif.linkdt	string	引用链路探测对象名称

请求成功样例:

```
{
  "data": {
    "enable": true,
    "applications": [
      "全部"
    ],
    "srcIpGroups": [
      "全部"
    ],
    "dstCountryArea": [],
    "dstIpGroups": [
      "全部"
    ],
    "uuid": "FBB3947A93714437AECB21B7ABA014DF",
    "outif": [
      {
        "outifName": "eth0",
        "gateway": "2000::1"
      }
    ],
    "pbrType": "SRCADDRESS",
    "position": 0,
    "srcZones": [
      "manage"
    ],
    "dstIp": [],
    "services": [
      "any"
    ],
    "name": "test",
    "schedule": "全天",
    "description": ""
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3.5.4.2 获取IPv6策略路由列表

功能说明: 通过此接口, 获取配置中心所有IPv6策略路由表项

1) API请求数据

GET

https://{host}/api/v1/namespaces/@namespace/pbrs/ipv6

2) 输入参数

• URL参数

参数	类型	是否必选	描述
_start	int32	否	起始位置, 从 0 开始 最小值:0 最大值:2000000
position	int32	否	拿取指定位置的配置
_search	string	否	模糊搜索关键字 最大长度:95
_select	string	否	选择字段, 字段用逗号分隔 最大长度:1000
movesearch	string	否	移动搜索参数项 最小长度:1 最大长度:95
_order	string	否	指定排序方式 (正序/倒序) 默认值:true 枚举值:[asc:按升序排列 desc:按降序排列]
_length	int32	否	获取最大长度, 限制最大值, 如果数据不够可能返回小于 length 最小值:1 最大值:200 默认值:true
_sortby	string	否	指定排序字段 最大长度:100

请求样例:

获取多条ipv6策略路由的配置信息, 查询条件: 起始位置为0, 最多展示100条

GET https://192.168.1.1/api/v1/namespaces/public/pbrs/ipv6?_start=0&_length=100

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	分页数据
data.totalItems	int32	总共多少项目
data.totalPages	int32	总共多少页
data.pageNumber	int32	当前页码, 从 1 开始
data.pageSize	int32	每页多大
data.itemsOffset	int32	当前条目偏移, 从 0 开始
data.itemLength	int32	数据列表长度

data.items	object[]	有效数据列表
data.items.uuid	string	策略路由唯一资源ID
data.items.name	string	策略路由名称 最小长度:1 最大长度:95
data.items.enable	bool	启禁用开关
data.items.description	string	描述字段 最小长度:0 最大长度:95
data.items.schedule	string	时间计划名称
data.items.srcZones	string[]	关联区域列表
data.items.srcIpGroups	string[]	源地址引用IP组列表
data.items.dstIpGroups	string[]	目的地址引用IP组列表
data.items.dstIsp	string[]	目的地址关联ISP列表
data.items.dstCountryArea	string[]	目的地址关联国家/地区列表
data.items.services	string[]	服务对象列表
data.items.applications	string[]	关联应用识别列表
data.items.lbMethod	string	多线路模式下，接口选择策略 枚举值: POLL:轮询 BANDWIDTH:带宽比例 WEIGHTED_MINIMUM_FLOW:加权最小流量 PRIORITY_FRONT_LINE:优先使用前面线路]
data.items.tableId	uint32	策略路由表ID，默认为0，当前版本不可修改
data.items.status	uint8	策略路由的状态值，状态字段不可设置或修改
data.items.position	int32	策略的位置，状态字段不可设置或修改
data.items.pbrType	string	策略路由类型 枚举值: SRCADDRESS:源地址策略路由]
data.items.outif	object[]	策略路由出接口列表
data.items.outif.outifName	string	出接口名称
data.items.outif.gateway	string	下一跳ip地址 格式:ip6-address
data.items.outif.linkdt	string	引用链路探测对象名称

请求成功样例：

```
{
  "data": {
    "itemsOffset": 0,
    "items": [
      {
        "enable": true,
        "applications": [
```

```
        "全部"
      ],
      "srcIpGroups": [
        "全部"
      ],
      "dstCountryArea": [],
      "dstIpGroups": [
        "全部"
      ],
      "uuid": "FBB3947A93714437AECB21B7ABA014DF",
      "outif": [
        {
          "outifName": "eth0",
          "gateway": "2000::1"
        }
      ],
      "pbrType": "SRCADDRESS",
      "position": 0,
      "srcZones": [
        "manage"
      ],
      "dstIsp": [],
      "services": [
        "any"
      ],
      "name": "test",
      "schedule": "全天",
      "description": ""
    }
  ],
  "totalItems": 1,
  "pageNumber": 1,
  "itemLength": 1,
  "totalPages": 1,
  "pageSize": 100
},
{
  "code": 0,
  "message": "成功"
}
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3.5.4.3 添加一条IPv6策略路由信息

功能说明：通过此接口，添加单条IPv6策略路由表项

1) API请求数据

POST

https://{host}/api/v1/namespaces/@namespace/pbrs/ipv6

2) 输入参数

- URL 参数

参数	类型	是否必选	描述
_where	string	否	指定位置, 用于移动/插入 枚举值:[top:顶部 bottom:底部 before:在目标位置之前 after:在目标位置之后]

_dest	string	否	目标位置关键字对应的值 最大长度:100
_key	string	否	目标位置关键字 枚举值:[position:指定目标位置关键字为 position name:指定目标位置关键字为 name]

• Body参数

参数	类型	是否必选	描述
obj	object	是	策略路由属性模板
obj.name	string	是	策略路由名称 最小长度:1 最大长度:95
obj.enable	bool	否	启禁用开关 默认值:true
obj.description	string	否	描述字段 最小长度:0 最大长度:95
obj.schedule	string	否	时间计划名称
obj.srcZones	string[]	否	关联区域列表
obj.srcIpGroups	string[]	否	源地址引用IP组列表
obj.dstIpGroups	string[]	否	目的地址引用IP组列表
obj.dstIsp	string[]	否	目的地址关联ISP列表
obj.dstCountryArea	string[]	否	目的地址关联国家/地区列表
obj.services	string[]	否	服务对象列表
obj.applications	string[]	否	关联应用识别列表
obj.lbMethod	string	否	多线路模式下，接口选择策略 枚举值:[POLL:轮询 BANDWIDTH:带宽比例 WEIGHTED_MINIMUM_FLOW:加权最小流量 PRIORITY_FRONT_LINE:优先使用前面线路]
obj.tableId	uint32	否	策略路由表ID，默认为0，当前版本不可修改
obj.status	uint8	否	策略路由的状态值，状态字段不可设置或修改
obj.position	int32	否	策略的位置，状态字段不可设置或修改
obj.pbrType	string	是	策略路由类型 枚举值:[SRCADDRESS:源地址策略路由]
obj.outif	object[]	否	策略路由出接口列表

obj.outif.outifName	string	否	出接口名称
obj.outif.gateway	string	否	下一跳ip地址 格式:ip6-address
obj.outif.linkdt	string	否	引用链路探测对象名称

请求样例:

```
新增单条ipv6策略路由
POST https://192.168.1.1/api/v1/namespaces/public/pbrs/ipv6
{
  "enable": true,
  "applications": [
    "全部"
  ],
  "srcIpGroups": [
    "全部"
  ],
  "outif": [
    {
      "outifName": "eth0",
      "gateway": "2000::1"
    }
  ],
  "dstIpGroups": [
    "全部"
  ],
  "srcZones": [
    "manage"
  ],
  "pbrType": "SRCADDRESS",
  "services": [
    "any"
  ],
  "schedule": "全天",
  "name": "test",
  "description": "create test"
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	策略路由属性模板
data.uuid	string	策略路由唯一资源ID
data.name	string	策略路由名称 最小长度:1 最大长度:95
data.enable	bool	启禁用开关
data.description	string	描述字段 最小长度:0 最大长度:95
data.schedule	string	时间计划名称
data.srcZones	string[]	关联区域列表
data.srcIpGroups	string[]	源地址引用IP组列表
data.dstIpGroups	string[]	目的地址引用IP组列表

data.dstIsp	string[]	目的地址关联ISP列表
data.dstCountryArea	string[]	目的地址关联国家/地区列表
data.services	string[]	服务对象列表
data.applications	string[]	关联应用识别列表
data.lbMethod	string	多线路模式下，接口选择策略 枚举值:[POLL:轮询 BANDWIDTH:带宽比例 WEIGHTED_MINIMUM_FLOW:加权最小流量 PRIORITY_FRONT_LINE:优先使用前面线路]
data.tableId	uint32	策略路由表ID，默认为0，当前版本不可修改
data.status	uint8	策略路由的状态值，状态字段不可设置或修改
data.position	int32	策略的位置，状态字段不可设置或修改
data.pbrType	string	策略路由类型 枚举值:[SRCADDRESS:源地址策略路由]
data.outif	object[]	策略路由出接口列表
data.outif.outifName	string	出接口名称
data.outif.gateway	string	下一跳ip地址 格式:ip6-address
data.outif.linkdt	string	引用链路探测对象名称

请求成功样例：

```
{
  "data": {
    "enable": true,
    "applications": [
      "全部"
    ],
    "srcIpGroups": [
      "全部"
    ],
    "dstCountryArea": [],
    "dstIpGroups": [
      "全部"
    ],
    "uuid": "FBB3947A93714437AECB21B7ABA014DF",
    "outif": [
      {
        "outifName": "eth0",
        "gateway": "2000::1"
      }
    ],
    "pbrType": "SRCADDRESS",
    "srcZones": [
      "manage"
    ],
    "dstIsp": [],
    "services": [
      "any"
    ],
    "name": "test",
    "schedule": "全天",
    "description": "create test"
  },
  "code": 0,
  "message": "成功"
}
```

}

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3.5.4.4 添加多条IPv6策略路由信息

功能说明: 通过此接口, 添加多条IPv6策略路由表项

1) API请求数据

POST <https://{host}/api/batch/v1/namespaces/@namespace/pbrs/ipv6>

2) 输入参数

• URL参数

参数	类型	是否必选	描述
_where	string	否	指定位置, 用于移动/插入 枚举值:[top:顶部 bottom:底部 before:在目标位置之前 after:在目标位置之后]
_dest	string	否	目标位置关键字对应的值 最大长度:100
_key	string	否	目标位置关键字 枚举值:[position:指定目标位置关键字为 position name:指定目标位置关键字为 name]

• Body参数

参数	类型	是否必选	描述
obj	object[]	是	IPv6策略路由表项列表
obj.name	string	是	策略路由名称 最小长度:1 最大长度:95
obj.enable	bool	否	启禁用开关 默认值:true
obj.description	string	否	描述字段 最小长度:0 最大长度:95
obj.schedule	string	否	时间计划名称
obj.srcZones	string[]	否	关联区域列表
obj.srcIpGroups	string[]	否	源地址引用IP组列表
obj.dstIpGroups	string[]	否	目的地址引用IP组列表
obj.dstIsp	string[]	否	目的地址关联ISP列表

obj.dstCountryArea	string[]	否	目的地址关联国家/地区列表
obj.services	string[]	否	服务对象列表
obj.applications	string[]	否	关联应用识别列表
obj.lbMethod	string	否	多线路模式下，接口选择策略 枚举值:[POLL:轮询 BANDWIDTH:带宽比例 WEIGHTED_MINIMUM_FLOW:加权最小流量 PRIORITY_FRONT_LINE:优先使用前面线路]
obj.tableId	uint32	否	策略路由表ID，默认为0，当前版本不可修改
obj.status	uint8	否	策略路由的状态值，状态字段不可设置或修改
obj.position	int32	否	策略的位置，状态字段不可设置或修改
obj.pbrType	string	是	策略路由类型 枚举值:[SRCADDRESS:源地址策略路由]
obj.outif	object[]	否	策略路由出接口列表
obj.outif.outifName	string	否	出接口名称
obj.outif.gateway	string	否	下一跳ip地址 格式:ip6-address
obj.outif.linkdt	string	否	引用链路探测对象名称

请求样例：

新增多条ipv6策略路由

POST https://192.168.1.1/api/batch/v1/namespaces/public/pbrs/ipv6

```
[
  {
    "enable": true,
    "applications": [
      "全部"
    ],
    "srcIpGroups": [
      "全部"
    ],
    "outif": [
      {
        "outifName": "eth0",
        "gateway": "2000::1"
      }
    ],
    "dstIpGroups": [
      "全部"
    ],
    "srcZones": [
      "manage"
    ],
    "pbrType": "SRCADDRESS",
    "services": [
      "any"
    ],
    "schedule": "全天",
    "name": "test",
    "description": "create test"
  }
]
```

```
    },
    {
      "enable": true,
      "applications": [
        "全部"
      ],
      "srcIpGroups": [
        "全部"
      ],
      "pbrType": "SRCADDRESS",
      "services": [
        "any"
      ],
      "name": "testother",
      "srcZones": [
        "manage"
      ],
      "description": "create testother"
    }
  ]
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	IPv6策略路由表项列表
data.uuid	string	策略路由唯一资源ID
data.name	string	策略路由名称 最小长度:1 最大长度:95
data.enable	bool	启禁用开关
data.description	string	描述字段 最小长度:0 最大长度:95
data.schedule	string	时间计划名称
data.srcZones	string[]	关联区域列表
data.srcIpGroups	string[]	源地址引用IP组列表
data.dstIpGroups	string[]	目的地址引用IP组列表
data.dstIsp	string[]	目的地址关联ISP列表
data.dstCountryArea	string[]	目的地址关联国家/地区列表
data.services	string[]	服务对象列表
data.applications	string[]	关联应用识别列表
data.lbMethod	string	多线路模式下，接口选择策略 枚举值:[POLL:轮询 BANDWIDTH:带宽比例 WEIGHTED_MINIMUM_FLOW:加权最小流量 PRIORITY_FRONT_LINE:优先使用前面线路]
data.tableId	uint32	策略路由表ID，默认为0，当前版本不可修改

data.status	uint8	策略路由的状态值，状态字段不可设置或修改
data.position	int32	策略的位置，状态字段不可设置或修改
data.pbrType	string	策略路由类型 枚举值:[SRCADDRESS:源地址策略路由]
data.outif	object[]	策略路由出接口列表
data.outif.outifName	string	出接口名称
data.outif.gateway	string	下一跳ip地址 格式:ip6-address
data.outif.linkdt	string	引用链路探测对象名称

请求成功样例：

```
{
  "data": [
    {
      "enable": true,
      "applications": [
        "全部"
      ],
      "srcIpGroups": [
        "全部"
      ],
      "outif": [
        {
          "outifName": "eth0",
          "gateway": "2000::1"
        }
      ],
      "dstIpGroups": [
        "全部"
      ],
      "srcZones": [
        "manage"
      ],
      "pbrType": "SRCADDRESS",
      "schedule": "全天",
      "services": [
        "any"
      ],
      "uuid": "FBB3947A93714437AECB21B7ABA014DF",
      "name": "test",
      "description": "create test"
    },
    {
      "enable": true,
      "applications": [
        "全部"
      ],
      "srcIpGroups": [
        "全部"
      ],
      "srcZones": [
        "manage"
      ],
      "pbrType": "SRCADDRESS",
      "name": "testother",
      "services": [
        "any"
      ],
      "uuid": "1B853B8C70754BA6984AF24B14CAB9DB",
      "schedule": "全天",
      "description": "create testother"
    }
  ],
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3.5.4.5 删除指定的IPv6策略路由信息

功能说明：通过此接口，删除单条IPv6策略路由表项

1) API请求数据

DELETE https://{host}/api/v1/namespaces/@namespace/pbrs/ipv6/@name

2) 输入参数

请求样例：

删除一条ipv6策略路由，删除名为testother的策略路由

DELETE https://192.168.1.1/api/v1/namespaces/public/pbrs/ipv6/testother

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	策略路由属性模板
data.uuid	string	策略路由唯一资源ID
data.name	string	策略路由名称 最小长度:1 最大长度:95
data.enable	bool	启禁用开关
data.description	string	描述字段 最小长度:0 最大长度:95
data.schedule	string	时间计划名称
data.srcZones	string[]	关联区域列表
data.srcIpGroups	string[]	源地址引用IP组列表
data.dstIpGroups	string[]	目的地址引用IP组列表
data.dstIsp	string[]	目的地址关联ISP列表
data.dstCountryArea	string[]	目的地址关联国家/地区列表
data.services	string[]	服务对象列表
data.applications	string[]	关联应用识别列表

data.lbMethod	string	多线路模式下，接口选择策略 枚举值:[POLL:轮询 BANDWIDTH:带宽比例 WEIGHTED_MINIMUM_FLOW:加权最小流量 PRIORITY_FRONT_LINE:优先使用前面线路]
data.tableId	uint32	策略路由表ID，默认为0，当前版本不可修改
data.status	uint8	策略路由的状态值，状态字段不可设置或修改
data.position	int32	策略的位置，状态字段不可设置或修改
data.pbrType	string	策略路由类型 枚举值:[SRCADDRESS:源地址策略路由]
data.outif	object[]	策略路由出接口列表
data.outif.outifName	string	出接口名称
data.outif.gateway	string	下一跳ip地址 格式:ip6-address
data.outif.linkdt	string	引用链路探测对象名称

请求成功样例：

```
{
  "data": {
    "enable": true,
    "uuid": "1B853B8C70754BA6984AF24B14CAB9DB",
    "pbrType": "SRCADDRESS",
    "srcZones": [
      "manage"
    ],
    "name": "testother",
    "dstIpGroups": [
      "全部"
    ],
    "schedule": "全天",
    "description": "put testother"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3.5.4.6 删除指定的多条IPv6策略路由信息

功能说明：通过此接口，删除多条IPv6策略路由表项

1) API请求数据

POST

https://{host}/api/batch/v1/namespaces/@namespace/pbrs/ipv6?_method=delete

2) 输入参数

• Body参数

参数	类型	是否必选	描述
objs	object[]	是	IPv6策略路由表项列表
objs.name	string	是	策略路由名称 最小长度:1 最大长度:95
objs.enable	bool	否	启禁用开关 默认值:true
objs.description	string	否	描述字段 最小长度:0 最大长度:95
objs.schedule	string	否	时间计划名称 默认值:true
objs.srcZones	string[]	否	关联区域列表
objs.srcIpGroups	string[]	否	源地址引用IP组列表
objs.dstIpGroups	string[]	否	目的地址引用IP组列表
objs.dstIsp	string[]	否	目的地址关联ISP列表
objs.dstCountryArea	string[]	否	目的地址关联国家/地区列表
objs.services	string[]	否	服务对象列表
objs.applications	string[]	否	关联应用识别列表
objs.lbMethod	string	否	多线路模式下，接口选择策略 枚举值:[POLL:轮询 BANDWIDTH:带宽比例 WEIGHTED_MINIMUM_FLOW:加权最小流量 PRIORITY_FRONT_LINE:优先使用前面线路]
objs.tableId	uint32	否	策略路由表ID，默认为0，当前版本不可修改
objs.status	uint8	否	策略路由的状态值，状态字段不可设置或修改
objs.position	int32	否	策略的位置，状态字段不可设置或修改
objs.pbrType	string	是	策略路由类型 枚举值:[SRCADDRESS:源地址策略路由]
objs.outif	object[]	否	策略路由出接口列表
objs.outif.outifName	string	否	出接口名称
objs.outif.gateway	string	否	下一跳ip地址 格式:ip6-address
objs.outif.linkdt	string	否	引用链路探测对象名称

请求样例:

```
删除多条ipv6策略路由，删除名为test和testother的策略路由
POST https://192.168.1.1/api/batch/v1/namespaces/public/pbrs/ipv6?_method=DELETE
[
  {
    "name": "test"
  },
  {
    "name": "testother"
  }
]
```

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	IPv6策略路由表项列表
data.uuid	string	策略路由唯一资源ID
data.name	string	策略路由名称 最小长度:1 最大长度:95
data.enable	bool	启禁用开关
data.description	string	描述字段 最小长度:0 最大长度:95
data.schedule	string	时间计划名称
data.srcZones	string[]	关联区域列表
data.srcIpGroups	string[]	源地址引用IP组列表
data.dstIpGroups	string[]	目的地址引用IP组列表
data.dstIsp	string[]	目的地址关联ISP列表
data.dstCountryArea	string[]	目的地址关联国家/地区列表
data.services	string[]	服务对象列表
data.applications	string[]	关联应用识别列表
data.lbMethod	string	多线路模式下，接口选择策略 枚举值: POLL:轮询 BANDWIDTH:带宽比例 WEIGHTED_MINIMUM_FLOW:加权最小流量 PRIORITY_FRONT_LINE:优先使用前面线路]
data.tableId	uint32	策略路由表ID，默认为0，当前版本不可修改
data.status	uint8	策略路由的状态值，状态字段不可设置或修改
data.position	int32	策略的位置，状态字段不可设置或修改

data.pbrType	string	策略路由类型 枚举值:[SRCADDRESS:源地址策略路由]
data.outif	object[]	策略路由出接口列表
data.outif.outifName	string	出接口名称
data.outif.gateway	string	下一跳ip地址 格式:ip6-address
data.outif.linkdt	string	引用链路探测对象名称

请求成功样例:

```
{
  "data": [
    {
      "enable": false,
      "applications": [
        "全部"
      ],
      "srcIpGroups": [
        "全部"
      ],
      "outif": [
        {
          "outifName": "eth0",
          "gateway": "2000::1"
        }
      ],
      "dstIpGroups": [
        "全部"
      ],
      "srcZones": [
        "manage"
      ],
      "pbrType": "SRCADDRESS",
      "schedule": "全天",
      "services": [
        "any"
      ],
      "uuid": "FBB3947A93714437AECB21B7ABA014DF",
      "name": "test",
      "description": "create test"
    },
    {
      "enable": true,
      "uuid": "4B853B8C70754BA6984AF24B14CAB9DB",
      "pbrType": "SRCADDRESS",
      "srcZones": [
        "manage"
      ],
      "name": "testother",
      "dstIpGroups": [
        "全部"
      ],
      "schedule": "全天",
      "description": "put testother"
    }
  ],
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3.5.4.7 全量修改指定的IPv6策略路由信息

功能说明：通过此接口，全量修改单条IPv6策略路由表项

1) API请求数据

PUT <https://{host}/api/v1/namespaces/@namespace/pbrs/ipv6/@name>

2) 输入参数

• Body参数

参数	类型	是否必选	描述
obj	object	是	策略路由属性模板
obj.name	string	是	策略路由名称 最小长度:1 最大长度:95
obj.enable	bool	否	启禁用开关 默认值:true
obj.description	string	否	描述字段 最小长度:0 最大长度:95
obj.schedule	string	否	时间计划名称
obj.srcZones	string[]	否	关联区域列表
obj.srcIpGroups	string[]	否	源地址引用IP组列表
obj.dstIpGroups	string[]	否	目的地址引用IP组列表
obj.dstIsp	string[]	否	目的地址关联ISP列表
obj.dstCountryArea	string[]	否	目的地址关联国家/地区列表
obj.services	string[]	否	服务对象列表
obj.applications	string[]	否	关联应用识别列表
obj.lbMethod	string	否	多线路模式下，接口选择策略 枚举值:[POLL:轮询 BANDWIDTH:带宽比例 WEIGHTED_MINIMUM_FLOW:加权最小流量 PRIORITY_FRONT_LINE:优先使用前面线路]
obj.tableId	uint32	否	策略路由表ID，默认为0，当前版本不可修改
obj.status	uint8	否	策略路由的状态值，状态字段不可设置或修改
obj.position	int32	否	策略的位置，状态字段不可设置或修改
obj.pbrType	string	是	策略路由类型 枚举值:[SRCADDRESS:源地址策略路由]
obj.outif	object[]	否	策略路由出接口列表

obj.outif.outifName	string	否	出接口名称
obj.outif.gateway	string	否	下一跳ip地址 格式:ip6-address
obj.outif.linkdt	string	否	引用链路探测对象名称

请求样例:

全量更新一条ipv6策略路由，更新名为testother的策略路由，修改描述为put testother，禁用该策略，路由类型不变，时间计划引用全天，源区域引用manage，其余全部默认

```
PUT https://192.168.1.1/api/v1/namespaces/public/pbrs/ipv6/testother
{
  "enable": false,
  "pbrType": "SRCADDRESS",
  "srcZones": [
    "manage"
  ],
  "name": "testother",
  "schedule": "全天",
  "description": "put testother"
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	策略路由属性模板
data.uuid	string	策略路由唯一资源ID
data.name	string	策略路由名称 最小长度:1 最大长度:95
data.enable	bool	启禁用开关
data.description	string	描述字段 最小长度:0 最大长度:95
data.schedule	string	时间计划名称
data.srcZones	string[]	关联区域列表
data.srcIpGroups	string[]	源地址引用IP组列表
data.dstIpGroups	string[]	目的地址引用IP组列表
data.dstIsp	string[]	目的地址关联ISP列表
data.dstCountryArea	string[]	目的地址关联国家/地区列表
data.services	string[]	服务对象列表
data.applications	string[]	关联应用识别列表

data.lbMethod	string	多线路模式下，接口选择策略 枚举值:[POLL:轮询 BANDWIDTH:带宽比例 WEIGHTED_MINIMUM_FLOW:加权最小流量 PRIORITY_FRONT_LINE:优先使用前面线路]
data.tableId	uint32	策略路由表ID，默认为0，当前版本不可修改
data.status	uint8	策略路由的状态值，状态字段不可设置或修改
data.position	int32	策略的位置，状态字段不可设置或修改
data.pbrType	string	策略路由类型 枚举值:[SRCADDRESS:源地址策略路由]
data.outif	object[]	策略路由出接口列表
data.outif.outifName	string	出接口名称
data.outif.gateway	string	下一跳ip地址 格式:ip6-address
data.outif.linkdt	string	引用链路探测对象名称

请求成功样例：

```
{
  "data": {
    "enable": false,
    "pbrType": "SRCADDRESS",
    "srcZones": [
      "manage"
    ],
    "name": "testother",
    "uuid": "1B853B8C70754BA6984AF24B14CAB9DB",
    "schedule": "全天",
    "description": "put testother"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3.5.4.8 增量修改指定IPv6策略路由信息

功能说明：通过此接口，增量修改单条IPv6策略路由表项

1) API请求数据

PATCH

https://{host}/api/v1/namespaces/@namespace/pbrs/ipv6/@name

2) 输入参数

- URL参数

参数	类型	是否必选	描述
----	----	------	----

_arrayop	string	否	对象属性是数组的操作控制 枚举值:[add:添加对应的数组元素 remove:删除对应的数组元素]
_key	string	否	目标位置关键字 枚举值:[position:指定目标位置关键字为 position name:指定目标位置关键字为 name]
_dest	string	否	目标位置关键字对应的值 最大长度:100
_where	string	否	指定位置, 用于移动/插入 枚举值:[top:顶部 bottom:底部 before:在目标位置之前 after:在目标位置之后]

• Body参数

参数	类型	是否必选	描述
obj	object	是	策略路由属性模板
obj.name	string	是	策略路由名称 最小长度:1 最大长度:95
obj.enable	bool	否	启禁用开关 默认值:true
obj.description	string	否	描述字段 最小长度:0 最大长度:95
obj.schedule	string	否	时间计划名称
obj.srcZones	string[]	否	关联区域列表
obj.srcIpGroups	string[]	否	源地址引用IP组列表
obj.dstIpGroups	string[]	否	目的地址引用IP组列表
obj.dstIsp	string[]	否	目的地址关联ISP列表
obj.dstCountryArea	string[]	否	目的地址关联国家/地区列表
obj.services	string[]	否	服务对象列表
obj.applications	string[]	否	关联应用识别列表
obj.lbMethod	string	否	多线路模式下, 接口选择策略 枚举值:[POLL:轮询 BANDWIDTH:带宽比例 WEIGHTED_MINIMUM_FLOW:加权最小流量 PRIORITY_FRONT_LINE:优先使用前面线路]

obj.tableId	uint32	否	策略路由表ID，默认为0，当前版本不可修改
obj.status	uint8	否	策略路由的状态值，状态字段不可设置或修改
obj.position	int32	否	策略的位置，状态字段不可设置或修改
obj.pbrType	string	是	策略路由类型 枚举值:[SRCADDRESS:源地址策略路由]
obj.outif	object[]	否	策略路由出接口列表
obj.outif.outifName	string	否	出接口名称
obj.outif.gateway	string	否	下一跳ip地址 格式:ip6-address
obj.outif.linkdt	string	否	引用链路探测对象名称

请求样例：

修改单条ipv6策略路由，修改名为testother的策略路由为启用，其余不改

```
PATCH https://192.168.1.1/api/v1/namespaces/public/pbrs/ipv6/testother
{
  "enable": true
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	策略路由属性模板
data.uuid	string	策略路由唯一资源ID
data.name	string	策略路由名称 最小长度:1 最大长度:95
data.enable	bool	启禁用开关
data.description	string	描述字段 最小长度:0 最大长度:95
data.schedule	string	时间计划名称
data.srcZones	string[]	关联区域列表
data.srcIpGroups	string[]	源地址引用IP组列表
data.dstIpGroups	string[]	目的地址引用IP组列表
data.dstIsp	string[]	目的地址关联ISP列表
data.dstCountryArea	string[]	目的地址关联国家/地区列表
data.services	string[]	服务对象列表
data.applications	string[]	关联应用识别列表

data.lbMethod	string	多线路模式下，接口选择策略 枚举值:[POLL:轮询 BANDWIDTH:带宽比例 WEIGHTED_MINIMUM_FLOW:加权最小流量 PRIORITY_FRONT_LINE:优先使用前面线路]
data.tableId	uint32	策略路由表ID，默认为0，当前版本不可修改
data.status	uint8	策略路由的状态值，状态字段不可设置或修改
data.position	int32	策略的位置，状态字段不可设置或修改
data.pbrType	string	策略路由类型 枚举值:[SRCADDRESS:源地址策略路由]
data.outif	object[]	策略路由出接口列表
data.outif.outifName	string	出接口名称
data.outif.gateway	string	下一跳ip地址 格式:ip6-address
data.outif.linkdt	string	引用链路探测对象名称

请求成功样例：

```
{
  "data": {
    "enable": true,
    "pbrType": "SRCADDRESS",
    "srcZones": [
      "manage"
    ],
    "name": "testother",
    "uuid": "1B853B8C70754BA6984AF24B14CAB9DB",
    "schedule": "全天",
    "description": "put testother"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3.5.4.9 增量修改多条IPv6策略路由信息

功能说明：通过此接口，增量修改多条IPv6策略路由表项

1) API请求数据

PATCH

https://{host}/api/batch/v1/namespaces/@namespace/pbrs/ipv6

2) 输入参数

- URL参数

参数	类型	是否必选	描述
----	----	------	----

_arrayop	string	否	对象属性是数组的操作控制 枚举值:[add:添加对应的数组元素 remove:删除对应的数组元素]
_key	string	否	目标位置关键字 枚举值:[position:指定目标位置关键字为 position name:指定目标位置关键字为 name]
_dest	string	否	目标位置关键字对应的值 最大长度:100
_where	string	否	指定位置, 用于移动/插入 枚举值:[top:顶部 bottom:底部 before:在目标位置之前 after:在目标位置之后]

• Body参数

参数	类型	是否必选	描述
obj	object[]	是	IPv6策略路由表项列表
obj.name	string	是	策略路由名称 最小长度:1 最大长度:95
obj.enable	bool	否	启禁用开关 默认值:true
obj.description	string	否	描述字段 最小长度:0 最大长度:95
obj.schedule	string	否	时间计划名称
obj.srcZones	string[]	否	关联区域列表
obj.srcIpGroups	string[]	否	源地址引用IP组列表
obj.dstIpGroups	string[]	否	目的地址引用IP组列表
obj.dstIsp	string[]	否	目的地址关联ISP列表
obj.dstCountryArea	string[]	否	目的地址关联国家/地区列表
obj.services	string[]	否	服务对象列表
obj.applications	string[]	否	关联应用识别列表
obj.lbMethod	string	否	多线路模式下, 接口选择策略 枚举值:[POLL:轮询 BANDWIDTH:带宽比例 WEIGHTED_MINIMUM_FLOW:加权最小流量 PRIORITY_FRONT_LINE:优先使用前面线路]

obj.tableId	uint32	否	策略路由表ID，默认为0，当前版本不可修改
obj.status	uint8	否	策略路由的状态值，状态字段不可设置或修改
obj.position	int32	否	策略的位置，状态字段不可设置或修改
obj.pbrType	string	是	策略路由类型 枚举值:[SRCADDRESS:源地址策略路由]
obj.outif	object[]	否	策略路由出接口列表
obj.outif.outifName	string	否	出接口名称
obj.outif.gateway	string	否	下一跳ip地址 格式:ip6-address
obj.outif.linkdt	string	否	引用链路探测对象名称

请求样例：

修改多条ipv6策略路由，修改名称为test的策略路由为禁用，修改名称为testother的策略路由目的引用网络对象的全部

```
PATCH https://192.168.1.1/api/batch/v1/namespaces/public/pbrs/ipv6
[
  {
    "name": "test",
    "enable": false
  },
  {
    "name": "testother",
    "dstIpGroups": [
      "全部"
    ]
  }
]
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	IPv6策略路由表项列表
data.uuid	string	策略路由唯一资源ID
data.name	string	策略路由名称 最小长度:1 最大长度:95
data.enable	bool	启禁用开关
data.description	string	描述字段 最小长度:0 最大长度:95
data.schedule	string	时间计划名称
data.srcZones	string[]	关联区域列表
data.srcIpGroups	string[]	源地址引用IP组列表

data.dstIpGroups	string[]	目的地址引用IP组列表
data.dstIsp	string[]	目的地址关联ISP列表
data.dstCountryArea	string[]	目的地址关联国家/地区列表
data.services	string[]	服务对象列表
data.applications	string[]	关联应用识别列表
data.lbMethod	string	多线路模式下，接口选择策略 枚举值:[POLL:轮询 BANDWIDTH:带宽比例 WEIGHTED_MINIMUM_FLOW:加权最小流量 PRIORITY_FRONT_LINE:优先使用前面线路]
data.tableId	uint32	策略路由表ID，默认为0，当前版本不可修改
data.status	uint8	策略路由的状态值，状态字段不可设置或修改
data.position	int32	策略的位置，状态字段不可设置或修改
data.pbrType	string	策略路由类型 枚举值:[SRCADDRESS:源地址策略路由]
data.outif	object[]	策略路由出接口列表
data.outif.outifName	string	出接口名称
data.outif.gateway	string	下一跳ip地址 格式:ip6-address
data.outif.linkdt	string	引用链路探测对象名称

请求成功样例：

```
{
  "data": [
    {
      "enable": false,
      "applications": [
        "全部"
      ],
      "srcIpGroups": [
        "全部"
      ],
      "outif": [
        {
          "outifName": "eth0",
          "gateway": "2000::1"
        }
      ],
      "dstIpGroups": [
        "全部"
      ],
      "srcZones": [
        "manage"
      ],
      "pbrType": "SRCADDRESS",
      "schedule": "全天",
      "services": [
        "any"
      ],
      "uuid": "FBB3947A93714437AECB21B7ABA014DF",
      "name": "test",
      "description": "create test"
    }
  ],
}
```

```
    "enable": true,
    "uuid": "1B853B8C70754BA6984AF24B14CAB9DB",
    "pbrType": "SRCADDRESS",
    "srcZones": [
      "manage"
    ],
    "name": "testother",
    "dstIpGroups": [
      "全部"
    ],
    "schedule": "全天",
    "description": "put testother"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3.5.5 查询IPV4路由状态

3.5.5.1 获取后台IPv4路由信息列表

功能说明：通过此接口，可以获取后台IPv4路由信息

1) API请求数据

GET

https://{host}/api/v1/namespaces/@namespace/routes/ipv4

2) 输入参数

• URL参数

参数	类型	是否必选	描述
_start	int32	否	起始位置, 从 0 开始 最小值:0 最大值:2000000
routeType	string	否	获取路由的类型 枚举值:[ALL_ROUTE:所有路由 STATIC_ROUTE:静态路由 DIRECT_ROUTE:直接路由 OSPF_ROUTE:OSPF 路由 RIP_ROUTE:RIP 路由 VPN_ROUTE:VPN 路由 SSL_VPN_ROUTE:SSL_VPN 路由 IBGP_ROUTE:IBGP 路由 EBGP_ROUTE:EBGP 路由]
_search	string	否	模糊搜索关键字 最大长度:95

_length	int32	否	获取最大长度，限制最大值，如果数据不够可能返回小于 length 最小值:1 最大值:200 默认值:true
---------	-------	---	--

请求样例:

获取后台ipv4路由信息列表，查询条件：查看所有类型的路由，起始位置为0，最多展示100条

GET https://192.168.1.1/api/v1/namespaces/public/routes/ipv4?routeType=ALL_ROUTE&_start=0&_length=100

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	分页数据
data.totalItems	int32	总共多少项目
data.totalPages	int32	总共多少页
data.pageNumber	int32	当前页码，从 1 开始
data.pageSize	int32	每页多大
data.itemsOffset	int32	当前条目偏移, 从 0 开始
data.itemLength	int32	数据列表长度
data.items	object[]	有效数据列表
data.items.routeType	string	类型 最小长度:1 最大长度:95
data.items.prefix	string	目的ip地址/掩码
data.items.gateway	string	下一跳地址
data.items.metric	uint32	度量值
data.items.ifname	string	出接口名称 最小长度:1 最大长度:95

请求成功样例:

```
{
  "data": {
    "itemsOffset": 0,
    "items": [
      {
        "metric": 0,
        "prefix": "56.56.56.0/255.255.255.0",
        "gateway": "0.0.0.0",
        "routeType": "直连路由",
        "ifname": "Tunnel1"
      },
      {
        "metric": 0,
        "prefix": "10.251.251.0/255.255.255.0",
        "gateway": "0.0.0.0",
        "routeType": "直连路由",
      }
    ]
  }
}
```

```
        "ifname": "eth0"
      },
      {
        "metric": 0,
        "prefix": "10.154.0.0/255.254.0.0",
        "gateway": "0.0.0.0",
        "routeType": "直连路由",
        "ifname": "eth0"
      },
      {
        "metric": 0,
        "prefix": "0.0.0.0/0.0.0.0",
        "gateway": "10.155.255.254",
        "routeType": "静态路由",
        "ifname": "eth0"
      }
    ],
    "totalItems": 4,
    "pageNumber": 1,
    "itemLength": 4,
    "totalPages": 1,
    "pageSize": 100
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3.5.6 查询IPV6路由状态

3.5.6.1 获取后台IPv6路由信息列表

功能说明: 通过此接口, 可以获取后台IPv6路由信息

1) API请求数据

GET

https://{host}/api/v1/namespaces/@namespace/routes/ipv6

2) 输入参数

• URL参数

参数	类型	是否必选	描述
_start	int32	否	起始位置, 从 0 开始 最小值:0 最大值:2000000
routeType	string	否	获取路由的类型 枚举值:[ALL_ROUTE:所有路由 STATIC_ROUTE:静态路由 DIRECT_ROUTE:直接路由 OSPF_ROUTE:OSPF 路由 RIP_ROUTE:RIP 路由 VPN_ROUTE:VPN 路由 SSL_VPN_ROUTE:SSL_VPN 路由 IBGP_ROUTE:IBGP 路由 EBGP_ROUTE:EBGP 路由]

_search	string	否	模糊搜索关键字 最大长度:95
_length	int32	否	获取最大长度，限制最大值，如果数据不够可能返回小于 length 最小值:1 最大值:200 默认值:true

请求样例：

获取后台ipv6路由信息列表，查询条件：查看所有静态路由路由，起始位置为0，最多展示100条

GET https://192.168.1.1/api/v1/namespaces/public/routes/ipv6?routeType=STATIC_ROUTE&_start=0
&_length=100

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	分页数据
data.totalItems	int32	总共多少项目
data.totalPages	int32	总共多少页
data.pageNumber	int32	当前页码，从 1 开始
data.pageSize	int32	每页多大
data.itemsOffset	int32	当前条目偏移，从 0 开始
data.itemLength	int32	数据列表长度
data.items	object[]	有效数据列表
data.items.routeType	string	类型 最小长度:1 最大长度:95
data.items.prefix	string	目的IPv6地址/掩码
data.items.gateway	string	下一跳IPv6地址 格式:ip6-address
data.items.metric	uint32	度量值
data.items.ifname	string	出接口名称 最小长度:1 最大长度:95

请求成功样例：

```
{
  "data": {
    "itemsOffset": 0,
    "items": [
      {
        "metric": 256,
        "prefix": "3000::/64",
        "gateway": "2000::161",
        "routeType": "静态路由",
        "ifname": "eth0"
      }
    ]
  }
}
```



```
{
  "metric": 256,
  "prefix": "4000::/64",
  "gateway": "2000::55",
  "routeType": "静态路由",
  "ifname": "eth0"
},
{
  "totalItems": 2,
  "pageNumber": 1,
  "itemLength": 2,
  "totalPages": 1,
  "pageSize": 100
},
{
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

3.5.7 IPV4路由测试

3.5.7.1 测试路由

功能说明：通过此接口，可以进行路由测试

1) API请求数据

POST https://{host}/api/v1/namespaces/@namespace/routes/testroute

2) 输入参数

• Body参数

参数	类型	是否必选	描述
obj	object	是	路由测试匹配参数
obj.dstIp	string	是	目的地址
obj.dstPort	uint32	否	目的端口 最小值:1 最大值:65535
obj.protoNum	uint8	否	协议号 最小值:0 最大值:255
obj.srcIp	string	否	源地址
obj.srcPort	uint32	否	源端口 最小值:1 最大值:65535

请求样例：

ipv4路由测试，查询条件：匹配目的地址200.200.0.0的路由

```
POST https://192.168.1.1/api/v1/namespaces/public/routes/testroute
{
  "dstIp": "200.200.0.0"
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	路由信息列表
data.routeType	string	类型 最小长度:1 最大长度:256
data.prefix	string	目的ip地址/掩码 最小长度:1 最大长度:95
data.gateway	string	下一跳地址 最小长度:1 最大长度:96
data.ifname	string	接口 最小长度:1 最大长度:95
data.index	uint32	匹配优先级

请求成功样例：

```
{
  "data": [
    {
      "index": 0,
      "routeType": "静态路由",
      "gateway": "10.155.255.254",
      "prefix": "0.0.0.0/0.0.0.0",
      "ifname": "eth0"
    }
  ],
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4 策略

4.1 地址转换

4.1.1 NAT策略

4.1.1.1 模拟匹配

功能说明：通过此接口可以查询匹配上的nat策略。

1) API请求数据

POST https://{host}/api/v1/namespaces/@namespace/matchednat

2) 输入参数

• URL参数

参数	类型	是否必选	描述
_length	int32	否	获取最大长度，限制最大值，如果数据不够可能返回小于 length 最小值:1 最大值:200 默认值:true
firstpage	bool	否	是否首入
_start	int32	否	起始位置, 从 0 开始 最小值:0 最大值:2000000

• Body参数

参数	类型	是否必选	描述
obj	object	是	nat模拟匹配body
obj.dstIp	string	否	目的IP
obj.dstPort	int32	否	目的端口 最小值:0 最大值:65535
obj.srcIp	string	否	源ip

请求样例：

模拟匹配查询nat策略

```
POST https://192.168.1.1/api/v1/namespaces/public/matchednat
{
  "srcIp": "1.1.1.1",
  "dstIp": "2.2.2.2",
  "dstPort": 13
}
```

3) 输出参数

• API返回数据

参数	类型	描述
----	----	----

code	int	错误码
message	string	错误信息
data	object	nat模拟匹配return
data.names	string[]	匹配到的策略名数组
data.total	int16	匹配上的总数

请求成功样例：

```
{
  "data": {
    "names": [],
    "total": 0
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.1.1.2 获取nat策略的匹配数

功能说明：nat地址转换策略匹配数查询

1) API请求数据

POST https://{host}/api/batch/v1/namespaces/@namespace/nathit

2) 输入参数

• Body参数

参数	类型	是否必选	描述
obj	object	是	地址转换匹配数请求body
obj.filter	object[]	是	规则名称数组
obj.filter.name	string	是	规则名称 最小长度:1 最大长度:95

请求样例：

批量查询NAT策略匹配数

```
POST https://192.168.1.1/api/batch/v1/namespaces/public/nathit
{
  "filter": [
    {
      "name": "rule1"
    },
    {
      "name": "rule2"
    }
  ]
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	地址转换匹配数数组
data.name	string	规则名称 最小长度:1 最大长度:95
data.hitCount	uint32	规则匹配数 最小值:0

请求成功样例:

```
{
  "data": [
    {
      "name": "rule1",
      "hitCount": 0
    },
    {
      "name": "rule2",
      "hitCount": 0
    }
  ],
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.1.1.3 批量清除nat匹配数

功能说明: nat地址转换策略匹配数批量清除

1) API请求数据

PATCH

https://{host}/api/batch/v1/namespaces/@namespace/nathit

2) 输入参数

- Body参数

参数	类型	是否必选	描述
obj	object	是	地址转换匹配数请求body
obj.filter	object[]	是	规则名称数组
obj.filter.name	string	是	规则名称 最小长度:1 最大长度:95

请求样例:

批量清除NAT策略匹配数

```
PATCH https://192.168.1.1/api/batch/v1/namespaces/public/nathit
{
  "filter": [
    {
      "name": "test2"
    },
    {
      "name": "tes1"
    }
  ]
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	地址转换匹配数数组
data.name	string	规则名称 最小长度:1 最大长度:95
data.hitCount	uint32	规则匹配数 最小值:0

请求成功样例:

```
{
  "data": [
    {
      "name": "test2",
      "hitCount": 0
    },
    {
      "name": "tes1",
      "hitCount": 0
    }
  ],
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.1.1.4 单个清除nat匹配数

功能说明: nat地址转换策略匹配数清除

1) API请求数据

PATCH

https://{host}/api/v1/namespaces/@namespace/nathit/@name

2) 输入参数

- Body参数

参数	类型	是否必选	描述
obj	object	是	清空单个策略匹配数请求body

obj.hitCount	uint8	否	匹配数 最大值:0
--------------	-------	---	--------------

请求样例:

单个清除NAT策略匹配数

```
PATCH https://192.168.1.1/api/v1/namespaces/public/nathit/test
{
  "hitCount": 0
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	地址转换匹配数
data.name	string	规则名称 最小长度:1 最大长度:95
data.hitCount	uint32	规则匹配数 最小值:0

请求成功样例:

```
{
  "data": {
    "name": "test",
    "hitCount": 0
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.1.1.5 获取符合过滤条件的全部nat

功能说明: 获取符合过滤条件的全部nat

1) API请求数据

GET https://{host}/api/v1/namespaces/@namespace/nats

2) 输入参数

- URL 参数

参数	类型	是否必选	描述
_start	int32	否	起始位置, 从 0 开始 最小值:0 最大值:2000000

__nameprefix	string	否	NAT名字前缀过滤 最小长度:1 最大长度:95
_search	string	否	模糊搜索关键字 最大长度:95
_select	string	否	选择字段, 字段用逗号分隔 最大长度:1000
position	int32	否	移动搜索位置参数项
movesearch	string	否	移动搜索参数项 最小长度:1 最大长度:95
transType	string	否	用于指定转换类型, 可选SNAT,DNAT,BNAT 枚举值:[SNAT:转换类型为snat DNAT:转换类型为dnat BNAT:转换类型为bnat]
_length	int32	否	获取最大长度, 限制最大值, 如果数据不够 可能返回小于 length 最小值:1 最大值:200 默认值:true

请求样例:

获取符合过滤条件的全部NAT策略,查询条件: 起始位置为0,返回最多100条数据

GET https://192.168.1.1/api/v1/namespaces/public/nats?_start=0&_length=100

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	分页数据
data.totalItems	int32	总共多少项目
data.totalPages	int32	总共多少页
data.pageNumber	int32	当前页码, 从 1 开始
data.pageSize	int32	每页多大
data.itemsOffset	int32	当前条目偏移, 从 0 开始
data.itemLength	int32	数据列表长度
data.items	object[]	有效数据列表
data.items.uuid	string	唯一标识id
data.items.name	string	唯一标识名称 最小长度:1 最大长度:95

data.items.natHit	int64	nat策略匹配次数
data.items.enable	bool	是否启用
data.items.position	int32	表示位置状态
data.items.natType	string	nat类型 枚举值:[SNAT:转换类型为snat DNAT:转换类型为dnat BNAT:转换类型为bnat]
data.items.description	string	描述 最小长度:0 最大长度:95
data.items.schedule	string	生效时间
data.items.snat	object	snat信息,策略类型为源转换时使用
data.items.snat.srcZones	string[]	源区域
data.items.snat.srcIpGroups	string[]	源ip组
data.items.snat.dstNetobj	object	目的区域或者接口
data.items.snat.dstNetobj.dstNetobjType	string	目的区域类型 枚举值:[ZONE:区域 INTERFACE:接口]
data.items.snat.dstNetobj.interface	string	接口
data.items.snat.dstNetobj.zone	string[]	区域
data.items.snat.dstIpGroups	string[]	目的ip组
data.items.snat.natService	string[]	引用的服务组
data.items.snat.transfer	object	源转换相关数据
data.items.snat.transfer.transferType	string	转换类型 枚举值:[OUTIF_IP:出接口ip IP_RANGE:ip范围 IP:单个ip IPGROUP:ip组 NO_TRANS:不转换]
data.items.snat.transfer.ipRange	object	ip范围 格式:ip-range
data.items.snat.transfer.ipRange.start	string	ip范围下界 格式:ip-address

data.items.snat.transfer.ipRange.end	string	ip范围上界 格式:ip-address
data.items.snat.transfer.ipRange.transferMode	string	转换模式 枚举值:[STATIC:静态转换 DYNAMIC:动态转换]
data.items.snat.transfer.specifyIp	string	单个ip
data.items.snat.transfer.ipGroups	string[]	ip组
data.items.dnat	object	dnat信息,策略类型为目的转换时使用
data.items.dnat.srcZones	string[]	源区域
data.items.dnat.srcIpGroups	string[]	源ip组
data.items.dnat.dstIpObj	object	目的ip组或者ip
data.items.dnat.dstIpObj.dstIpObjType	string	目的类型 枚举值:[IPGROUP:ip组 IP:ip]
data.items.dnat.dstIpObj.specifyIp	string[]	ip数组
data.items.dnat.dstIpObj.ipGroups	string[]	ip组
data.items.dnat.natService	string[]	引用的服务组
data.items.dnat.transfer	object	目的转换相关数据
data.items.dnat.transfer.transferType	string	转换类型 枚举值:[IP_RANGE:ip范围 IP:单个ip IPGROUP:ip组 NO_TRANS:不转换]
data.items.dnat.transfer.dnatEnableScheduling	bool	dnat 使用负载均衡, 开资质才能生效
data.items.dnat.transfer.ipRange	object	ip范围 格式:ip-range
data.items.dnat.transfer.ipRange.start	string	ip范围下界 格式:ip-address
data.items.dnat.transfer.ipRange.end	string	ip范围上界 格式:ip-address
data.items.dnat.transfer.specifyIp	string	单个ip

data.items.dnat.transfer.ipGroups	string[]	ip组
data.items.dnat.transfer.transferPort	object[]	转换端口, 数组为空表示不转换端口
data.items.dnat.transfer.transferPort.start	int32	网络端口起始值 最小值:0 最大值:65535
data.items.dnat.transfer.transferPort.end	int32	网络端口结束值 最小值:0 最大值:65535
data.items.dnat.dnatIgnoreAcl	bool	是否忽略访问控制
data.items.bnat	object	bnat信息,策略类型为双向转换时使用
data.items.bnat.srcZones	string[]	源区域
data.items.bnat.srcIpGroups	string[]	源ip组
data.items.bnat.dstIpobj	object	目的ip组或者ip
data.items.bnat.dstIpobj.dstIpobjType	string	目的类型 枚举值:[IPGROUP:ip组 IP:ip]
data.items.bnat.dstIpobj.specifyIp	string[]	单个ip
data.items.bnat.dstIpobj.ipGroups	string[]	ip组
data.items.bnat.natService	string[]	引用的服务组
data.items.bnat.transferDst	object	目的转换相关数据
data.items.bnat.transferDst.transferType	string	转换类型 枚举值:[IP_RANGE:ip范围 IP:单个ip IPGROUP:ip组 NO_TRANS:不转换]
data.items.bnat.transferDst.dnatEnableScheduling	bool	dnat 使用负载均衡, 开资质才能生效
data.items.bnat.transferDst.ipRange	object	ip范围 格式:ip-range
data.items.bnat.transferDst.ipRange.start	string	ip范围下界 格式:ip-address
data.items.bnat.transferDst.ipRange.end	string	ip范围上界 格式:ip-address

data.items.bnat.transfer Dst.specifyIp	string	单个ip
data.items.bnat.transfer Dst.ipGroups	string[]	ip组
data.items.bnat.transfer Dst.transferPort	object[]	转换端口, 数组为空表示不转换端口
data.items.bnat.transfer Dst.transferPort.start	int32	网络端口起始值 最小值:0 最大值:65535
data.items.bnat.transfer Dst.transferPort.end	int32	网络端口结束值 最小值:0 最大值:65535
data.items.bnat.dnatIgnoreAcl	bool	是否忽略访问控制
data.items.bnat.transfer Src	object	源转换相关数据
data.items.bnat.transfer Src.transferType	string	转换类型 枚举值: OUTIF_IP:出接口ip IP_RANGE:ip范围 IP:单个ip IPGROUP:ip组 NO_TRANS:不转换]
data.items.bnat.transfer Src.ipRange	object	ip范围 格式:ip-range
data.items.bnat.transfer Src.ipRange.start	string	ip范围下界 格式:ip-address
data.items.bnat.transfer Src.ipRange.end	string	ip范围上界 格式:ip-address
data.items.bnat.transfer Src.ipRange.transferMode	string	转换模式 枚举值: STATIC:静态转换 DYNAMIC:动态转换]
data.items.bnat.transfer Src.specifyIp	string	单个ip
data.items.bnat.transfer Src.ipGroups	string[]	ip组

请求成功样例:

```
{
  "data": {
    "itemsOffset": 0,
    "items": [
      {
        "enable": true,
        "position": 0,
        "natType": "SNAT",
        "uuid": "7FB24191361F44D7881E5638A0914BB5",
        "snat": {
          "srcIpGroups": [
            "ipgTest"
          ]
        }
      }
    ]
  }
}
```

```
    ],
    "dstNetobj": {
      "zone": [
        "L3_trust_A"
      ],
      "dstNetobjType": "ZONE"
    },
    "natService": [
      "any"
    ],
    "dstipGroups": [
      "ipgTest"
    ],
    "srcZones": [
      "L3_manage"
    ],
    "transfer": {
      "transferType": "OUTIF_IP"
    }
  },
  "natHit": 0,
  "name": "test",
  "schedule": "全天",
  "description": ""
}
},
"totalItems": 1,
"pageNumber": 1,
"itemLength": 1,
"totalPages": 1,
"pageSize": 100
},
"code": 0,
"message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.1.1.6 批量查询NAT规则

功能说明: 批量查询NAT规则

1) API请求数据

POST https://{host}/api/batch/v1/namespaces/@namespace/nats?_method=get

2) 输入参数

• URL参数

参数	类型	是否必选	描述
_select	string	否	选择字段, 字段用逗号分隔 最大长度:1000

• Body参数

参数	类型	是否必选	描述
obj	object[]	是	nat列表
obj.name	string	是	唯一标识名称 最小长度:1 最大长度:95

obj.enable	bool	否	是否启用 默认值:true
obj.position	int32	否	表示位置状态
obj.natType	string	是	nat类型 枚举值:[SNAT:转换类型为snat DNAT:转换类型为dnat BNAT:转换类型为bnat]
obj.description	string	否	描述 最小长度:0 最大长度:95
obj.schedule	string	否	生效时间 默认值:true
obj.snat	object	是	snat信息,策略类型为源转换时使用
obj.snat.srcZones	string[]	否	源区域
obj.snat.srcIpGroups	string[]	否	源ip组
obj.snat.dstNetobj	object	是	目的区域或者接口
obj.snat.dstNetobj.dstNetobjType	string	否	目的区域类型 默认值:true 枚举值:[ZONE:区域 INTERFACE:接口]
obj.snat.dstNetobj.interface	string	否	接口
obj.snat.dstNetobj.zone	string[]	否	区域
obj.snat.dstIpGroups	string[]	否	目的ip组
obj.snat.natService	string[]	否	引用的服务组
obj.snat.transfer	object	是	源转换相关数据
obj.snat.transfer.transferType	string	否	转换类型 默认值:true 枚举值:[OUTIF_IP:出接口ip IP_RANGE:ip范围 IP:单个ip IPGROUP:ip组 NO_TRANS:不转换]
obj.snat.transfer.ipRange	object	否	ip范围 格式:ip-range
obj.snat.transfer.ipRange.start	string	是	ip范围下界 格式:ip-address
obj.snat.transfer.ipRange.end	string	是	ip范围上界 格式:ip-address

obj.snat.transfer.ipRange.transferMode	string	否	转换模式 默认值:true 枚举值:[STATIC:静态转换 DYNAMIC:动态转换]
obj.snat.transfer.specifyIp	string	否	单个ip
obj.snat.transfer.ipGroups	string[]	否	ip组
obj.dnat	object	是	dnat信息,策略类型为目的转换时使用
obj.dnat.srcZones	string[]	否	源区域
obj.dnat.srcIpGroups	string[]	否	源ip组
obj.dnat.dstIpobj	object	是	目的ip组或者ip
obj.dnat.dstIpobj.dstIpobjType	string	否	目的类型 默认值:true 枚举值:[IPGROUP:ip组 IP:ip]
obj.dnat.dstIpobj.specifyIp	string[]	否	ip数组
obj.dnat.dstIpobj.ipGroups	string[]	否	ip组
obj.dnat.natService	string[]	否	引用的服务组
obj.dnat.transfer	object	是	目的转换相关数据
obj.dnat.transfer.transferType	string	否	转换类型 默认值:true 枚举值:[IP_RANGE:ip范围 IP:单个ip IPGROUP:ip组 NO_TRANS:不转换]
obj.dnat.transfer.dnatEnableScheduling	bool	否	dnat 使用负载均衡, 开资质才能生效
obj.dnat.transfer.ipRange	object	否	ip范围 格式:ip-range
obj.dnat.transfer.ipRange.start	string	是	ip范围下界 格式:ip-address
obj.dnat.transfer.ipRange.end	string	是	ip范围上界 格式:ip-address
obj.dnat.transfer.specifyIp	string	否	单个ip
obj.dnat.transfer.ipGroups	string[]	否	ip组

obj.dnat.transfer.transferPort	object[]	否	转换端口, 数组为空表示不转换端口
obj.dnat.transfer.transferPort.start	int32	是	网络端口起始值 最小值:0 最大值:65535
obj.dnat.transfer.transferPort.end	int32	否	网络端口结束值 最小值:0 最大值:65535
obj.dnat.dnatIgnoreAcl	bool	否	是否忽略访问控制 默认值:true
obj.bnat	object	是	bnat信息,策略类型为双向转换时使用
obj.bnat.srcZones	string[]	否	源区域
obj.bnat.srcIpGroups	string[]	否	源ip组
obj.bnat.dstIpobj	object	是	目的ip组或者ip
obj.bnat.dstIpobj.dstIpobjType	string	否	目的类型 默认值:true 枚举值:[IPGROUP:ip组 IP:ip]
obj.bnat.dstIpobj.specifyIp	string[]	否	单个ip
obj.bnat.dstIpobj.ipGroups	string[]	否	ip组
obj.bnat.natService	string[]	否	引用的服务组
obj.bnat.transferDst	object	是	目的转换相关数据
obj.bnat.transferDst.transferType	string	否	转换类型 默认值:true 枚举值:[IP_RANGE:ip范围 IP:单个ip IPGROUP:ip组 NO_TRANS:不转换]
obj.bnat.transferDst.dnatEnableScheduling	bool	否	dnat 使用负载均衡, 开资质才能生效
obj.bnat.transferDst.ipRange	object	否	ip范围 格式:ip-range
obj.bnat.transferDst.ipRange.start	string	是	ip范围下界 格式:ip-address
obj.bnat.transferDst.ipRange.end	string	是	ip范围上界 格式:ip-address
obj.bnat.transferDst.specifyIp	string	否	单个ip
obj.bnat.transferDst.ipGroups	string[]	否	ip组

obj.bnat.transferDst.transferPort	object[]	否	转换端口, 数组为空表示不转换端口
obj.bnat.transferDst.transferPort.start	int32	是	网络端口起始值 最小值:0 最大值:65535
obj.bnat.transferDst.transferPort.end	int32	否	网络端口结束值 最小值:0 最大值:65535
obj.bnat.dnatIgnoreAcl	bool	否	是否忽略访问控制 默认值:true
obj.bnat.transferSrc	object	是	源转换相关数据
obj.bnat.transferSrc.transferType	string	否	转换类型 默认值:true 枚举值:[OUTIF_IP:出接口ip IP_RANGE:ip范围 IP:单个ip IPGROUP:ip组 NO_TRANS:不转换]
obj.bnat.transferSrc.ipRange	object	否	ip范围 格式:ip-range
obj.bnat.transferSrc.ipRange.start	string	是	ip范围下界 格式:ip-address
obj.bnat.transferSrc.ipRange.end	string	是	ip范围上界 格式:ip-address
obj.bnat.transferSrc.ipRange.transferMode	string	否	转换模式 默认值:true 枚举值:[STATIC:静态转换 DYNAMIC:动态转换]
obj.bnat.transferSrc.specifyIp	string	否	单个ip
obj.bnat.transferSrc.ipGroups	string[]	否	ip组

请求样例:

批量查询NAT规则

```
POST https://192.168.1.1/api/batch/v1/namespaces/public/nats?_method=GET
[
  {
    "name": "test"
  }
]
```

3) 输出参数

- API返回数据

参数	类型	描述
----	----	----

code	int	错误码
message	string	错误信息
data	object[]	nat列表
data.uuid	string	唯一标识id
data.name	string	唯一标识名称 最小长度:1 最大长度:95
data.natHit	int64	nat策略匹配次数
data.enable	bool	是否启用
data.position	int32	表示位置状态
data.natType	string	nat类型 枚举值:[SNAT:转换类型为snat DNAT:转换类型为dnat BNAT:转换类型为bnat]
data.description	string	描述 最小长度:0 最大长度:95
data.schedule	string	生效时间
data.snat	object	snat信息,策略类型为源转换时使用
data.snat.srcZones	string[]	源区域
data.snat.srcIpGroups	string[]	源ip组
data.snat.dstNetobj	object	目的区域或者接口
data.snat.dstNetobj.dstNetobjType	string	目的区域类型 枚举值:[ZONE:区域 INTERFACE:接口]
data.snat.dstNetobj.interface	string	接口
data.snat.dstNetobj.zone	string[]	区域
data.snat.dstIpGroups	string[]	目的ip组
data.snat.natService	string[]	引用的服务组
data.snat.transfer	object	源转换相关数据
data.snat.transfer.transferType	string	转换类型 枚举值:[OUTIF_IP:出接口ip IP_RANGE:ip范围 IP:单个ip IPGROUP:ip组 NO_TRANS:不转换]

data.snat.transfer.ipRange	object	ip范围 格式:ip-range
data.snat.transfer.ipRange.start	string	ip范围下界 格式:ip-address
data.snat.transfer.ipRange.end	string	ip范围上界 格式:ip-address
data.snat.transfer.ipRange.transferMode	string	转换模式 枚举值:[STATIC:静态转换 DYNAMIC:动态转换]
data.snat.transfer.specifyIp	string	单个ip
data.snat.transfer.ipGroups	string[]	ip组
data.dnat	object	dnat信息,策略类型为目的转换时使用
data.dnat.srcZones	string[]	源区域
data.dnat.srcIpGroups	string[]	源ip组
data.dnat.dstIpobj	object	目的ip组或者ip
data.dnat.dstIpobj.dstIpobjType	string	目的类型 枚举值:[IPGROUP:ip组 IP:ip]
data.dnat.dstIpobj.specifyIp	string[]	ip数组
data.dnat.dstIpobj.ipGroups	string[]	ip组
data.dnat.natService	string[]	引用的服务组
data.dnat.transfer	object	目的转换相关数据
data.dnat.transfer.transferType	string	转换类型 枚举值:[IP_RANGE:ip范围 IP:单个ip IPGROUP:ip组 NO_TRANS:不转换]
data.dnat.transfer.dnatEnableScheduling	bool	dnat 使用负载均衡, 开资质才能生效
data.dnat.transfer.ipRange	object	ip范围 格式:ip-range
data.dnat.transfer.ipRange.start	string	ip范围下界 格式:ip-address
data.dnat.transfer.ipRange.end	string	ip范围上界 格式:ip-address

data.dnat.transfer.specifyIp	string	单个ip
data.dnat.transfer.ipGroups	string[]	ip组
data.dnat.transfer.transferPort	object[]	转换端口, 数组为空表示不转换端口
data.dnat.transfer.transferPort.start	int32	网络端口起始值 最小值:0 最大值:65535
data.dnat.transfer.transferPort.end	int32	网络端口结束值 最小值:0 最大值:65535
data.dnat.dnatIgnoreAcl	bool	是否忽略访问控制
data.bnat	object	bnat信息,策略类型为双向转换时使用
data.bnat.srcZones	string[]	源区域
data.bnat.srcIpGroups	string[]	源ip组
data.bnat.dstIpobj	object	目的ip组或者ip
data.bnat.dstIpobj.dstIpobjType	string	目的类型 枚举值:[IPGROUP:ip组 IP:ip]
data.bnat.dstIpobj.specifyIp	string[]	单个ip
data.bnat.dstIpobj.ipGroups	string[]	ip组
data.bnat.natService	string[]	引用的服务组
data.bnat.transferDst	object	目的转换相关数据
data.bnat.transferDst.transferType	string	转换类型 枚举值:[IP_RANGE:ip范围 IP:单个ip IPGROUP:ip组 NO_TRANS:不转换]
data.bnat.transferDst.dnatEnableScheduling	bool	dnat 使用负载均衡, 开资质才能生效
data.bnat.transferDst.ipRange	object	ip范围 格式:ip-range
data.bnat.transferDst.ipRange.start	string	ip范围下界 格式:ip-address
data.bnat.transferDst.ipRange.end	string	ip范围上界 格式:ip-address
data.bnat.transferDst.specifyIp	string	单个ip

data.bnat.transferDst.ipGroups	string[]	ip组
data.bnat.transferDst.transferPort	object[]	转换端口, 数组为空表示不转换端口
data.bnat.transferDst.transferPort.start	int32	网络端口起始值 最小值:0 最大值:65535
data.bnat.transferDst.transferPort.end	int32	网络端口结束值 最小值:0 最大值:65535
data.bnat.dnatIgnoreAcl	bool	是否忽略访问控制
data.bnat.transferSrc	object	源转换相关数据
data.bnat.transferSrc.transferType	string	转换类型 枚举值: OUTIF_IP:出接口ip IP_RANGE:ip范围 IP:单个ip IPGROUP:ip组 NO_TRANS:不转换]
data.bnat.transferSrc.ipRange	object	ip范围 格式:ip-range
data.bnat.transferSrc.ipRange.start	string	ip范围下界 格式:ip-address
data.bnat.transferSrc.ipRange.end	string	ip范围上界 格式:ip-address
data.bnat.transferSrc.ipRange.transferMode	string	转换模式 枚举值: STATIC:静态转换 DYNAMIC:动态转换]
data.bnat.transferSrc.specifyIp	string	单个ip
data.bnat.transferSrc.ipGroups	string[]	ip组

请求成功样例:

```
{
  "data": [
    {
      "enable": true,
      "position": 0,
      "natType": "SNAT",
      "uuid": "7FB24191361F44D7881E5638A0914BB5",
      "snat": {
        "srcIpGroups": [
          "ipgTest"
        ],
        "dstNetobj": {
          "zone": [
            "L3_trust_A"
          ],
          "dstNetobjType": "ZONE"
        }
      },
      "natService": [

```

```
        "any"
      ],
      "dstIpGroups": [
        "ipgTest"
      ],
      "srcZones": [
        "L3_manage"
      ],
      "transfer": {
        "transferType": "OUTIF_IP"
      }
    },
    "natHit": 0,
    "name": "test",
    "schedule": "全天",
    "description": ""
  }
},
"code": 0,
"message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.1.1.7 获取单条nat详细信息

功能说明: 获取单条nat详细信息

1) API请求数据

GET

https://{host}/api/v1/namespaces/@namespace/nats/@name

2) 输入参数

- URL 参数

参数	类型	是否必选	描述
_select	string	否	选择字段, 字段用逗号分隔 最大长度:1000

请求样例:

获取单条NAT策略详细信息,查询条件: 查看名为test的NAT策略

GET https://192.168.1.1/api/v1/namespaces/public/nats/test

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	nat信息
data.uuid	string	唯一标识id
data.name	string	唯一标识名称 最小长度:1 最大长度:95

data.natHit	int64	nat策略匹配次数
data.enable	bool	是否启用
data.position	int32	表示位置状态
data.natType	string	nat类型 枚举值:[SNAT:转换类型为snat DNAT:转换类型为dnat BNAT:转换类型为bnat]
data.description	string	描述 最小长度:0 最大长度:95
data.schedule	string	生效时间
data.snat	object	snat信息,策略类型为源转换时使用
data.snat.srcZones	string[]	源区域
data.snat.srcIpGroups	string[]	源ip组
data.snat.dstNetobj	object	目的区域或者接口
data.snat.dstNetobj.dstNetobjType	string	目的区域类型 枚举值:[ZONE:区域 INTERFACE:接口]
data.snat.dstNetobj.interface	string	接口
data.snat.dstNetobj.zone	string[]	区域
data.snat.dstIpGroups	string[]	目的ip组
data.snat.natService	string[]	引用的服务组
data.snat.transfer	object	源转换相关数据
data.snat.transfer.transferType	string	转换类型 枚举值:[OUTIF_IP:出接口ip IP_RANGE:ip范围 IP:单个ip IPGROUP:ip组 NO_TRANS:不转换]
data.snat.transfer.ipRange	object	ip范围 格式:ip-range
data.snat.transfer.ipRange.start	string	ip范围下界 格式:ip-address
data.snat.transfer.ipRange.end	string	ip范围上界 格式:ip-address

data.snat.transfer.ipRange.transferMode	string	转换模式 枚举值:[STATIC:静态转换 DYNAMIC:动态转换]
data.snat.transfer.specifyIp	string	单个ip
data.snat.transfer.ipGroups	string[]	ip组
data.dnat	object	dnat信息,策略类型为目的转换时使用
data.dnat.srcZones	string[]	源区域
data.dnat.srcIpGroups	string[]	源ip组
data.dnat.dstIpobj	object	目的ip组或者ip
data.dnat.dstIpobj.dstIpobjType	string	目的类型 枚举值:[IPGROUP:ip组 IP:ip]
data.dnat.dstIpobj.specifyIp	string[]	ip数组
data.dnat.dstIpobj.ipGroups	string[]	ip组
data.dnat.natService	string[]	引用的服务组
data.dnat.transfer	object	目的转换相关数据
data.dnat.transfer.transferType	string	转换类型 枚举值:[IP_RANGE:ip范围 IP:单个ip IPGROUP:ip组 NO_TRANS:不转换]
data.dnat.transfer.dnatEnableScheduling	bool	dnat 使用负载均衡, 开资质才能生效
data.dnat.transfer.ipRange	object	ip范围 格式:ip-range
data.dnat.transfer.ipRange.start	string	ip范围下界 格式:ip-address
data.dnat.transfer.ipRange.end	string	ip范围上界 格式:ip-address
data.dnat.transfer.specifyIp	string	单个ip
data.dnat.transfer.ipGroups	string[]	ip组
data.dnat.transfer.transferPort	object[]	转换端口, 数组为空表示不转换端口

data.dnat.transfer.transferPort.start	int32	网络端口起始值 最小值:0 最大值:65535
data.dnat.transfer.transferPort.end	int32	网络端口结束值 最小值:0 最大值:65535
data.dnat.dnatIgnoreAcl	bool	是否忽略访问控制
data.bnat	object	bnat信息,策略类型为双向转换时使用
data.bnat.srcZones	string[]	源区域
data.bnat.srcIpGroups	string[]	源ip组
data.bnat.dstIpobj	object	目的ip组或者ip
data.bnat.dstIpobj.dstIpobjType	string	目的类型 枚举值:[IPGROUP:ip组 IP:ip]
data.bnat.dstIpobj.specifyIp	string[]	单个ip
data.bnat.dstIpobj.ipGroups	string[]	ip组
data.bnat.natService	string[]	引用的服务组
data.bnat.transferDst	object	目的转换相关数据
data.bnat.transferDst.transferType	string	转换类型 枚举值:[IP_RANGE:ip范围 IP:单个ip IPGROUP:ip组 NO_TRANS:不转换]
data.bnat.transferDst.dnatEnableScheduling	bool	dnat 使用负载均衡, 开资质才能生效
data.bnat.transferDst.ipRange	object	ip范围 格式:ip-range
data.bnat.transferDst.ipRange.start	string	ip范围下界 格式:ip-address
data.bnat.transferDst.ipRange.end	string	ip范围上界 格式:ip-address
data.bnat.transferDst.specifyIp	string	单个ip
data.bnat.transferDst.ipGroups	string[]	ip组
data.bnat.transferDst.transferPort	object[]	转换端口, 数组为空表示不转换端口
data.bnat.transferDst.transferPort.start	int32	网络端口起始值 最小值:0 最大值:65535

data.bnat.transferDst.transferPort.end	int32	网络端口结束值 最小值:0 最大值:65535
data.bnat.dnatIgnoreAcl	bool	是否忽略访问控制
data.bnat.transferSrc	object	源转换相关数据
data.bnat.transferSrc.transferType	string	转换类型 枚举值: OUTIF_IP:出接口ip IP_RANGE:ip范围 IP:单个ip IPGROUP:ip组 NO_TRANS:不转换]
data.bnat.transferSrc.ipRange	object	ip范围 格式:ip-range
data.bnat.transferSrc.ipRange.start	string	ip范围下界 格式:ip-address
data.bnat.transferSrc.ipRange.end	string	ip范围上界 格式:ip-address
data.bnat.transferSrc.ipRange.transferMode	string	转换模式 枚举值: STATIC:静态转换 DYNAMIC:动态转换]
data.bnat.transferSrc.specifyIp	string	单个ip
data.bnat.transferSrc.ipGroups	string[]	ip组

请求成功样例:

```
{
  "data": {
    "enable": true,
    "position": 0,
    "natType": "SNAT",
    "uuid": "7FB24191361F44D7881E5638A0914BB5",
    "snat": {
      "srcIpGroups": [
        "ipgTest"
      ],
      "dstNetobj": {
        "zone": [
          "L3_trust_A"
        ],
        "dstNetobjType": "ZONE"
      },
      "natService": [
        "any"
      ],
      "dstIpGroups": [
        "ipgTest"
      ],
      "srcZones": [
        "L3_manage"
      ],
      "transfer": {
        "transferType": "OUTIF_IP"
      }
    }
  },
}
```

```
{
  "natHit": 0,
  "name": "test",
  "schedule": "全天",
  "description": ""
},
{
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.1.1.8 添加nat

功能说明: 添加一条nat策略

1) API请求数据

POST <https://{host}/api/v1/namespaces/@namespace/nats>

2) 输入参数

• URL 参数

参数	类型	是否必选	描述
_where	string	否	指定位置, 用于移动/插入 枚举值:[top:顶部 bottom:底部 before:在目标位置之前 after:在目标位置之后]
_dest	string	否	目标位置关键字对应的值 最大长度:100
_key	string	否	目标位置关键字 枚举值:[position:指定目标位置关键字为 position name:指定目标位置关键字为 name]

• Body参数

参数	类型	是否必选	描述
obj	object	是	nat信息
obj.name	string	是	唯一标识名称 最小长度:1 最大长度:95
obj.enable	bool	否	是否启用 默认值:true
obj.position	int32	否	表示位置状态

obj.natType	string	是	nat类型 枚举值:[SNAT:转换类型为snat DNAT:转换类型为dnat BNAT:转换类型为bnat]
obj.description	string	否	描述 最小长度:0 最大长度:95
obj.schedule	string	否	生效时间 默认值:true
obj.snat	object	是	snat信息,策略类型为源转换时使用
obj.snat.srcZones	string[]	否	源区域
obj.snat.srcIpGroups	string[]	否	源ip组
obj.snat.dstNetobj	object	是	目的区域或者接口
obj.snat.dstNetobj.dstNetobjType	string	否	目的区域类型 默认值:true 枚举值:[ZONE:区域 INTERFACE:接口]
obj.snat.dstNetobj.interface	string	否	接口
obj.snat.dstNetobj.zone	string[]	否	区域
obj.snat.dstIpGroups	string[]	否	目的ip组
obj.snat.natService	string[]	否	引用的服务组
obj.snat.transfer	object	是	源转换相关数据
obj.snat.transfer.transferType	string	否	转换类型 默认值:true 枚举值:[OUTIF_IP:出接口ip IP_RANGE:ip范围 IP:单个ip IPGROUP:ip组 NO_TRANS:不转换]
obj.snat.transfer.ipRange	object	否	ip范围 格式:ip-range
obj.snat.transfer.ipRange.start	string	是	ip范围下界 格式:ip-address
obj.snat.transfer.ipRange.end	string	是	ip范围上界 格式:ip-address
obj.snat.transfer.ipRange.transferMode	string	否	转换模式 默认值:true 枚举值:[STATIC:静态转换 DYNAMIC:动态转换]

obj.snat.transfer.specifyIp	string	否	单个ip
obj.snat.transfer.ipGroups	string[]	否	ip组
obj.dnat	object	是	dnat信息,策略类型为目的转换时使用
obj.dnat.srcZones	string[]	否	源区域
obj.dnat.srcIpGroups	string[]	否	源ip组
obj.dnat.dstIpobj	object	是	目的ip组或者ip
obj.dnat.dstIpobj.dstIpobjType	string	否	目的类型 默认值:true 枚举值:[IPGROUP:ip组 IP:ip]
obj.dnat.dstIpobj.specifyIp	string[]	否	ip数组
obj.dnat.dstIpobj.ipGroups	string[]	否	ip组
obj.dnat.natService	string[]	否	引用的服务组
obj.dnat.transfer	object	是	目的转换相关数据
obj.dnat.transfer.transferType	string	否	转换类型 默认值:true 枚举值:[IP_RANGE:ip范围 IP:单个ip IPGROUP:ip组 NO_TRANS:不转换]
obj.dnat.transfer.dnatEnableScheduling	bool	否	dnat 使用负载均衡, 开资质才能生效
obj.dnat.transfer.ipRange	object	否	ip范围 格式:ip-range
obj.dnat.transfer.ipRange.start	string	是	ip范围下界 格式:ip-address
obj.dnat.transfer.ipRange.end	string	是	ip范围上界 格式:ip-address
obj.dnat.transfer.specifyIp	string	否	单个ip
obj.dnat.transfer.ipGroups	string[]	否	ip组
obj.dnat.transfer.transferPort	object[]	否	转换端口, 数组为空表示不转换端口
obj.dnat.transfer.transferPort.start	int32	是	网络端口起始值 最小值:0 最大值:65535

obj.dnat.transfer.transferPort.end	int32	否	网络端口结束值 最小值:0 最大值:65535
obj.dnat.dnatIgnoreAcl	bool	否	是否忽略访问控制 默认值:true
obj.bnat	object	是	bnat信息,策略类型为双向转换时使用
obj.bnat.srcZones	string[]	否	源区域
obj.bnat.srcIpGroups	string[]	否	源ip组
obj.bnat.dstIpobj	object	是	目的ip组或者ip
obj.bnat.dstIpobj.dstIpobjType	string	否	目的类型 默认值:true 枚举值:[IPGROUP:ip组 IP:ip]
obj.bnat.dstIpobj.specifyIp	string[]	否	单个ip
obj.bnat.dstIpobj.ipGroups	string[]	否	ip组
obj.bnat.natService	string[]	否	引用的服务组
obj.bnat.transferDst	object	是	目的转换相关数据
obj.bnat.transferDst.transferType	string	否	转换类型 默认值:true 枚举值:[IP_RANGE:ip范围 IP:单个ip IPGROUP:ip组 NO_TRANS:不转换]
obj.bnat.transferDst.dnatEnableScheduling	bool	否	dnat 使用负载均衡, 开资质才能生效
obj.bnat.transferDst.ipRange	object	否	ip范围 格式:ip-range
obj.bnat.transferDst.ipRange.start	string	是	ip范围下界 格式:ip-address
obj.bnat.transferDst.ipRange.end	string	是	ip范围上界 格式:ip-address
obj.bnat.transferDst.specifyIp	string	否	单个ip
obj.bnat.transferDst.ipGroups	string[]	否	ip组
obj.bnat.transferDst.transferPort	object[]	否	转换端口, 数组为空表示不转换端口
obj.bnat.transferDst.transferPort.start	int32	是	网络端口起始值 最小值:0 最大值:65535

obj.bnat.transferDst.transferPort.end	int32	否	网络端口结束值 最小值:0 最大值:65535
obj.bnat.dnatIgnoreAcl	bool	否	是否忽略访问控制 默认值:true
obj.bnat.transferSrc	object	是	源转换相关数据
obj.bnat.transferSrc.transferType	string	否	转换类型 默认值:true 枚举值:[OUTIF_IP:出接口ip IP_RANGE:ip范围 IP:单个ip IPGROUP:ip组 NO_TRANS:不转换]
obj.bnat.transferSrc.ipRange	object	否	ip范围 格式:ip-range
obj.bnat.transferSrc.ipRange.start	string	是	ip范围下界 格式:ip-address
obj.bnat.transferSrc.ipRange.end	string	是	ip范围上界 格式:ip-address
obj.bnat.transferSrc.ipRange.transferMode	string	否	转换模式 默认值:true 枚举值:[STATIC:静态转换 DYNAMIC:动态转换]
obj.bnat.transferSrc.specifyIp	string	否	单个ip
obj.bnat.transferSrc.ipGroups	string[]	否	ip组

请求样例:

添加一条NAT策略

POST https://192.168.1.1/api/v1/namespaces/public/nats

```
{
  "enable": true,
  "snat": {
    "srcIpGroups": [
      "ipgTest"
    ],
    "dstNetobj": {
      "zone": [
        "L3_trust_A"
      ],
      "dstNetobjType": "ZONE"
    },
    "natService": [
      "any"
    ],
    "dstIpGroups": [
      "ipgTest"
    ],
    "srcZones": [
      "L3_manage"
    ],
    "transfer": {
```

```

        "transferType": "OUTIF_IP"
    },
    "schedule": "全天",
    "natType": "SNAT",
    "name": "test1",
    "description": ""
}

```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	nat信息
data.uuid	string	唯一标识id
data.name	string	唯一标识名称 最小长度:1 最大长度:95
data.natHit	int64	nat策略匹配次数
data.enable	bool	是否启用
data.position	int32	表示位置状态
data.natType	string	nat类型 枚举值: SNAT:转换类型为snat DNAT:转换类型为dnat BNAT:转换类型为bnat]
data.description	string	描述 最小长度:0 最大长度:95
data.schedule	string	生效时间
data.sn timer	object	snat信息,策略类型为源转换时使用
data.sn timer.srcZones	string[]	源区域
data.sn timer.srcIpGroups	string[]	源ip组
data.sn timer.dstNetobj	object	目的区域或者接口
data.sn timer.dstNetobj.dstNetobjType	string	目的区域类型 枚举值: ZONE:区域 INTERFACE:接口]
data.sn timer.dstNetobj.interface	string	接口
data.sn timer.dstNetobj.zone	string[]	区域
data.sn timer.dstIpGroups	string[]	目的ip组
data.sn timer.natService	string[]	引用的服务组

data.snat.transfer	object	源转换相关数据
data.snat.transfer.transferType	string	转换类型 枚举值:[OUTIF_IP:出接口ip IP_RANGE:ip范围 IP:单个ip IPGROUP:ip组 NO_TRANS:不转换]
data.snat.transfer.ipRange	object	ip范围 格式:ip-range
data.snat.transfer.ipRange.start	string	ip范围下界 格式:ip-address
data.snat.transfer.ipRange.end	string	ip范围上界 格式:ip-address
data.snat.transfer.ipRange.transferMode	string	转换模式 枚举值:[STATIC:静态转换 DYNAMIC:动态转换]
data.snat.transfer.specifyIp	string	单个ip
data.snat.transfer.ipGroups	string[]	ip组
data.dnat	object	dnat信息,策略类型为目的转换时使用
data.dnat.srcZones	string[]	源区域
data.dnat.srcIpGroups	string[]	源ip组
data.dnat.dstIpobj	object	目的ip组或者ip
data.dnat.dstIpobj.dstIpobjType	string	目的类型 枚举值:[IPGROUP:ip组 IP:ip]
data.dnat.dstIpobj.specifyIp	string[]	ip数组
data.dnat.dstIpobj.ipGroups	string[]	ip组
data.dnat.natService	string[]	引用的服务组
data.dnat.transfer	object	目的转换相关数据
data.dnat.transfer.transferType	string	转换类型 枚举值:[IP_RANGE:ip范围 IP:单个ip IPGROUP:ip组 NO_TRANS:不转换]

data.dnat.transfer.dnatEnableScheduling	bool	dnat 使用负载均衡, 开资质才能生效
data.dnat.transfer.ipRange	object	ip范围 格式:ip-range
data.dnat.transfer.ipRange.start	string	ip范围下界 格式:ip-address
data.dnat.transfer.ipRange.end	string	ip范围上界 格式:ip-address
data.dnat.transfer.specifyIp	string	单个ip
data.dnat.transfer.ipGroups	string[]	ip组
data.dnat.transfer.transferPort	object[]	转换端口, 数组为空表示不转换端口
data.dnat.transfer.transferPort.start	int32	网络端口起始值 最小值:0 最大值:65535
data.dnat.transfer.transferPort.end	int32	网络端口结束值 最小值:0 最大值:65535
data.dnat.dnatIgnoreAcl	bool	是否忽略访问控制
data.bnat	object	bnat信息,策略类型为双向转换时使用
data.bnat.srcZones	string[]	源区域
data.bnat.srcIpGroups	string[]	源ip组
data.bnat.dstIpobj	object	目的ip组或者ip
data.bnat.dstIpobj.dstIpobjType	string	目的类型 枚举值:[IPGROUP:ip组 IP:ip]
data.bnat.dstIpobj.specifyIp	string[]	单个ip
data.bnat.dstIpobj.ipGroups	string[]	ip组
data.bnat.natService	string[]	引用的服务组
data.bnat.transferDst	object	目的转换相关数据
data.bnat.transferDst.transferType	string	转换类型 枚举值:[IP_RANGE:ip范围 IP:单个ip IPGROUP:ip组 NO_TRANS:不转换]
data.bnat.transferDst.dnatEnableScheduling	bool	dnat 使用负载均衡, 开资质才能生效

data.bnat.transferDst.ipRange	object	ip范围 格式:ip-range
data.bnat.transferDst.ipRange.start	string	ip范围下界 格式:ip-address
data.bnat.transferDst.ipRange.end	string	ip范围上界 格式:ip-address
data.bnat.transferDst.specifyIp	string	单个ip
data.bnat.transferDst.ipGroups	string[]	ip组
data.bnat.transferDst.transferPort	object[]	转换端口, 数组为空表示不转换端口
data.bnat.transferDst.transferPort.start	int32	网络端口起始值 最小值:0 最大值:65535
data.bnat.transferDst.transferPort.end	int32	网络端口结束值 最小值:0 最大值:65535
data.bnat.dnatIgnoreAcl	bool	是否忽略访问控制
data.bnat.transferSrc	object	源转换相关数据
data.bnat.transferSrc.transferType	string	转换类型 枚举值: OUTIF_IP:出接口ip IP_RANGE:ip范围 IP:单个ip IPGROUP:ip组 NO_TRANS:不转换]
data.bnat.transferSrc.ipRange	object	ip范围 格式:ip-range
data.bnat.transferSrc.ipRange.start	string	ip范围下界 格式:ip-address
data.bnat.transferSrc.ipRange.end	string	ip范围上界 格式:ip-address
data.bnat.transferSrc.ipRange.transferMode	string	转换模式 枚举值: STATIC:静态转换 DYNAMIC:动态转换]
data.bnat.transferSrc.specifyIp	string	单个ip
data.bnat.transferSrc.ipGroups	string[]	ip组

请求成功样例:

```
{
  "data": {
    "enable": true,
    "snat": {
```

```
      "srcIpGroups": [
        "ipgTest"
      ],
      "dstNetobj": {
        "zone": [
          "L3_trust_A"
        ],
        "dstNetobjType": "ZONE"
      },
      "natService": [
        "any"
      ],
      "dstIpGroups": [
        "ipgTest"
      ],
      "srcZones": [
        "L3_manage"
      ],
      "transfer": {
        "transferType": "OUTIF_IP"
      }
    },
    "schedule": "全天",
    "name": "test1",
    "natType": "SNAT",
    "uuid": "0003B4F322A64205912212AB6890892F",
    "description": ""
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.1.1.9 删除单条nat

功能说明：删除单条nat

1) API请求数据

DELETE

https://{host}/api/v1/namespaces/@namespace/nats/@name

2) 输入参数

请求样例：

删除单条NAT策略

DELETE https://192.168.1.1/api/v1/namespaces/public/nats/test

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	nat信息
data.uuid	string	唯一标识id

data.name	string	唯一标识名称 最小长度:1 最大长度:95
data.natHit	int64	nat策略匹配次数
data.enable	bool	是否启用
data.position	int32	表示位置状态
data.natType	string	nat类型 枚举值:[SNAT:转换类型为snat DNAT:转换类型为dnat BNAT:转换类型为bnat]
data.description	string	描述 最小长度:0 最大长度:95
data.schedule	string	生效时间
data.snat	object	snat信息,策略类型为源转换时使用
data.snat.srcZones	string[]	源区域
data.snat.srcIpGroups	string[]	源ip组
data.snat.dstNetobj	object	目的区域或者接口
data.snat.dstNetobj.dstNetobjType	string	目的区域类型 枚举值:[ZONE:区域 INTERFACE:接口]
data.snat.dstNetobj.interface	string	接口
data.snat.dstNetobj.zone	string[]	区域
data.snat.dstIpGroups	string[]	目的ip组
data.snat.natService	string[]	引用的服务组
data.snat.transfer	object	源转换相关数据
data.snat.transfer.transferType	string	转换类型 枚举值:[OUTIF_IP:出接口ip IP_RANGE:ip范围 IP:单个ip IPGROUP:ip组 NO_TRANS:不转换]
data.snat.transfer.ipRange	object	ip范围 格式:ip-range
data.snat.transfer.ipRange.start	string	ip范围下界 格式:ip-address
data.snat.transfer.ipRange.end	string	ip范围上界 格式:ip-address

data.snat.transfer.ipRange.transferMode	string	转换模式 枚举值:[STATIC:静态转换 DYNAMIC:动态转换]
data.snat.transfer.specifyIp	string	单个ip
data.snat.transfer.ipGroups	string[]	ip组
data.dnat	object	dnat信息,策略类型为目的转换时使用
data.dnat.srcZones	string[]	源区域
data.dnat.srcIpGroups	string[]	源ip组
data.dnat.dstIpobj	object	目的ip组或者ip
data.dnat.dstIpobj.dstIpobjType	string	目的类型 枚举值:[IPGROUP:ip组 IP:ip]
data.dnat.dstIpobj.specifyIp	string[]	ip数组
data.dnat.dstIpobj.ipGroups	string[]	ip组
data.dnat.natService	string[]	引用的服务组
data.dnat.transfer	object	目的转换相关数据
data.dnat.transfer.transferType	string	转换类型 枚举值:[IP_RANGE:ip范围 IP:单个ip IPGROUP:ip组 NO_TRANS:不转换]
data.dnat.transfer.dnatEnableScheduling	bool	dnat 使用负载均衡, 开资质才能生效
data.dnat.transfer.ipRange	object	ip范围 格式:ip-range
data.dnat.transfer.ipRange.start	string	ip范围下界 格式:ip-address
data.dnat.transfer.ipRange.end	string	ip范围上界 格式:ip-address
data.dnat.transfer.specifyIp	string	单个ip
data.dnat.transfer.ipGroups	string[]	ip组
data.dnat.transfer.transferPort	object[]	转换端口, 数组为空表示不转换端口

data.dnat.transfer.transferPort.start	int32	网络端口起始值 最小值:0 最大值:65535
data.dnat.transfer.transferPort.end	int32	网络端口结束值 最小值:0 最大值:65535
data.dnat.dnatIgnoreAcl	bool	是否忽略访问控制
data.bnat	object	bnat信息,策略类型为双向转换时使用
data.bnat.srcZones	string[]	源区域
data.bnat.srcIpGroups	string[]	源ip组
data.bnat.dstIpobj	object	目的ip组或者ip
data.bnat.dstIpobj.dstIpobjType	string	目的类型 枚举值:[IPGROUP:ip组 IP:ip]
data.bnat.dstIpobj.specifyIp	string[]	单个ip
data.bnat.dstIpobj.ipGroups	string[]	ip组
data.bnat.natService	string[]	引用的服务组
data.bnat.transferDst	object	目的转换相关数据
data.bnat.transferDst.transferType	string	转换类型 枚举值:[IP_RANGE:ip范围 IP:单个ip IPGROUP:ip组 NO_TRANS:不转换]
data.bnat.transferDst.dnatEnableScheduling	bool	dnat 使用负载均衡, 开资质才能生效
data.bnat.transferDst.ipRange	object	ip范围 格式:ip-range
data.bnat.transferDst.ipRange.start	string	ip范围下界 格式:ip-address
data.bnat.transferDst.ipRange.end	string	ip范围上界 格式:ip-address
data.bnat.transferDst.specifyIp	string	单个ip
data.bnat.transferDst.ipGroups	string[]	ip组
data.bnat.transferDst.transferPort	object[]	转换端口, 数组为空表示不转换端口
data.bnat.transferDst.transferPort.start	int32	网络端口起始值 最小值:0 最大值:65535

data.bnat.transferDst.transferPort.end	int32	网络端口结束值 最小值:0 最大值:65535
data.bnat.dnatIgnoreAcl	bool	是否忽略访问控制
data.bnat.transferSrc	object	源转换相关数据
data.bnat.transferSrc.transferType	string	转换类型 枚举值: OUTIF_IP:出接口ip IP_RANGE:ip范围 IP:单个ip IPGROUP:ip组 NO_TRANS:不转换]
data.bnat.transferSrc.ipRange	object	ip范围 格式:ip-range
data.bnat.transferSrc.ipRange.start	string	ip范围下界 格式:ip-address
data.bnat.transferSrc.ipRange.end	string	ip范围上界 格式:ip-address
data.bnat.transferSrc.ipRange.transferMode	string	转换模式 枚举值: STATIC:静态转换 DYNAMIC:动态转换]
data.bnat.transferSrc.specifyIp	string	单个ip
data.bnat.transferSrc.ipGroups	string[]	ip组

请求成功样例:

```
{
  "data": {
    "enable": true,
    "snat": {
      "srcIpGroups": [
        "ipgTest"
      ],
      "dstNetobj": {
        "zone": [
          "L3_trust_A"
        ],
        "dstNetobjType": "ZONE"
      },
      "natService": [
        "any"
      ],
      "dstIpGroups": [
        "ipgTest"
      ],
      "srcZones": [
        "L3_manage"
      ],
      "transfer": {
        "transferType": "OUTIF_IP"
      }
    },
    "schedule": "全天",
    "name": "test1",
    "natType": "SNAT",
```



```

    "uuid": "0003B4F322A64205912212AB6890892F",
    "description": ""
  },
  "code": 0,
  "message": "成功"
}

```

请求失败样例:

```

{
  "data": "",
  "code": 22,
  "message": "无效参数"
}

```

4.1.1.10 增量修改一条nat配置

功能说明: 增量修改一条nat配置

1) API请求数据

PATCH <https://{host}/api/v1/namespaces/@namespace/nats/@name>

2) 输入参数

• URL参数

参数	类型	是否必选	描述
_arrayop	string	否	对象属性是数组的操作控制 枚举值:[add:添加对应的数组元素 remove:删除对应的数组元素]
_key	string	否	目标位置关键字 枚举值:[position:指定目标位置关键字为 position name:指定目标位置关键字为 name]
_dest	string	否	目标位置关键字对应的值 最大长度:100
_where	string	否	指定位置, 用于移动/插入 枚举值:[top:顶部 bottom:底部 before:在目标位置之前 after:在目标位置之后]

• Body参数

参数	类型	是否必选	描述
obj	object	是	nat信息
obj.name	string	是	唯一标识名称 最小长度:1 最大长度:95
obj.enable	bool	否	是否启用 默认值:true
obj.position	int32	否	表示位置状态

obj.natType	string	是	nat类型 枚举值:[SNAT:转换类型为snat DNAT:转换类型为dnat BNAT:转换类型为bnat]
obj.description	string	否	描述 最小长度:0 最大长度:95
obj.schedule	string	否	生效时间 默认值:true
obj.snat	object	是	snat信息,策略类型为源转换时使用
obj.snat.srcZones	string[]	否	源区域
obj.snat.srcIpGroups	string[]	否	源ip组
obj.snat.dstNetobj	object	是	目的区域或者接口
obj.snat.dstNetobj.dstNetobjType	string	否	目的区域类型 默认值:true 枚举值:[ZONE:区域 INTERFACE:接口]
obj.snat.dstNetobj.interface	string	否	接口
obj.snat.dstNetobj.zone	string[]	否	区域
obj.snat.dstIpGroups	string[]	否	目的ip组
obj.snat.natService	string[]	否	引用的服务组
obj.snat.transfer	object	是	源转换相关数据
obj.snat.transfer.transferType	string	否	转换类型 默认值:true 枚举值:[OUTIF_IP:出接口ip IP_RANGE:ip范围 IP:单个ip IPGROUP:ip组 NO_TRANS:不转换]
obj.snat.transfer.ipRange	object	否	ip范围 格式:ip-range
obj.snat.transfer.ipRange.start	string	是	ip范围下界 格式:ip-address
obj.snat.transfer.ipRange.end	string	是	ip范围上界 格式:ip-address
obj.snat.transfer.ipRange.transferMode	string	否	转换模式 默认值:true 枚举值:[STATIC:静态转换 DYNAMIC:动态转换]

obj.snat.transfer.specifyIp	string	否	单个ip
obj.snat.transfer.ipGroups	string[]	否	ip组
obj.dnat	object	是	dnat信息,策略类型为目的转换时使用
obj.dnat.srcZones	string[]	否	源区域
obj.dnat.srcIpGroups	string[]	否	源ip组
obj.dnat.dstIpObj	object	是	目的ip组或者ip
obj.dnat.dstIpObj.dstIpObjType	string	否	目的类型 默认值:true 枚举值:[IPGROUP:ip组 IP:ip]
obj.dnat.dstIpObj.specifyIp	string[]	否	ip数组
obj.dnat.dstIpObj.ipGroups	string[]	否	ip组
obj.dnat.natService	string[]	否	引用的服务组
obj.dnat.transfer	object	是	目的转换相关数据
obj.dnat.transfer.transferType	string	否	转换类型 默认值:true 枚举值:[IP_RANGE:ip范围 IP:单个ip IPGROUP:ip组 NO_TRANS:不转换]
obj.dnat.transfer.dnatEnableScheduling	bool	否	dnat 使用负载均衡, 开资质才能生效
obj.dnat.transfer.ipRange	object	否	ip范围 格式:ip-range
obj.dnat.transfer.ipRange.start	string	是	ip范围下界 格式:ip-address
obj.dnat.transfer.ipRange.end	string	是	ip范围上界 格式:ip-address
obj.dnat.transfer.specifyIp	string	否	单个ip
obj.dnat.transfer.ipGroups	string[]	否	ip组
obj.dnat.transfer.transferPort	object[]	否	转换端口, 数组为空表示不转换端口
obj.dnat.transfer.transferPort.start	int32	是	网络端口起始值 最小值:0 最大值:65535

obj.dnat.transfer.transferPort.end	int32	否	网络端口结束值 最小值:0 最大值:65535
obj.dnat.dnatIgnoreAcl	bool	否	是否忽略访问控制 默认值:true
obj.bnat	object	是	bnat信息,策略类型为双向转换时使用
obj.bnat.srcZones	string[]	否	源区域
obj.bnat.srcIpGroups	string[]	否	源ip组
obj.bnat.dstIpobj	object	是	目的ip组或者ip
obj.bnat.dstIpobj.dstIpobjType	string	否	目的类型 默认值:true 枚举值:[IPGROUP:ip组 IP:ip]
obj.bnat.dstIpobj.specifyIp	string[]	否	单个ip
obj.bnat.dstIpobj.ipGroups	string[]	否	ip组
obj.bnat.natService	string[]	否	引用的服务组
obj.bnat.transferDst	object	是	目的转换相关数据
obj.bnat.transferDst.transferType	string	否	转换类型 默认值:true 枚举值:[IP_RANGE:ip范围 IP:单个ip IPGROUP:ip组 NO_TRANS:不转换]
obj.bnat.transferDst.dnatEnableScheduling	bool	否	dnat 使用负载均衡, 开资质才能生效
obj.bnat.transferDst.ipRange	object	否	ip范围 格式:ip-range
obj.bnat.transferDst.ipRange.start	string	是	ip范围下界 格式:ip-address
obj.bnat.transferDst.ipRange.end	string	是	ip范围上界 格式:ip-address
obj.bnat.transferDst.specifyIp	string	否	单个ip
obj.bnat.transferDst.ipGroups	string[]	否	ip组
obj.bnat.transferDst.transferPort	object[]	否	转换端口, 数组为空表示不转换端口
obj.bnat.transferDst.transferPort.start	int32	是	网络端口起始值 最小值:0 最大值:65535

obj.bnat.transferDst.transferPort.end	int32	否	网络端口结束值 最小值:0 最大值:65535
obj.bnat.dnatIgnoreAcl	bool	否	是否忽略访问控制 默认值:true
obj.bnat.transferSrc	object	是	源转换相关数据
obj.bnat.transferSrc.transferType	string	否	转换类型 默认值:true 枚举值:[OUTIF_IP:出接口ip IP_RANGE:ip范围 IP:单个ip IPGROUP:ip组 NO_TRANS:不转换]
obj.bnat.transferSrc.ipRange	object	否	ip范围 格式:ip-range
obj.bnat.transferSrc.ipRange.start	string	是	ip范围下界 格式:ip-address
obj.bnat.transferSrc.ipRange.end	string	是	ip范围上界 格式:ip-address
obj.bnat.transferSrc.ipRange.transferMode	string	否	转换模式 默认值:true 枚举值:[STATIC:静态转换 DYNAMIC:动态转换]
obj.bnat.transferSrc.specifyIp	string	否	单个ip
obj.bnat.transferSrc.ipGroups	string[]	否	ip组

请求样例:

增量修改一条NAT策略, 修改策略的启禁用字段

```
PATCH https://192.168.1.1/api/v1/namespaces/public/nats/test
{
  "enable": false
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	nat信息
data.uuid	string	唯一标识id

data.name	string	唯一标识名称 最小长度:1 最大长度:95
data.natHit	int64	nat策略匹配次数
data.enable	bool	是否启用
data.position	int32	表示位置状态
data.natType	string	nat类型 枚举值:[SNAT:转换类型为snat DNAT:转换类型为dnat BNAT:转换类型为bnat]
data.description	string	描述 最小长度:0 最大长度:95
data.schedule	string	生效时间
data.snat	object	snat信息,策略类型为源转换时使用
data.snat.srcZones	string[]	源区域
data.snat.srcIpGroups	string[]	源ip组
data.snat.dstNetobj	object	目的区域或者接口
data.snat.dstNetobj.dstNetobjType	string	目的区域类型 枚举值:[ZONE:区域 INTERFACE:接口]
data.snat.dstNetobj.interface	string	接口
data.snat.dstNetobj.zone	string[]	区域
data.snat.dstIpGroups	string[]	目的ip组
data.snat.natService	string[]	引用的服务组
data.snat.transfer	object	源转换相关数据
data.snat.transfer.transferType	string	转换类型 枚举值:[OUTIF_IP:出接口ip IP_RANGE:ip范围 IP:单个ip IPGROUP:ip组 NO_TRANS:不转换]
data.snat.transfer.ipRange	object	ip范围 格式:ip-range
data.snat.transfer.ipRange.start	string	ip范围下界 格式:ip-address
data.snat.transfer.ipRange.end	string	ip范围上界 格式:ip-address

data.snat.transfer.ipRange.transferMode	string	转换模式 枚举值:[STATIC:静态转换 DYNAMIC:动态转换]
data.snat.transfer.specifyIp	string	单个ip
data.snat.transfer.ipGroups	string[]	ip组
data.dnat	object	dnat信息,策略类型为目的转换时使用
data.dnat.srcZones	string[]	源区域
data.dnat.srcIpGroups	string[]	源ip组
data.dnat.dstIpobj	object	目的ip组或者ip
data.dnat.dstIpobj.dstIpobjType	string	目的类型 枚举值:[IPGROUP:ip组 IP:ip]
data.dnat.dstIpobj.specifyIp	string[]	ip数组
data.dnat.dstIpobj.ipGroups	string[]	ip组
data.dnat.natService	string[]	引用的服务组
data.dnat.transfer	object	目的转换相关数据
data.dnat.transfer.transferType	string	转换类型 枚举值:[IP_RANGE:ip范围 IP:单个ip IPGROUP:ip组 NO_TRANS:不转换]
data.dnat.transfer.dnatEnableScheduling	bool	dnat 使用负载均衡, 开资质才能生效
data.dnat.transfer.ipRange	object	ip范围 格式:ip-range
data.dnat.transfer.ipRange.start	string	ip范围下界 格式:ip-address
data.dnat.transfer.ipRange.end	string	ip范围上界 格式:ip-address
data.dnat.transfer.specifyIp	string	单个ip
data.dnat.transfer.ipGroups	string[]	ip组
data.dnat.transfer.transferPort	object[]	转换端口, 数组为空表示不转换端口

data.dnat.transfer.transferPort.start	int32	网络端口起始值 最小值:0 最大值:65535
data.dnat.transfer.transferPort.end	int32	网络端口结束值 最小值:0 最大值:65535
data.dnat.dnatIgnoreAcl	bool	是否忽略访问控制
data.bnat	object	bnat信息,策略类型为双向转换时使用
data.bnat.srcZones	string[]	源区域
data.bnat.srcIpGroups	string[]	源ip组
data.bnat.dstIpobj	object	目的ip组或者ip
data.bnat.dstIpobj.dstIpobjType	string	目的类型 枚举值:[IPGROUP:ip组 IP:ip]
data.bnat.dstIpobj.specifyIp	string[]	单个ip
data.bnat.dstIpobj.ipGroups	string[]	ip组
data.bnat.natService	string[]	引用的服务组
data.bnat.transferDst	object	目的转换相关数据
data.bnat.transferDst.transferType	string	转换类型 枚举值:[IP_RANGE:ip范围 IP:单个ip IPGROUP:ip组 NO_TRANS:不转换]
data.bnat.transferDst.dnatEnableScheduling	bool	dnat 使用负载均衡, 开资质才能生效
data.bnat.transferDst.ipRange	object	ip范围 格式:ip-range
data.bnat.transferDst.ipRange.start	string	ip范围下界 格式:ip-address
data.bnat.transferDst.ipRange.end	string	ip范围上界 格式:ip-address
data.bnat.transferDst.specifyIp	string	单个ip
data.bnat.transferDst.ipGroups	string[]	ip组
data.bnat.transferDst.transferPort	object[]	转换端口, 数组为空表示不转换端口
data.bnat.transferDst.transferPort.start	int32	网络端口起始值 最小值:0 最大值:65535

data.bnat.transferDst.transferPort.end	int32	网络端口结束值 最小值:0 最大值:65535
data.bnat.dnatIgnoreAcl	bool	是否忽略访问控制
data.bnat.transferSrc	object	源转换相关数据
data.bnat.transferSrc.transferType	string	转换类型 枚举值: OUTIF_IP:出接口ip IP_RANGE:ip范围 IP:单个ip IPGROUP:ip组 NO_TRANS:不转换]
data.bnat.transferSrc.ipRange	object	ip范围 格式:ip-range
data.bnat.transferSrc.ipRange.start	string	ip范围下界 格式:ip-address
data.bnat.transferSrc.ipRange.end	string	ip范围上界 格式:ip-address
data.bnat.transferSrc.ipRange.transferMode	string	转换模式 枚举值: STATIC:静态转换 DYNAMIC:动态转换]
data.bnat.transferSrc.specifyIp	string	单个ip
data.bnat.transferSrc.ipGroups	string[]	ip组

请求成功样例:

```
{
  "data": {
    "enable": false,
    "snat": {
      "srcIpGroups": [
        "ipgTest"
      ],
      "dstNetobj": {
        "zone": [
          "L3_trust_A"
        ],
        "dstNetobjType": "ZONE"
      },
      "natService": [
        "any"
      ],
      "dstIpGroups": [
        "ipgTest"
      ],
      "srcZones": [
        "L3_manage"
      ],
      "transfer": {
        "transferType": "OUTIF_IP"
      }
    },
    "schedule": "全天",
    "name": "test",
    "natType": "SNAT",
```

```

    "uuid": "7FB24191361F44D7881E5638A0914BB5",
    "description": ""
  },
  "code": 0,
  "message": "成功"
}

```

请求失败样例:

```

{
  "data": "",
  "code": 22,
  "message": "无效参数"
}

```

4.1.1.11 全量修改单条nat

功能说明: 全量修改单条nat

1) API请求数据

PUT <https://{host}/api/v1/namespaces/@namespace/nats/@name>

2) 输入参数

• Body参数

参数	类型	是否必选	描述
obj	object	是	nat信息
obj.name	string	是	唯一标识名称 最小长度:1 最大长度:95
obj.enable	bool	否	是否启用 默认值:true
obj.position	int32	否	表示位置状态
obj.natType	string	是	nat类型 枚举值:[SNAT:转换类型为snat DNAT:转换类型为dnat BNAT:转换类型为bnat]
obj.description	string	否	描述 最小长度:0 最大长度:95
obj.schedule	string	否	生效时间 默认值:true
obj.snat	object	是	snat信息,策略类型为源转换时使用
obj.snat.srcZones	string[]	否	源区域
obj.snat.srcIpGroups	string[]	否	源ip组
obj.snat.dstNetobj	object	是	目的区域或者接口
obj.snat.dstNetobj.dstNetobjType	string	否	目的区域类型 默认值:true 枚举值:[ZONE:区域 INTERFACE:接口]

obj.snat.dstNetobj.interface	string	否	接口
obj.snat.dstNetobj.zone	string[]	否	区域
obj.snat.dstIpGroups	string[]	否	目的ip组
obj.snat.natService	string[]	否	引用的服务组
obj.snat.transfer	object	是	源转换相关数据
obj.snat.transfer.transferType	string	否	转换类型 默认值:true 枚举值:[OUTIF_IP:出接口ip IP_RANGE:ip范围 IP:单个ip IPGROUP:ip组 NO_TRANS:不转换]
obj.snat.transfer.ipRange	object	否	ip范围 格式:ip-range
obj.snat.transfer.ipRange.start	string	是	ip范围下界 格式:ip-address
obj.snat.transfer.ipRange.end	string	是	ip范围上界 格式:ip-address
obj.snat.transfer.ipRange.transferMode	string	否	转换模式 默认值:true 枚举值:[STATIC:静态转换 DYNAMIC:动态转换]
obj.snat.transfer.specifyIp	string	否	单个ip
obj.snat.transfer.ipGroups	string[]	否	ip组
obj.dnat	object	是	dnat信息,策略类型为目的转换时使用
obj.dnat.srcZones	string[]	否	源区域
obj.dnat.srcIpGroups	string[]	否	源ip组
obj.dnat.dstIpobj	object	是	目的ip组或者ip
obj.dnat.dstIpobj.dstIpobjType	string	否	目的类型 默认值:true 枚举值:[IPGROUP:ip组 IP:ip]
obj.dnat.dstIpobj.specifyIp	string[]	否	ip数组
obj.dnat.dstIpobj.ipGroups	string[]	否	ip组
obj.dnat.natService	string[]	否	引用的服务组

obj.dnat.transfer	object	是	目的转换相关数据
obj.dnat.transfer.transferType	string	否	转换类型 默认值:true 枚举值:[IP_RANGE:ip范围 IP:单个ip IPGROUP:ip组 NO_TRANS:不转换]
obj.dnat.transfer.dnatEnableScheduling	bool	否	dnat 使用负载均衡, 开资质才能生效
obj.dnat.transfer.ipRange	object	否	ip范围 格式:ip-range
obj.dnat.transfer.ipRange.start	string	是	ip范围下界 格式:ip-address
obj.dnat.transfer.ipRange.end	string	是	ip范围上界 格式:ip-address
obj.dnat.transfer.specifyIp	string	否	单个ip
obj.dnat.transfer.ipGroups	string[]	否	ip组
obj.dnat.transfer.transferPort	object[]	否	转换端口, 数组为空表示不转换端口
obj.dnat.transfer.transferPort.start	int32	是	网络端口起始值 最小值:0 最大值:65535
obj.dnat.transfer.transferPort.end	int32	否	网络端口结束值 最小值:0 最大值:65535
obj.dnat.dnatIgnoreAcl	bool	否	是否忽略访问控制 默认值:true
obj.bnat	object	是	bnat信息,策略类型为双向转换时使用
obj.bnat.srcZones	string[]	否	源区域
obj.bnat.srcIpGroups	string[]	否	源ip组
obj.bnat.dstIpobj	object	是	目的ip组或者ip
obj.bnat.dstIpobj.dstIpobjType	string	否	目的类型 默认值:true 枚举值:[IPGROUP:ip组 IP:ip]
obj.bnat.dstIpobj.specifyIp	string[]	否	单个ip
obj.bnat.dstIpobj.ipGroups	string[]	否	ip组
obj.bnat.natService	string[]	否	引用的服务组

obj.bnat.transferDst	object	是	目的转换相关数据
obj.bnat.transferDst.transferType	string	否	转换类型 默认值:true 枚举值:[IP_RANGE:ip范围 IP:单个ip IPGROUP:ip组 NO_TRANS:不转换]
obj.bnat.transferDst.dnatEnableScheduling	bool	否	dnat 使用负载均衡, 开资质才能生效
obj.bnat.transferDst.ipRange	object	否	ip范围 格式:ip-range
obj.bnat.transferDst.ipRange.start	string	是	ip范围下界 格式:ip-address
obj.bnat.transferDst.ipRange.end	string	是	ip范围上界 格式:ip-address
obj.bnat.transferDst.specifyIp	string	否	单个ip
obj.bnat.transferDst.ipGroups	string[]	否	ip组
obj.bnat.transferDst.transferPort	object[]	否	转换端口, 数组为空表示不转换端口
obj.bnat.transferDst.transferPort.start	int32	是	网络端口起始值 最小值:0 最大值:65535
obj.bnat.transferDst.transferPort.end	int32	否	网络端口结束值 最小值:0 最大值:65535
obj.bnat.dnatIgnoreAcl	bool	否	是否忽略访问控制 默认值:true
obj.bnat.transferSrc	object	是	源转换相关数据
obj.bnat.transferSrc.transferType	string	否	转换类型 默认值:true 枚举值:[OUTIF_IP:出接口ip IP_RANGE:ip范围 IP:单个ip IPGROUP:ip组 NO_TRANS:不转换]
obj.bnat.transferSrc.ipRange	object	否	ip范围 格式:ip-range
obj.bnat.transferSrc.ipRange.start	string	是	ip范围下界 格式:ip-address
obj.bnat.transferSrc.ipRange.end	string	是	ip范围上界 格式:ip-address

obj.bnat.transferSrc.ipRange.transferMode	string	否	转换模式 默认值:true 枚举值:[STATIC:静态转换 DYNAMIC:动态转换]
obj.bnat.transferSrc.specifyIp	string	否	单个ip
obj.bnat.transferSrc.ipGroups	string[]	否	ip组

请求样例:

全量修改单条NAT策略

```
PUT https://192.168.1.1/api/v1/namespaces/public/nats/test
{
  "enable": true,
  "position": 1,
  "snat": {
    "srcIpGroups": [
      "ipgTest"
    ],
    "dstNetobj": {
      "zone": [
        "L3_trust_A"
      ],
      "dstNetobjType": "ZONE"
    },
    "natService": [
      "any"
    ],
    "dstIpGroups": [
      "ipgTest"
    ],
    "srcZones": [
      "L3_manage"
    ],
    "transfer": {
      "transferType": "OUTIF_IP"
    }
  },
  "schedule": "全天",
  "natType": "SNAT",
  "name": "test",
  "description": "12we"
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	nat信息
data.uuid	string	唯一标识id
data.name	string	唯一标识名称 最小长度:1 最大长度:95
data.natHit	int64	nat策略匹配次数
data.enable	bool	是否启用

data.position	int32	表示位置状态
data.natType	string	nat类型 枚举值:[SNAT:转换类型为snat DNAT:转换类型为dnat BNAT:转换类型为bnat]
data.description	string	描述 最小长度:0 最大长度:95
data.schedule	string	生效时间
data.snat	object	snat信息,策略类型为源转换时使用
data.snat.srcZones	string[]	源区域
data.snat.srcIpGroups	string[]	源ip组
data.snat.dstNetobj	object	目的区域或者接口
data.snat.dstNetobj.dstNetobjType	string	目的区域类型 枚举值:[ZONE:区域 INTERFACE:接口]
data.snat.dstNetobj.interface	string	接口
data.snat.dstNetobj.zone	string[]	区域
data.snat.dstIpGroups	string[]	目的ip组
data.snat.natService	string[]	引用的服务组
data.snat.transfer	object	源转换相关数据
data.snat.transfer.transferType	string	转换类型 枚举值:[OUTIF_IP:出接口ip IP_RANGE:ip范围 IP:单个ip IPGROUP:ip组 NO_TRANS:不转换]
data.snat.transfer.ipRange	object	ip范围 格式:ip-range
data.snat.transfer.ipRange.start	string	ip范围下界 格式:ip-address
data.snat.transfer.ipRange.end	string	ip范围上界 格式:ip-address
data.snat.transfer.ipRange.transferMode	string	转换模式 枚举值:[STATIC:静态转换 DYNAMIC:动态转换]

data.snat.transfer.specifyIp	string	单个ip
data.snat.transfer.ipGroups	string[]	ip组
data.dnat	object	dnat信息,策略类型为目的转换时使用
data.dnat.srcZones	string[]	源区域
data.dnat.srcIpGroups	string[]	源ip组
data.dnat.dstIpobj	object	目的ip组或者ip
data.dnat.dstIpobj.dstIpobjType	string	目的类型 枚举值:[IPGROUP:ip组 IP:ip]
data.dnat.dstIpobj.specifyIp	string[]	ip数组
data.dnat.dstIpobj.ipGroups	string[]	ip组
data.dnat.natService	string[]	引用的服务组
data.dnat.transfer	object	目的转换相关数据
data.dnat.transfer.transferType	string	转换类型 枚举值:[IP_RANGE:ip范围 IP:单个ip IPGROUP:ip组 NO_TRANS:不转换]
data.dnat.transfer.dnatEnableScheduling	bool	dnat 使用负载均衡, 开资质才能生效
data.dnat.transfer.ipRange	object	ip范围 格式:ip-range
data.dnat.transfer.ipRange.start	string	ip范围下界 格式:ip-address
data.dnat.transfer.ipRange.end	string	ip范围上界 格式:ip-address
data.dnat.transfer.specifyIp	string	单个ip
data.dnat.transfer.ipGroups	string[]	ip组
data.dnat.transfer.transferPort	object[]	转换端口, 数组为空表示不转换端口
data.dnat.transfer.transferPort.start	int32	网络端口起始值 最小值:0 最大值:65535
data.dnat.transfer.transferPort.end	int32	网络端口结束值 最小值:0 最大值:65535

data.dnat.dnatIgnoreAcl	bool	是否忽略访问控制
data.bnat	object	bnat信息,策略类型为双向转换时使用
data.bnat.srcZones	string[]	源区域
data.bnat.srcIpGroups	string[]	源ip组
data.bnat.dstIpobj	object	目的ip组或者ip
data.bnat.dstIpobj.dstIpobjType	string	目的类型 枚举值:[IPGROUP:ip组 IP:ip]
data.bnat.dstIpobj.specifyIp	string[]	单个ip
data.bnat.dstIpobj.ipGroups	string[]	ip组
data.bnat.natService	string[]	引用的服务组
data.bnat.transferDst	object	目的转换相关数据
data.bnat.transferDst.transferType	string	转换类型 枚举值:[IP_RANGE:ip范围 IP:单个ip IPGROUP:ip组 NO_TRANS:不转换]
data.bnat.transferDst.dnatEnableScheduling	bool	dnat 使用负载均衡, 开资质才能生效
data.bnat.transferDst.ipRange	object	ip范围 格式:ip-range
data.bnat.transferDst.ipRange.start	string	ip范围下界 格式:ip-address
data.bnat.transferDst.ipRange.end	string	ip范围上界 格式:ip-address
data.bnat.transferDst.specifyIp	string	单个ip
data.bnat.transferDst.ipGroups	string[]	ip组
data.bnat.transferDst.transferPort	object[]	转换端口, 数组为空表示不转换端口
data.bnat.transferDst.transferPort.start	int32	网络端口起始值 最小值:0 最大值:65535
data.bnat.transferDst.transferPort.end	int32	网络端口结束值 最小值:0 最大值:65535
data.bnat.dnatIgnoreAcl	bool	是否忽略访问控制
data.bnat.transferSrc	object	源转换相关数据

data.bnat.transferSrc.transferType	string	转换类型 枚举值:[OUTIF_IP:出接口ip IP_RANGE:ip范围 IP:单个ip IPGROUP:ip组 NO_TRANS:不转换]
data.bnat.transferSrc.ipRange	object	ip范围 格式:ip-range
data.bnat.transferSrc.ipRange.start	string	ip范围下界 格式:ip-address
data.bnat.transferSrc.ipRange.end	string	ip范围上界 格式:ip-address
data.bnat.transferSrc.ipRange.transferMode	string	转换模式 枚举值:[STATIC:静态转换 DYNAMIC:动态转换]
data.bnat.transferSrc.specifyIp	string	单个ip
data.bnat.transferSrc.ipGroups	string[]	ip组

请求成功样例:

```
{
  "data": {
    "enable": true,
    "snat": {
      "srcIpGroups": [
        "ipgTest"
      ],
      "dstNetobj": {
        "zone": [
          "L3_trust_A"
        ],
        "dstNetobjType": "ZONE"
      },
      "natService": [
        "any"
      ],
      "dstIpGroups": [
        "ipgTest"
      ],
      "srcZones": [
        "L3_manage"
      ],
      "transfer": {
        "transferType": "OUTIF_IP"
      }
    },
    "schedule": "全天",
    "name": "test",
    "natType": "SNAT",
    "uuid": "7FB24191361F44D7881E5638A0914BB5",
    "description": "12we"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.1.1.12 批量删除nat对象

功能说明：批量删除nat策略

1) API请求数据

POST https://{host}/api/batch/v1/namespaces/@namespace/nats?_method=delete

2) 输入参数

• Body参数

参数	类型	是否必选	描述
obj	object[]	是	nat列表
obj.name	string	是	唯一标识名称 最小长度:1 最大长度:95
obj.enable	bool	否	是否启用 默认值:true
obj.position	int32	否	表示位置状态
obj.natType	string	是	nat类型 枚举值:[SNAT:转换类型为snat DNAT:转换类型为dnat BNAT:转换类型为bnat]
obj.description	string	否	描述 最小长度:0 最大长度:95
obj.schedule	string	否	生效时间 默认值:true
obj.snat	object	是	snat信息,策略类型为源转换时使用
obj.snat.srcZones	string[]	否	源区域
obj.snat.srcIpGroups	string[]	否	源ip组
obj.snat.dstNetobj	object	是	目的区域或者接口
obj.snat.dstNetobj.dstNetobjType	string	否	目的区域类型 默认值:true 枚举值:[ZONE:区域 INTERFACE:接口]
obj.snat.dstNetobj.interface	string	否	接口
obj.snat.dstNetobj.zone	string[]	否	区域
obj.snat.dstIpGroups	string[]	否	目的ip组

obj.snat.natService	string[]	否	引用的服务组
obj.snat.transfer	object	是	源转换相关数据
obj.snat.transfer.transferType	string	否	转换类型 默认值:true 枚举值:[OUTIF_IP:出接口ip IP_RANGE:ip范围 IP:单个ip IPGROUP:ip组 NO_TRANS:不转换]
obj.snat.transfer.ipRange	object	否	ip范围 格式:ip-range
obj.snat.transfer.ipRange.start	string	是	ip范围下界 格式:ip-address
obj.snat.transfer.ipRange.end	string	是	ip范围上界 格式:ip-address
obj.snat.transfer.ipRange.transferMode	string	否	转换模式 默认值:true 枚举值:[STATIC:静态转换 DYNAMIC:动态转换]
obj.snat.transfer.specifyIp	string	否	单个ip
obj.snat.transfer.ipGroups	string[]	否	ip组
obj.dnat	object	是	dnat信息,策略类型为目的转换时使用
obj.dnat.srcZones	string[]	否	源区域
obj.dnat.srcIpGroups	string[]	否	源ip组
obj.dnat.dstIpobj	object	是	目的ip组或者ip
obj.dnat.dstIpobj.dstIpobjType	string	否	目的类型 默认值:true 枚举值:[IPGROUP:ip组 IP:ip]
obj.dnat.dstIpobj.specifyIp	string[]	否	ip数组
obj.dnat.dstIpobj.ipGroups	string[]	否	ip组
obj.dnat.natService	string[]	否	引用的服务组
obj.dnat.transfer	object	是	目的转换相关数据

obj.dnat.transfer.transferType	string	否	转换类型 默认值:true 枚举值:[IP_RANGE:ip范围 IP:单个ip IPGROUP:ip组 NO_TRANS:不转换]
obj.dnat.transfer.dnatEnableScheduling	bool	否	dnat 使用负载均衡, 开资质才能生效
obj.dnat.transfer.ipRange	object	否	ip范围 格式:ip-range
obj.dnat.transfer.ipRange.start	string	是	ip范围下界 格式:ip-address
obj.dnat.transfer.ipRange.end	string	是	ip范围上界 格式:ip-address
obj.dnat.transfer.specifyIp	string	否	单个ip
obj.dnat.transfer.ipGroups	string[]	否	ip组
obj.dnat.transfer.transferPort	object[]	否	转换端口, 数组为空表示不转换端口
obj.dnat.transfer.transferPort.start	int32	是	网络端口起始值 最小值:0 最大值:65535
obj.dnat.transfer.transferPort.end	int32	否	网络端口结束值 最小值:0 最大值:65535
obj.dnat.dnatIgnoreAcl	bool	否	是否忽略访问控制 默认值:true
obj.bnat	object	是	bnat信息,策略类型为双向转换时使用
obj.bnat.srcZones	string[]	否	源区域
obj.bnat.srcIpGroups	string[]	否	源ip组
obj.bnat.dstIpObj	object	是	目的ip组或者ip
obj.bnat.dstIpObj.dstIpObjType	string	否	目的类型 默认值:true 枚举值:[IPGROUP:ip组 IP:ip]
obj.bnat.dstIpObj.specifyIp	string[]	否	单个ip
obj.bnat.dstIpObj.ipGroups	string[]	否	ip组
obj.bnat.natService	string[]	否	引用的服务组
obj.bnat.transferDst	object	是	目的转换相关数据

obj.bnat.transferDst.transferType	string	否	转换类型 默认值:true 枚举值:[IP_RANGE:ip范围 IP:单个ip IPGROUP:ip组 NO_TRANS:不转换]
obj.bnat.transferDst.dnatEnableScheduling	bool	否	dnat 使用负载均衡, 开资质才能生效
obj.bnat.transferDst.ipRange	object	否	ip范围 格式:ip-range
obj.bnat.transferDst.ipRange.start	string	是	ip范围下界 格式:ip-address
obj.bnat.transferDst.ipRange.end	string	是	ip范围上界 格式:ip-address
obj.bnat.transferDst.specifyIp	string	否	单个ip
obj.bnat.transferDst.ipGroups	string[]	否	ip组
obj.bnat.transferDst.transferPort	object[]	否	转换端口, 数组为空表示不转换端口
obj.bnat.transferDst.transferPort.start	int32	是	网络端口起始值 最小值:0 最大值:65535
obj.bnat.transferDst.transferPort.end	int32	否	网络端口结束值 最小值:0 最大值:65535
obj.bnat.dnatIgnoreAcl	bool	否	是否忽略访问控制 默认值:true
obj.bnat.transferSrc	object	是	源转换相关数据
obj.bnat.transferSrc.transferType	string	否	转换类型 默认值:true 枚举值:[OUTIF_IP:出接口ip IP_RANGE:ip范围 IP:单个ip IPGROUP:ip组 NO_TRANS:不转换]
obj.bnat.transferSrc.ipRange	object	否	ip范围 格式:ip-range
obj.bnat.transferSrc.ipRange.start	string	是	ip范围下界 格式:ip-address
obj.bnat.transferSrc.ipRange.end	string	是	ip范围上界 格式:ip-address

obj.bnat.transferSrc.ipRange.transferMode	string	否	转换模式 默认值:true 枚举值:[STATIC:静态转换 DYNAMIC:动态转换]
obj.bnat.transferSrc.specifyIp	string	否	单个ip
obj.bnat.transferSrc.ipGroups	string[]	否	ip组

请求样例:

批量删除NAT策略

POST https://192.168.1.1/api/batch/v1/namespaces/public/nats?_method=DELETE

```
[
  {
    "name": "tes4"
  },
  {
    "name": "test2"
  }
]
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	nat列表
data.uuid	string	唯一标识id
data.name	string	唯一标识名称 最小长度:1 最大长度:95
data.natHit	int64	nat策略匹配次数
data.enable	bool	是否启用
data.position	int32	表示位置状态
data.natType	string	nat类型 枚举值:[SNAT:转换类型为snat DNAT:转换类型为dnat BNAT:转换类型为bnat]
data.description	string	描述 最小长度:0 最大长度:95
data.schedule	string	生效时间
data.snat	object	snat信息,策略类型为源转换时使用
data.snat.srcZones	string[]	源区域

data.snat.srcIpGroups	string[]	源ip组
data.snat.dstNetobj	object	目的区域或者接口
data.snat.dstNetobj.dstNetobjType	string	目的区域类型 枚举值:[ZONE:区域 INTERFACE:接口]
data.snat.dstNetobj.interface	string	接口
data.snat.dstNetobj.zone	string[]	区域
data.snat.dstIpGroups	string[]	目的ip组
data.snat.natService	string[]	引用的服务组
data.snat.transfer	object	源转换相关数据
data.snat.transfer.transferType	string	转换类型 枚举值:[OUTIF_IP:出接口ip IP_RANGE:ip范围 IP:单个ip IPGROUP:ip组 NO_TRANS:不转换]
data.snat.transfer.ipRange	object	ip范围 格式:ip-range
data.snat.transfer.ipRange.start	string	ip范围下界 格式:ip-address
data.snat.transfer.ipRange.end	string	ip范围上界 格式:ip-address
data.snat.transfer.ipRange.transferMode	string	转换模式 枚举值:[STATIC:静态转换 DYNAMIC:动态转换]
data.snat.transfer.specifyIp	string	单个ip
data.snat.transfer.ipGroups	string[]	ip组
data.dnat	object	dnat信息,策略类型为目的转换时使用
data.dnat.srcZones	string[]	源区域
data.dnat.srcIpGroups	string[]	源ip组
data.dnat.dstIpobj	object	目的ip组或者ip
data.dnat.dstIpobj.dstIpobjType	string	目的类型 枚举值:[IPGROUP:ip组 IP:ip]

data.dnat.dstIpobj.specifyIp	string[]	ip数组
data.dnat.dstIpobj.ipGroups	string[]	ip组
data.dnat.natService	string[]	引用的服务组
data.dnat.transfer	object	目的转换相关数据
data.dnat.transfer.transferType	string	转换类型 枚举值:[IP_RANGE:ip范围 IP:单个ip IPGROUP:ip组 NO_TRANS:不转换]
data.dnat.transfer.dnatEnableScheduling	bool	dnat 使用负载均衡, 开资质才能生效
data.dnat.transfer.ipRange	object	ip范围 格式:ip-range
data.dnat.transfer.ipRange.start	string	ip范围下界 格式:ip-address
data.dnat.transfer.ipRange.end	string	ip范围上界 格式:ip-address
data.dnat.transfer.specifyIp	string	单个ip
data.dnat.transfer.ipGroups	string[]	ip组
data.dnat.transfer.transferPort	object[]	转换端口, 数组为空表示不转换端口
data.dnat.transfer.transferPort.start	int32	网络端口起始值 最小值:0 最大值:65535
data.dnat.transfer.transferPort.end	int32	网络端口结束值 最小值:0 最大值:65535
data.dnat.dnatIgnoreAcl	bool	是否忽略访问控制
data.bnat	object	bnat信息,策略类型为双向转换时使用
data.bnat.srcZones	string[]	源区域
data.bnat.srcIpGroups	string[]	源ip组
data.bnat.dstIpobj	object	目的ip组或者ip
data.bnat.dstIpobj.dstIpobjType	string	目的类型 枚举值:[IPGROUP:ip组 IP:ip]
data.bnat.dstIpobj.specifyIp	string[]	单个ip

data.bnat.dstIpObj.ipGroups	string[]	ip组
data.bnat.natService	string[]	引用的服务组
data.bnat.transferDst	object	目的转换相关数据
data.bnat.transferDst.transferType	string	转换类型 枚举值: IP_RANGE:ip范围 IP:单个ip IPGROUP:ip组 NO_TRANS:不转换]
data.bnat.transferDst.dnatEnableScheduling	bool	dnat 使用负载均衡, 开资质才能生效
data.bnat.transferDst.ipRange	object	ip范围 格式:ip-range
data.bnat.transferDst.ipRange.start	string	ip范围下界 格式:ip-address
data.bnat.transferDst.ipRange.end	string	ip范围上界 格式:ip-address
data.bnat.transferDst.specifyIp	string	单个ip
data.bnat.transferDst.ipGroups	string[]	ip组
data.bnat.transferDst.transferPort	object[]	转换端口, 数组为空表示不转换端口
data.bnat.transferDst.transferPort.start	int32	网络端口起始值 最小值:0 最大值:65535
data.bnat.transferDst.transferPort.end	int32	网络端口结束值 最小值:0 最大值:65535
data.bnat.dnatIgnoreAcl	bool	是否忽略访问控制
data.bnat.transferSrc	object	源转换相关数据
data.bnat.transferSrc.transferType	string	转换类型 枚举值: OUTIF_IP:出接口ip IP_RANGE:ip范围 IP:单个ip IPGROUP:ip组 NO_TRANS:不转换]
data.bnat.transferSrc.ipRange	object	ip范围 格式:ip-range
data.bnat.transferSrc.ipRange.start	string	ip范围下界 格式:ip-address
data.bnat.transferSrc.ipRange.end	string	ip范围上界 格式:ip-address

data.bnat.transferSrc.ip Range.transferMode	string	转换模式 枚举值: STATIC:静态转换 DYNAMIC:动态转换]
data.bnat.transferSrc.s pecifyIp	string	单个ip
data.bnat.transferSrc.ip Groups	string[]	ip组

请求成功样例:

```
{
  "data": [
    {
      "enable": true,
      "snat": {
        "srcIpGroups": [
          "ipgTest"
        ],
        "dstNetobj": {
          "zone": [
            "L3_trust_A"
          ],
          "dstNetobjType": "ZONE"
        },
        "natService": [
          "any"
        ],
        "dstIpGroups": [
          "ipgTest"
        ],
        "srcZones": [
          "L3_manage"
        ],
        "transfer": {
          "transferType": "OUTIF_IP"
        }
      },
      "schedule": "全天",
      "name": "tes4",
      "natType": "SNAT",
      "uuid": "EB0D9F4A291242C8AA778487E3FCE86D",
      "description": "12we"
    },
    {
      "enable": true,
      "snat": {
        "srcIpGroups": [
          "ipgTest"
        ],
        "dstNetobj": {
          "zone": [
            "L3_trust_A"
          ],
          "dstNetobjType": "ZONE"
        },
        "natService": [
          "any"
        ],
        "dstIpGroups": [
          "ipgTest"
        ],
        "srcZones": [
          "L3_manage"
        ],
        "transfer": {
          "transferType": "OUTIF_IP"
        }
      },
      "schedule": "全天",
      "name": "test2",
    }
  ]
}
```

```

    "natType": "SNAT",
    "uuid": "6ADE57EAC51C4759B027D8A904E4ADDF",
    "description": "12we"
  },
  "code": 0,
  "message": "成功"
}

```

请求失败样例:

```

{
  "data": "",
  "code": 22,
  "message": "无效参数"
}

```

4.1.1.13 批量增量更新nat对象

功能说明: 批量增量更新nat对象, 主要用于移动启禁

1) API请求数据

PATCH <https://{host}/api/batch/v1/namespaces/@namespace/nats>

2) 输入参数

• URL参数

参数	类型	是否必选	描述
_where	string	否	指定位置, 用于移动/插入 枚举值:[top:顶部 bottom:底部 before:在目标位置之前 after:在目标位置之后]
_dest	string	否	目标位置关键字对应的值 最大长度:100
_key	string	否	目标位置关键字 枚举值:[position:指定目标位置关键字为 position name:指定目标位置关键字为 name]

• Body参数

参数	类型	是否必选	描述
obj	object[]	是	nat列表
obj.name	string	是	唯一标识名称 最小长度:1 最大长度:95
obj.enable	bool	否	是否启用 默认值:true
obj.position	int32	否	表示位置状态

obj.natType	string	是	nat类型 枚举值:[SNAT:转换类型为snat DNAT:转换类型为dnat BNAT:转换类型为bnat]
obj.description	string	否	描述 最小长度:0 最大长度:95
obj.schedule	string	否	生效时间 默认值:true
obj.snat	object	是	snat信息,策略类型为源转换时使用
obj.snat.srcZones	string[]	否	源区域
obj.snat.srcIpGroups	string[]	否	源ip组
obj.snat.dstNetobj	object	是	目的区域或者接口
obj.snat.dstNetobj.dstNetobjType	string	否	目的区域类型 默认值:true 枚举值:[ZONE:区域 INTERFACE:接口]
obj.snat.dstNetobj.interface	string	否	接口
obj.snat.dstNetobj.zone	string[]	否	区域
obj.snat.dstIpGroups	string[]	否	目的ip组
obj.snat.natService	string[]	否	引用的服务组
obj.snat.transfer	object	是	源转换相关数据
obj.snat.transfer.transferType	string	否	转换类型 默认值:true 枚举值:[OUTIF_IP:出接口ip IP_RANGE:ip范围 IP:单个ip IPGROUP:ip组 NO_TRANS:不转换]
obj.snat.transfer.ipRange	object	否	ip范围 格式:ip-range
obj.snat.transfer.ipRange.start	string	是	ip范围下界 格式:ip-address
obj.snat.transfer.ipRange.end	string	是	ip范围上界 格式:ip-address
obj.snat.transfer.ipRange.transferMode	string	否	转换模式 默认值:true 枚举值:[STATIC:静态转换 DYNAMIC:动态转换]

obj.snat.transfer.specifyIp	string	否	单个ip
obj.snat.transfer.ipGroups	string[]	否	ip组
obj.dnat	object	是	dnat信息,策略类型为目的转换时使用
obj.dnat.srcZones	string[]	否	源区域
obj.dnat.srcIpGroups	string[]	否	源ip组
obj.dnat.dstIpObj	object	是	目的ip组或者ip
obj.dnat.dstIpObj.dstIpObjType	string	否	目的类型 默认值:true 枚举值:[IPGROUP:ip组 IP:ip]
obj.dnat.dstIpObj.specifyIp	string[]	否	ip数组
obj.dnat.dstIpObj.ipGroups	string[]	否	ip组
obj.dnat.natService	string[]	否	引用的服务组
obj.dnat.transfer	object	是	目的转换相关数据
obj.dnat.transfer.transferType	string	否	转换类型 默认值:true 枚举值:[IP_RANGE:ip范围 IP:单个ip IPGROUP:ip组 NO_TRANS:不转换]
obj.dnat.transfer.dnatEnableScheduling	bool	否	dnat 使用负载均衡, 开资质才能生效
obj.dnat.transfer.ipRange	object	否	ip范围 格式:ip-range
obj.dnat.transfer.ipRange.start	string	是	ip范围下界 格式:ip-address
obj.dnat.transfer.ipRange.end	string	是	ip范围上界 格式:ip-address
obj.dnat.transfer.specifyIp	string	否	单个ip
obj.dnat.transfer.ipGroups	string[]	否	ip组
obj.dnat.transfer.transferPort	object[]	否	转换端口, 数组为空表示不转换端口
obj.dnat.transfer.transferPort.start	int32	是	网络端口起始值 最小值:0 最大值:65535

obj.dnat.transfer.transferPort.end	int32	否	网络端口结束值 最小值:0 最大值:65535
obj.dnat.dnatIgnoreAcl	bool	否	是否忽略访问控制 默认值:true
obj.bnat	object	是	bnat信息,策略类型为双向转换时使用
obj.bnat.srcZones	string[]	否	源区域
obj.bnat.srcIpGroups	string[]	否	源ip组
obj.bnat.dstIpobj	object	是	目的ip组或者ip
obj.bnat.dstIpobj.dstIpobjType	string	否	目的类型 默认值:true 枚举值:[IPGROUP:ip组 IP:ip]
obj.bnat.dstIpobj.specifyIp	string[]	否	单个ip
obj.bnat.dstIpobj.ipGroups	string[]	否	ip组
obj.bnat.natService	string[]	否	引用的服务组
obj.bnat.transferDst	object	是	目的转换相关数据
obj.bnat.transferDst.transferType	string	否	转换类型 默认值:true 枚举值:[IP_RANGE:ip范围 IP:单个ip IPGROUP:ip组 NO_TRANS:不转换]
obj.bnat.transferDst.dnatEnableScheduling	bool	否	dnat 使用负载均衡, 开资质才能生效
obj.bnat.transferDst.ipRange	object	否	ip范围 格式:ip-range
obj.bnat.transferDst.ipRange.start	string	是	ip范围下界 格式:ip-address
obj.bnat.transferDst.ipRange.end	string	是	ip范围上界 格式:ip-address
obj.bnat.transferDst.specifyIp	string	否	单个ip
obj.bnat.transferDst.ipGroups	string[]	否	ip组
obj.bnat.transferDst.transferPort	object[]	否	转换端口, 数组为空表示不转换端口
obj.bnat.transferDst.transferPort.start	int32	是	网络端口起始值 最小值:0 最大值:65535

obj.bnat.transferDst.transferPort.end	int32	否	网络端口结束值 最小值:0 最大值:65535
obj.bnat.dnatIgnoreAcl	bool	否	是否忽略访问控制 默认值:true
obj.bnat.transferSrc	object	是	源转换相关数据
obj.bnat.transferSrc.transferType	string	否	转换类型 默认值:true 枚举值:[OUTIF_IP:出接口ip IP_RANGE:ip范围 IP:单个ip IPGROUP:ip组 NO_TRANS:不转换]
obj.bnat.transferSrc.ipRange	object	否	ip范围 格式:ip-range
obj.bnat.transferSrc.ipRange.start	string	是	ip范围下界 格式:ip-address
obj.bnat.transferSrc.ipRange.end	string	是	ip范围上界 格式:ip-address
obj.bnat.transferSrc.ipRange.transferMode	string	否	转换模式 默认值:true 枚举值:[STATIC:静态转换 DYNAMIC:动态转换]
obj.bnat.transferSrc.specifyIp	string	否	单个ip
obj.bnat.transferSrc.ipGroups	string[]	否	ip组

请求样例:

```
批量增量更新NAT策略,对一批策略禁用
PATCH https://192.168.1.1/api/batch/v1/namespaces/public/nats
[
  {
    "name": "test2",
    "enable": false
  },
  {
    "name": "tes1",
    "enable": false
  }
]
```

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息

data	object[]	nat列表
data.uuid	string	唯一标识id
data.name	string	唯一标识名称 最小长度:1 最大长度:95
data.natHit	int64	nat策略匹配次数
data.enable	bool	是否启用
data.position	int32	表示位置状态
data.natType	string	nat类型 枚举值:[SNAT:转换类型为snat DNAT:转换类型为dnat BNAT:转换类型为bnat]
data.description	string	描述 最小长度:0 最大长度:95
data.schedule	string	生效时间
data.snat	object	snat信息,策略类型为源转换时使用
data.snat.srcZones	string[]	源区域
data.snat.srcIpGroups	string[]	源ip组
data.snat.dstNetobj	object	目的区域或者接口
data.snat.dstNetobj.dstNetobjType	string	目的区域类型 枚举值:[ZONE:区域 INTERFACE:接口]
data.snat.dstNetobj.interface	string	接口
data.snat.dstNetobj.zone	string[]	区域
data.snat.dstIpGroups	string[]	目的ip组
data.snat.natService	string[]	引用的服务组
data.snat.transfer	object	源转换相关数据
data.snat.transfer.transferType	string	转换类型 枚举值:[OUTIF_IP:出接口ip IP_RANGE:ip范围 IP:单个ip IPGROUP:ip组 NO_TRANS:不转换]
data.snat.transfer.ipRange	object	ip范围 格式:ip-range
data.snat.transfer.ipRange.start	string	ip范围下界 格式:ip-address

data.snat.transfer.ipRange.end	string	ip范围上界 格式:ip-address
data.snat.transfer.ipRange.transferMode	string	转换模式 枚举值:[STATIC:静态转换 DYNAMIC:动态转换]
data.snat.transfer.specifyIp	string	单个ip
data.snat.transfer.ipGroups	string[]	ip组
data.dnat	object	dnat信息,策略类型为目的转换时使用
data.dnat.srcZones	string[]	源区域
data.dnat.srcIpGroups	string[]	源ip组
data.dnat.dstIpObj	object	目的ip组或者ip
data.dnat.dstIpObj.dstIpObjType	string	目的类型 枚举值:[IPGROUP:ip组 IP:ip]
data.dnat.dstIpObj.specifyIp	string[]	ip数组
data.dnat.dstIpObj.ipGroups	string[]	ip组
data.dnat.natService	string[]	引用的服务组
data.dnat.transfer	object	目的转换相关数据
data.dnat.transfer.transferType	string	转换类型 枚举值:[IP_RANGE:ip范围 IP:单个ip IPGROUP:ip组 NO_TRANS:不转换]
data.dnat.transfer.dnatEnableScheduling	bool	dnat 使用负载均衡, 开资质才能生效
data.dnat.transfer.ipRange	object	ip范围 格式:ip-range
data.dnat.transfer.ipRange.start	string	ip范围下界 格式:ip-address
data.dnat.transfer.ipRange.end	string	ip范围上界 格式:ip-address
data.dnat.transfer.specifyIp	string	单个ip
data.dnat.transfer.ipGroups	string[]	ip组

data.dnat.transfer.transferPort	object[]	转换端口, 数组为空表示不转换端口
data.dnat.transfer.transferPort.start	int32	网络端口起始值 最小值:0 最大值:65535
data.dnat.transfer.transferPort.end	int32	网络端口结束值 最小值:0 最大值:65535
data.dnat.dnatIgnoreAcl	bool	是否忽略访问控制
data.bnat	object	bnat信息,策略类型为双向转换时使用
data.bnat.srcZones	string[]	源区域
data.bnat.srcIpGroups	string[]	源ip组
data.bnat.dstIpobj	object	目的ip组或者ip
data.bnat.dstIpobj.dstIpobjType	string	目的类型 枚举值:[IPGROUP:ip组 IP:ip]
data.bnat.dstIpobj.specifyIp	string[]	单个ip
data.bnat.dstIpobj.ipGroups	string[]	ip组
data.bnat.natService	string[]	引用的服务组
data.bnat.transferDst	object	目的转换相关数据
data.bnat.transferDst.transferType	string	转换类型 枚举值:[IP_RANGE:ip范围 IP:单个ip IPGROUP:ip组 NO_TRANS:不转换]
data.bnat.transferDst.dnatEnableScheduling	bool	dnat 使用负载均衡, 开资质才能生效
data.bnat.transferDst.ipRange	object	ip范围 格式:ip-range
data.bnat.transferDst.ipRange.start	string	ip范围下界 格式:ip-address
data.bnat.transferDst.ipRange.end	string	ip范围上界 格式:ip-address
data.bnat.transferDst.specifyIp	string	单个ip
data.bnat.transferDst.ipGroups	string[]	ip组
data.bnat.transferDst.transferPort	object[]	转换端口, 数组为空表示不转换端口

data.bnat.transferDst.transferPort.start	int32	网络端口起始值 最小值:0 最大值:65535
data.bnat.transferDst.transferPort.end	int32	网络端口结束值 最小值:0 最大值:65535
data.bnat.dnatIgnoreAcl	bool	是否忽略访问控制
data.bnat.transferSrc	object	源转换相关数据
data.bnat.transferSrc.transferType	string	转换类型 枚举值: OUTIF_IP:出接口ip IP_RANGE:ip范围 IP:单个ip IPGROUP:ip组 NO_TRANS:不转换]
data.bnat.transferSrc.ipRange	object	ip范围 格式:ip-range
data.bnat.transferSrc.ipRange.start	string	ip范围下界 格式:ip-address
data.bnat.transferSrc.ipRange.end	string	ip范围上界 格式:ip-address
data.bnat.transferSrc.ipRange.transferMode	string	转换模式 枚举值: STATIC:静态转换 DYNAMIC:动态转换]
data.bnat.transferSrc.specifyIp	string	单个ip
data.bnat.transferSrc.ipGroups	string[]	ip组

请求成功样例:

```
{
  "data": [
    {
      "enable": false,
      "snat": {
        "srcIpGroups": [
          "ipgTest"
        ],
        "dstNetobj": {
          "zone": [
            "L3_trust_A"
          ],
          "dstNetobjType": "ZONE"
        },
        "natService": [
          "any"
        ],
        "dstIpGroups": [
          "ipgTest"
        ],
        "srcZones": [
          "L3_manage"
        ],
        "transfer": {
          "transferType": "OUTIF_IP"
        }
      }
    }
  ]
}
```

```
    },
    "schedule": "全天",
    "name": "test2",
    "natType": "SNAT",
    "uuid": "EEBD52280767464AAA040EEC2492EA48",
    "description": "12we"
  },
  {
    "enable": false,
    "snat": {
      "srcIpGroups": [
        "ipgTest"
      ],
      "dstNetobj": {
        "zone": [
          "L3_trust_A"
        ],
        "dstNetobjType": "ZONE"
      },
      "natService": [
        "any"
      ],
      "dstIpGroups": [
        "ipgTest"
      ],
      "srcZones": [
        "L3_manage"
      ],
      "transfer": {
        "transferType": "OUTIF_IP"
      }
    },
    "schedule": "全天",
    "name": "tes1",
    "natType": "SNAT",
    "uuid": "5052CC762D2B403EAD799DDABD58FBCD",
    "description": "12we"
  }
],
"code": 0,
"message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.1.2 NAT6策略

4.1.2.1 获取符合过滤条件的全部nat6

功能说明：获取符合过滤条件的全部nat6

1) API请求数据

GET

https://{host}/api/v1/namespaces/@namespace/nat6s

2) 输入参数

- URL参数

参数	类型	是否必选	描述
_start	int32	否	起始位置, 从 0 开始 最小值:0 最大值:2000000

position	int32	否	移动搜索位置参数项
_select	string	否	选择字段, 字段用逗号分隔 最大长度:1000
movesearch	string	否	移动搜索参数项 最小长度:1 最大长度:95
transType	string	否	用于指定转换类型, 可选SNAT,DNAT 枚举值:[SNAT:转换类型为源地址转换 DNAT:转换类型为目的地址转换]
_length	int32	否	获取最大长度, 限制最大值, 如果数据不够 可能返回小于 length 最小值:1 最大值:200 默认值:true

请求样例:

获取符合过滤条件的全部IPv6-NAT策略,查询条件: 起始位置为0,返回最多100条数据

GET https://192.168.1.1/api/v1/namespaces/public/nat6s?_start=0&_length=100

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	分页数据
data.totalItems	int32	总共多少项目
data.totalPages	int32	总共多少页
data.pageNumber	int32	当前页码, 从 1 开始
data.pageSize	int32	每页多大
data.itemsOffset	int32	当前条目偏移, 从 0 开始
data.itemLength	int32	数据列表长度
data.items	object[]	有效数据列表
data.items.uuid	string	唯一标识id
data.items.name	string	唯一标识名称 最小长度:1 最大长度:95
data.items.enable	bool	是否启用
data.items.position	int32	表示位置状态

data.items.natType	string	nat6策略类型 枚举值:[SNAT:转换类型为源地址转换 DNAT:转换类型为目的地址转换]
data.items.description	string	描述 最小长度:0 最大长度:95
data.items.snat	object	源地址转换信息,二选一
data.items.snat.srcZones	string[]	源区域信息
data.items.snat.srcIp	string	源IPv6 格式:ip6-address
data.items.snat.prefix	uint8	源IP的前缀 最小值:4 最大值:128
data.items.snat.dstZones	string[]	目的区域
data.items.snat.transfer	object	转换信息
data.items.snat.transfer.srcIpTrans	string	源IPv6转换 格式:ip6-address
data.items.snat.transfer.prefixTrans	uint8	源IP的前缀 最小值:4 最大值:128
data.items.dnat	object	目的地址转换信息,二选一
data.items.dnat.srcZones	string[]	源区域
data.items.dnat.dstIp	string	目的IP 格式:ip6-address
data.items.dnat.prefix	uint8	目的IP前缀 最小值:4 最大值:128
data.items.dnat.transfer	object	转换信息
data.items.dnat.transfer.dstIpTrans	string	目的IPv6转换 格式:ip6-address
data.items.dnat.transfer.prefixTrans	uint8	目的IP的前缀 最小值:4 最大值:128

请求成功样例:

```
{
  "data": {
    "itemsOffset": 0,
    "items": [
      {
        "enable": true,
        "position": 0,
        "snat": {
          "srcIp": "2001::abc",
          "dstZones": [
            "L3_trust_A"
          ]
        }
      }
    ]
  }
}
```

```
    ],
    "prefix": 128,
    "srcZones": [
      "L3_manage"
    ],
    "transfer": {
      "srcIpTrans": "2001::abc",
      "prefixTrans": 128
    }
  },
  "name": "test",
  "natType": "SNAT",
  "uuid": "9C70B6A1EF8F45FC8C26F80B27986FD8",
  "description": ""
},
{
  "totalItems": 1,
  "pageNumber": 1,
  "itemLength": 1,
  "totalPages": 1,
  "pageSize": 100
},
{
  "code": 0,
  "message": "成功"
}
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.1.2.2 获取单条nat6详细信息

功能说明：获取单条nat6详细信息

1) API请求数据

GET

https://{host}/api/v1/namespaces/@namespace/nat6s/@name

2) 输入参数

请求样例：

```
获取单条IPv6-NAT策略详细信息
GET https://192.168.1.1/api/v1/namespaces/public/nat6s/test
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	nat6信息
data.uuid	string	唯一标识id
data.name	string	唯一标识名称 最小长度:1 最大长度:95
data.enable	bool	是否启用
data.position	int32	表示位置状态

data.natType	string	nat6策略类型 枚举值:[SNAT:转换类型为源地址转换 DNAT:转换类型为目的地址转换]
data.description	string	描述 最小长度:0 最大长度:95
data.sn timer	object	源地址转换信息,二选一
data.sn timer.srcZones	string[]	源区域信息
data.sn timer.srcIp	string	源IPv6 格式:ip6-address
data.sn timer.prefix	uint8	源IP的前缀 最小值:4 最大值:128
data.sn timer.dstZones	string[]	目的区域
data.sn timer.transfer	object	转换信息
data.sn timer.transfer.srcIp Trans	string	源IPv6转换 格式:ip6-address
data.sn timer.transfer.prefix Trans	uint8	源IP的前缀 最小值:4 最大值:128
data.dnat	object	目的地址转换信息,二选一
data.dnat.srcZones	string[]	源区域
data.dnat.dstIp	string	目的IP 格式:ip6-address
data.dnat.prefix	uint8	目的IP前缀 最小值:4 最大值:128
data.dnat.transfer	object	转换信息
data.dnat.transfer.dstIp Trans	string	目的IPv6转换 格式:ip6-address
data.dnat.transfer.prefix Trans	uint8	目的IP的前缀 最小值:4 最大值:128

请求成功样例:

```
{
  "data": {
    "enable": true,
    "position": 0,
    "sn timer": {
      "srcIp": "2001::abc",
      "dstZones": [
        "L3_trust_A"
      ],
      "prefix": 128,
      "srcZones": [
        "L3_manage"
      ],
      "transfer": {
        "srcIpTrans": "2001::abc",
```

```

        "prefixTrans": 128
      },
      "name": "test",
      "natType": "SNAT",
      "uuid": "9C70B6A1EF8F45FC8C26F80B27986FD8",
      "description": ""
    },
    "code": 0,
    "message": "成功"
  }
}

```

请求失败样例：

```

{
  "data": "",
  "code": 22,
  "message": "无效参数"
}

```

4.1.2.3 添加nat6

功能说明：添加一条nat6策略

1) API请求数据

POST https://{host}/api/v1/namespaces/@namespace/nat6s

2) 输入参数

• URL参数

参数	类型	是否必选	描述
_where	string	否	指定位置, 用于移动/插入 枚举值:[top:顶部 bottom:底部 before:在目标位置之前 after:在目标位置之后]
_dest	string	否	目标位置关键字对应的值 最大长度:100
_key	string	否	目标位置关键字 枚举值:[position:指定目标位置关键字为 position name:指定目标位置关键字为 name]

• Body参数

参数	类型	是否必选	描述
obj	object	是	nat6信息
obj.name	string	是	唯一标识名称 最小长度:1 最大长度:95
obj.enable	bool	否	是否启用 默认值:true
obj.position	int32	否	表示位置状态

obj.natType	string	否	nat6策略类型 枚举值:[SNAT:转换类型为源地址转换 DNAT:转换类型为目的地址转换]
obj.description	string	否	描述 最小长度:0 最大长度:95
obj.snat	object	是	源地址转换信息,二选一
obj.snat.srcZones	string[]	否	源区域信息
obj.snat.srcIp	string	否	源IPv6 格式:ip6-address
obj.snat.prefix	uint8	否	源IP的前缀 最小值:4 最大值:128
obj.snat.dstZones	string[]	否	目的区域
obj.snat.transfer	object	否	转换信息
obj.snat.transfer.srcIpTrans	string	否	源IPv6转换 格式:ip6-address
obj.snat.transfer.prefixTrans	uint8	否	源IP的前缀 最小值:4 最大值:128
obj.dnat	object	是	目的地址转换信息,二选一
obj.dnat.srcZones	string[]	否	源区域
obj.dnat.dstIp	string	否	目的IP 格式:ip6-address
obj.dnat.prefix	uint8	否	目的IP前缀 最小值:4 最大值:128
obj.dnat.transfer	object	否	转换信息
obj.dnat.transfer.dstIpTrans	string	否	目的IPv6转换 格式:ip6-address
obj.dnat.transfer.prefixTrans	uint8	否	目的IP的前缀 最小值:4 最大值:128

请求样例:

添加一条IPv6-NAT策略

POST https://192.168.1.1/api/v1/namespaces/public/nat6s

```
{
  "enable": true,
  "position": 1,
  "dnat": {
    "dstIp": "2001::abc",
    "prefix": 128,
    "srcZones": [
      "L3_trust_B"
    ],
    "transfer": {
      "prefixTrans": 128,
      "dstIpTrans": "2001::abd"
    }
  }
}
```

```

    }
  },
  "natType": "DNAT",
  "name": "testD",
  "description": ""
}

```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	nat6信息
data.uuid	string	唯一标识id
data.name	string	唯一标识名称 最小长度:1 最大长度:95
data.enable	bool	是否启用
data.position	int32	表示位置状态
data.natType	string	nat6策略类型 枚举值: SNAT:转换类型为源地址转换 DNAT:转换类型为目的地址转换]
data.description	string	描述 最小长度:0 最大长度:95
data.snat	object	源地址转换信息,二选一
data.snat.srcZones	string[]	源区域信息
data.snat.srcIps	string	源IPv6 格式:ip6-address
data.snat.prefix	uint8	源IP的前缀 最小值:4 最大值:128
data.snat.dstZones	string[]	目的区域
data.snat.transfer	object	转换信息
data.snat.transfer.srcIps Trans	string	源IPv6转换 格式:ip6-address
data.snat.transfer.prefix Trans	uint8	源IP的前缀 最小值:4 最大值:128
data.dnat	object	目的地址转换信息,二选一
data.dnat.srcZones	string[]	源区域
data.dnat.dstIps	string	目的IP 格式:ip6-address

data.dnat.prefix	uint8	目的IP前缀 最小值:4 最大值:128
data.dnat.transfer	object	转换信息
data.dnat.transfer.dstIpTrans	string	目的IPv6转换 格式:ip6-address
data.dnat.transfer.prefixTrans	uint8	目的IP的前缀 最小值:4 最大值:128

请求成功样例:

```
{
  "data": {
    "enable": true,
    "dnat": {
      "dstIp": "2001::abc",
      "prefix": 128,
      "srcZones": [
        "L3_trust_B"
      ],
      "transfer": {
        "prefixTrans": 128,
        "dstIpTrans": "2001::abd"
      }
    },
    "name": "testD",
    "natType": "DNAT",
    "uuid": "D1C4B4F1E3C941BDB4534D1DAF1506E1",
    "description": ""
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.1.2.4 删除单条nat6

功能说明: 删除单条nat6

1) API请求数据

DELETE https://{host}/api/v1/namespaces/@namespace/nat6s/@name

2) 输入参数

请求样例:

删除单条IPv6-NAT策略

DELETE https://192.168.1.1/api/v1/namespaces/public/nat6s/test

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码

message	string	错误信息
data	object	nat6信息
data.uuid	string	唯一标识id
data.name	string	唯一标识名称 最小长度:1 最大长度:95
data.enable	bool	是否启用
data.position	int32	表示位置状态
data.natType	string	nat6策略类型 枚举值:[SNAT:转换类型为源地址转换 DNAT:转换类型为目的地址转换]
data.description	string	描述 最小长度:0 最大长度:95
data.snat	object	源地址转换信息,二选一
data.snat.srcZones	string[]	源区域信息
data.snat.srcIp	string	源IPv6 格式:ip6-address
data.snat.prefix	uint8	源IP的前缀 最小值:4 最大值:128
data.snat.dstZones	string[]	目的区域
data.snat.transfer	object	转换信息
data.snat.transfer.srcIp Trans	string	源IPv6转换 格式:ip6-address
data.snat.transfer.prefix Trans	uint8	源IP的前缀 最小值:4 最大值:128
data.dnat	object	目的地址转换信息,二选一
data.dnat.srcZones	string[]	源区域
data.dnat.dstIp	string	目的IP 格式:ip6-address
data.dnat.prefix	uint8	目的IP前缀 最小值:4 最大值:128
data.dnat.transfer	object	转换信息
data.dnat.transfer.dstIp Trans	string	目的IPv6转换 格式:ip6-address
data.dnat.transfer.prefix Trans	uint8	目的IP的前缀 最小值:4 最大值:128

请求成功样例:

```
{
  "data": {
    "enable": false,
    "snat": {
      "srcIp": "2001::abd",
      "dstZones": [
        "L3_trust_A"
      ],
      "prefix": 128,
      "srcZones": [
        "L3_manage"
      ],
      "transfer": {
        "srcIpTrans": "2001::abc",
        "prefixTrans": 128
      }
    },
    "name": "test",
    "natType": "SNAT",
    "uuid": "9C70B6A1EF8F45FC8C26F80B27986FD8",
    "description": ""
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.1.2.5 批量删除nat6策略

功能说明: 批量删除nat6策略

1) API请求数据

POST

https://{host}/api/batch/v1/namespaces/@namespace/nat6s?_method=delete

2) 输入参数

- Body参数

参数	类型	是否必选	描述
obj	object[]	是	nat6列表
obj.name	string	是	唯一标识名称 最小长度:1 最大长度:95
obj.enable	bool	否	是否启用 默认值:true
obj.position	int32	否	表示位置状态
obj.natType	string	否	nat6策略类型 枚举值:[SNAT:转换类型为源地址转换 DNAT:转换类型为目的地址转换]
obj.description	string	否	描述 最小长度:0 最大长度:95

obj.snat	object	是	源地址转换信息,二选一
obj.snat.srcZones	string[]	否	源区域信息
obj.snat.srcIps	string	否	源IPv6 格式:ip6-address
obj.snat.prefix	uint8	否	源IP的前缀 最小值:4 最大值:128
obj.snat.dstZones	string[]	否	目的区域
obj.snat.transfer	object	否	转换信息
obj.snat.transfer.srcIpsTr ans	string	否	源IPv6转换 格式:ip6-address
obj.snat.transfer.prefixT rans	uint8	否	源IP的前缀 最小值:4 最大值:128
obj.dnat	object	是	目的地址转换信息,二选一
obj.dnat.srcZones	string[]	否	源区域
obj.dnat.dstIps	string	否	目的IP 格式:ip6-address
obj.dnat.prefix	uint8	否	目的IP前缀 最小值:4 最大值:128
obj.dnat.transfer	object	否	转换信息
obj.dnat.transfer.dstIpsTr ans	string	否	目的IPv6转换 格式:ip6-address
obj.dnat.transfer.prefixT rans	uint8	否	目的IP的前缀 最小值:4 最大值:128

请求样例:

批量删除IPv6-NAT策略

POST https://192.168.1.1/api/batch/v1/namespaces/public/nat6s?_method=DELETE

```
[
  {
    "name": "testS"
  },
  {
    "name": "testD"
  }
]
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	nat6列表
data.uuid	string	唯一标识id

data.name	string	唯一标识名称 最小长度:1 最大长度:95
data.enable	bool	是否启用
data.position	int32	表示位置状态
data.natType	string	nat6策略类型 枚举值:[SNAT:转换类型为源地址转换 DNAT:转换类型为目的地址转换]
data.description	string	描述 最小长度:0 最大长度:95
data.snat	object	源地址转换信息,二选一
data.snat.srcZones	string[]	源区域信息
data.snat.srcIp	string	源IPv6 格式:ip6-address
data.snat.prefix	uint8	源IP的前缀 最小值:4 最大值:128
data.snat.dstZones	string[]	目的区域
data.snat.transfer	object	转换信息
data.snat.transfer.srcIp Trans	string	源IPv6转换 格式:ip6-address
data.snat.transfer.prefix Trans	uint8	源IP的前缀 最小值:4 最大值:128
data.dnat	object	目的地址转换信息,二选一
data.dnat.srcZones	string[]	源区域
data.dnat.dstIp	string	目的IP 格式:ip6-address
data.dnat.prefix	uint8	目的IP前缀 最小值:4 最大值:128
data.dnat.transfer	object	转换信息
data.dnat.transfer.dstIp Trans	string	目的IPv6转换 格式:ip6-address
data.dnat.transfer.prefix Trans	uint8	目的IP的前缀 最小值:4 最大值:128

请求成功样例:

```
{
  "data": [
    {
      "enable": false,
      "snat": {
        "srcIp": "2001::abc",
```

```
      "dstZones": [
        "L3_trust_A"
      ],
      "prefix": 128,
      "srcZones": [
        "L3_trust_B"
      ],
      "transfer": {
        "srcIpTrans": "2001::abc",
        "prefixTrans": 128
      }
    },
    "name": "testS",
    "natType": "SNAT",
    "uuid": "186E1015F9BB49E380BA320FD5C384BA",
    "description": ""
  },
  {
    "enable": false,
    "dnat": {
      "dstIp": "2001::abc",
      "prefix": 128,
      "srcZones": [
        "L3_trust_B"
      ],
      "transfer": {
        "prefixTrans": 128,
        "dstIpTrans": "2001::abd"
      }
    },
    "name": "testD",
    "natType": "DNAT",
    "uuid": "D1C4B4F1E3C941BDB4534D1DAF1506E1",
    "description": ""
  }
],
"code": 0,
"message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.1.2.6 批量修改nat6

功能说明：批量增量修改nat6策略

1) API请求数据

PATCH

https://{host}/api/batch/v1/namespaces/@namespace/nat6s

2) 输入参数

- URL 参数

参数	类型	是否必选	描述
_where	string	否	指定位置, 用于移动/插入 枚举值:[top:顶部 bottom:底部 before:在目标位置之前 after:在目标位置之后]

_dest	string	否	目标位置关键字对应的值 最大长度:100
_key	string	否	目标位置关键字 枚举值:[position:指定目标位置关键字为 position name:指定目标位置关键字为 name]

- Body参数

参数	类型	是否必选	描述
obj	object[]	是	nat6列表
obj.name	string	是	唯一标识名称 最小长度:1 最大长度:95
obj.enable	bool	否	是否启用 默认值:true
obj.position	int32	否	表示位置状态
obj.natType	string	否	nat6策略类型 枚举值:[SNAT:转换类型为源地址转换 DNAT:转换类型为目的地址转换]
obj.description	string	否	描述 最小长度:0 最大长度:95
obj.snat	object	是	源地址转换信息,二选一
obj.snat.srcZones	string[]	否	源区域信息
obj.snat.srclp	string	否	源IPv6 格式:ip6-address
obj.snat.prefix	uint8	否	源IP的前缀 最小值:4 最大值:128
obj.snat.dstZones	string[]	否	目的区域
obj.snat.transfer	object	否	转换信息
obj.snat.transfer.srclpTrans	string	否	源IPv6转换 格式:ip6-address
obj.snat.transfer.prefixTrans	uint8	否	源IP的前缀 最小值:4 最大值:128
obj.dnat	object	是	目的地址转换信息,二选一
obj.dnat.srcZones	string[]	否	源区域
obj.dnat.dstIp	string	否	目的IP 格式:ip6-address
obj.dnat.prefix	uint8	否	目的IP前缀 最小值:4 最大值:128

obj.dnat.transfer	object	否	转换信息
obj.dnat.transfer.dstIpTrans	string	否	目的IPv6转换 格式:ip6-address
obj.dnat.transfer.prefixTrans	uint8	否	目的IP的前缀 最小值:4 最大值:128

请求样例:

批量修改IPv6-NAT策略

PATCH https://192.168.1.1/api/batch/v1/namespaces/public/nat6s

```
[
  {
    "name": "testS",
    "enable": false
  },
  {
    "name": "testD",
    "enable": false
  }
]
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	nat6列表
data.uuid	string	唯一标识id
data.name	string	唯一标识名称 最小长度:1 最大长度:95
data.enable	bool	是否启用
data.position	int32	表示位置状态
data.natType	string	nat6策略类型 枚举值: SNAT:转换类型为源地址转换 DNAT:转换类型为目的地址转换]
data.description	string	描述 最小长度:0 最大长度:95
data.snat	object	源地址转换信息,二选一
data.snat.srcZones	string[]	源区域信息
data.snat.srcIp	string	源IPv6 格式:ip6-address
data.snat.prefix	uint8	源IP的前缀 最小值:4 最大值:128
data.snat.dstZones	string[]	目的区域

data.snat.transfer	object	转换信息
data.snat.transfer.srcIpTrans	string	源IPv6转换 格式:ip6-address
data.snat.transfer.prefixTrans	uint8	源IP的前缀 最小值:4 最大值:128
data.dnat	object	目的地址转换信息,二选一
data.dnat.srcZones	string[]	源区域
data.dnat.dstIp	string	目的IP 格式:ip6-address
data.dnat.prefix	uint8	目的IP前缀 最小值:4 最大值:128
data.dnat.transfer	object	转换信息
data.dnat.transfer.dstIpTrans	string	目的IPv6转换 格式:ip6-address
data.dnat.transfer.prefixTrans	uint8	目的IP的前缀 最小值:4 最大值:128

请求成功样例:

```
{
  "data": [
    {
      "enable": false,
      "snat": {
        "srcIp": "2001::abc",
        "dstZones": [
          "L3_trust_A"
        ],
        "prefix": 128,
        "srcZones": [
          "L3_trust_B"
        ],
        "transfer": {
          "srcIpTrans": "2001::abc",
          "prefixTrans": 128
        }
      },
      "name": "testS",
      "natType": "SNAT",
      "uuid": "186E1015F9BB49E380BA320FD5C384BA",
      "description": ""
    },
    {
      "enable": false,
      "dnat": {
        "dstIp": "2001::abc",
        "prefix": 128,
        "srcZones": [
          "L3_trust_B"
        ],
        "transfer": {
          "prefixTrans": 128,
          "dstIpTrans": "2001::abd"
        }
      },
      "name": "testD",
      "natType": "DNAT",
      "uuid": "D1C4B4F1E3C941BDB4534D1DAF1506E1",
      "description": ""
    }
  ],
}
```

```

    "code": 0,
    "message": "成功"
  }

```

请求失败样例:

```

{
  "data": "",
  "code": 22,
  "message": "无效参数"
}

```

4.1.2.7 全量修改nat6

功能说明: 全量修改单条nat6

1) API请求数据

PUT <https://{host}/api/v1/namespaces/@namespace/nat6s/@name>

2) 输入参数

• Body参数

参数	类型	是否必选	描述
obj	object	是	nat6信息
obj.name	string	是	唯一标识名称 最小长度:1 最大长度:95
obj.enable	bool	否	是否启用 默认值:true
obj.position	int32	否	表示位置状态
obj.natType	string	否	nat6策略类型 枚举值:[SNAT:转换类型为源地址转换 DNAT:转换类型为目的地址转换]
obj.description	string	否	描述 最小长度:0 最大长度:95
obj.snat	object	是	源地址转换信息,二选一
obj.snat.srcZones	string[]	否	源区域信息
obj.snat.srclp	string	否	源IPv6 格式:ip6-address
obj.snat.prefix	uint8	否	源IP的前缀 最小值:4 最大值:128
obj.snat.dstZones	string[]	否	目的区域
obj.snat.transfer	object	否	转换信息
obj.snat.transfer.srclpTrans	string	否	源IPv6转换 格式:ip6-address
obj.snat.transfer.prefixTrans	uint8	否	源IP的前缀 最小值:4 最大值:128

obj.dnat	object	是	目的地址转换信息,二选一
obj.dnat.srcZones	string[]	否	源区域
obj.dnat.dstIps	string	否	目的IP 格式:ip6-address
obj.dnat.prefix	uint8	否	目的IP前缀 最小值:4 最大值:128
obj.dnat.transfer	object	否	转换信息
obj.dnat.transfer.dstIpsTrans	string	否	目的IPv6转换 格式:ip6-address
obj.dnat.transfer.prefixTrans	uint8	否	目的IP的前缀 最小值:4 最大值:128

请求样例:

全量修改IPv6-NAT策略

```
PUT https://192.168.1.1/api/v1/namespaces/public/nat6s/test
{
  "enable": true,
  "position": 1,
  "snat": {
    "srcIps": "2001::abd",
    "dstZones": [
      "L3_trust_A"
    ],
    "prefix": 128,
    "srcZones": [
      "L3_manage"
    ],
    "transfer": {
      "srcIpsTrans": "2001::abc",
      "prefixTrans": 128
    }
  },
  "natType": "SNAT",
  "name": "test",
  "description": ""
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	nat6信息
data.uuid	string	唯一标识id
data.name	string	唯一标识名称 最小长度:1 最大长度:95
data.enable	bool	是否启用
data.position	int32	表示位置状态

data.natType	string	nat6策略类型 枚举值:[SNAT:转换类型为源地址转换 DNAT:转换类型为目的地址转换]
data.description	string	描述 最小长度:0 最大长度:95
data.sn timer	object	源地址转换信息,二选一
data.sn timer.srcZones	string[]	源区域信息
data.sn timer.srcIp	string	源IPv6 格式:ip6-address
data.sn timer.prefix	uint8	源IP的前缀 最小值:4 最大值:128
data.sn timer.dstZones	string[]	目的区域
data.sn timer.transfer	object	转换信息
data.sn timer.transfer.srcIp Trans	string	源IPv6转换 格式:ip6-address
data.sn timer.transfer.prefix Trans	uint8	源IP的前缀 最小值:4 最大值:128
data.dnat	object	目的地址转换信息,二选一
data.dnat.srcZones	string[]	源区域
data.dnat.dstIp	string	目的IP 格式:ip6-address
data.dnat.prefix	uint8	目的IP前缀 最小值:4 最大值:128
data.dnat.transfer	object	转换信息
data.dnat.transfer.dstIp Trans	string	目的IPv6转换 格式:ip6-address
data.dnat.transfer.prefix Trans	uint8	目的IP的前缀 最小值:4 最大值:128

请求成功样例:

```
{
  "data": {
    "enable": true,
    "sn timer": {
      "srcIp": "2001::abd",
      "dstZones": [
        "L3_trust_A"
      ],
      "prefix": 128,
      "srcZones": [
        "L3_manage"
      ],
      "transfer": {
        "srcIpTrans": "2001::abc",
        "prefixTrans": 128
      }
    }
  }
}
```



```

    },
    "name": "test",
    "natType": "SNAT",
    "uuid": "9C70B6A1EF8F45FC8C26F80B27986FD8",
    "description": ""
  },
  "code": 0,
  "message": "成功"
}

```

请求失败样例:

```

{
  "data": "",
  "code": 22,
  "message": "无效参数"
}

```

4.1.2.8 增量修改nat6

功能说明: 增量修改nat6策略

1) API请求数据

PATCH <https://{host}/api/v1/namespaces/@namespace/nat6s/@name>

2) 输入参数

• URL参数

参数	类型	是否必选	描述
_arrayop	string	否	对象属性是数组的操作控制 枚举值:[add:添加对应的数组元素 remove:删除对应的数组元素]
_key	string	否	目标位置关键字 枚举值:[position:指定目标位置关键字为 position name:指定目标位置关键字为 name]
_dest	string	否	目标位置关键字对应的值 最大长度:100
_where	string	否	指定位置, 用于移动/插入 枚举值:[top:顶部 bottom:底部 before:在目标位置之前 after:在目标位置之后]

• Body参数

参数	类型	是否必选	描述
obj	object	是	nat6信息
obj.name	string	是	唯一标识名称 最小长度:1 最大长度:95

obj.enable	bool	否	是否启用 默认值:true
obj.position	int32	否	表示位置状态
obj.natType	string	否	nat6策略类型 枚举值:[SNAT:转换类型为源地址转换 DNAT:转换类型为目的地址转换]
obj.description	string	否	描述 最小长度:0 最大长度:95
obj.snat	object	是	源地址转换信息,二选一
obj.snat.srcZones	string[]	否	源区域信息
obj.snat.srcIp	string	否	源IPv6 格式:ip6-address
obj.snat.prefix	uint8	否	源IP的前缀 最小值:4 最大值:128
obj.snat.dstZones	string[]	否	目的区域
obj.snat.transfer	object	否	转换信息
obj.snat.transfer.srcIpTrans	string	否	源IPv6转换 格式:ip6-address
obj.snat.transfer.prefixTrans	uint8	否	源IP的前缀 最小值:4 最大值:128
obj.dnat	object	是	目的地址转换信息,二选一
obj.dnat.srcZones	string[]	否	源区域
obj.dnat.dstIp	string	否	目的IP 格式:ip6-address
obj.dnat.prefix	uint8	否	目的IP前缀 最小值:4 最大值:128
obj.dnat.transfer	object	否	转换信息
obj.dnat.transfer.dstIpTrans	string	否	目的IPv6转换 格式:ip6-address
obj.dnat.transfer.prefixTrans	uint8	否	目的IP的前缀 最小值:4 最大值:128

请求样例:

增量修改IPv6-NAT策略,将策略启用字段修改成禁用

```
PATCH https://192.168.1.1/api/v1/namespaces/public/nat6s/test
{
  "enable": false
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	nat6信息
data.uuid	string	唯一标识id
data.name	string	唯一标识名称 最小长度:1 最大长度:95
data.enable	bool	是否启用
data.position	int32	表示位置状态
data.natType	string	nat6策略类型 枚举值:[SNAT:转换类型为源地址转换 DNAT:转换类型为目的地址转换]
data.description	string	描述 最小长度:0 最大长度:95
data.snat	object	源地址转换信息,二选一
data.snat.srcZones	string[]	源区域信息
data.snat.srcIp	string	源IPv6 格式:ip6-address
data.snat.prefix	uint8	源IP的前缀 最小值:4 最大值:128
data.snat.dstZones	string[]	目的区域
data.snat.transfer	object	转换信息
data.snat.transfer.srcIp Trans	string	源IPv6转换 格式:ip6-address
data.snat.transfer.prefix Trans	uint8	源IP的前缀 最小值:4 最大值:128
data.dnat	object	目的地址转换信息,二选一
data.dnat.srcZones	string[]	源区域
data.dnat.dstIp	string	目的IP 格式:ip6-address
data.dnat.prefix	uint8	目的IP前缀 最小值:4 最大值:128
data.dnat.transfer	object	转换信息

data.dnat.transfer.dstIpTrans	string	目的IPv6转换 格式:ip6-address
data.dnat.transfer.prefixTrans	uint8	目的IP的前缀 最小值:4 最大值:128

请求成功样例:

```
{
  "data": {
    "enable": false,
    "snat": {
      "srcIp": "2001::abd",
      "dstZones": [
        "L3_trust_A"
      ],
      "prefix": 128,
      "srcZones": [
        "L3_manage"
      ],
      "transfer": {
        "srcIpTrans": "2001::abc",
        "prefixTrans": 128
      }
    },
    "name": "test",
    "natType": "SNAT",
    "uuid": "9C70B6A1EF8F45FC8C26F80B27986FD8",
    "description": ""
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.1.3 NAT64策略

4.1.3.1 获取nat64策略的匹配数

功能说明: nat64地址转换策略匹配数查询

1) API请求数据

POST https://{host}/api/batch/v1/namespaces/@namespace/nat64hit

2) 输入参数

• Body参数

参数	类型	是否必选	描述
obj	object	是	地址转换匹配数请求body
obj.filter	object[]	是	规则名称数组
obj.filter.name	string	是	规则名称 最小长度:1 最大长度:95

请求样例:

```
获取nat64策略的匹配数
POST https://192.168.1.1/api/batch/v1/namespaces/public/nat64hit
{
  "filter": [
    {
      "name": "rule1"
    },
    {
      "name": "rule2"
    }
  ]
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	地址转换匹配数数组
data.name	string	规则名称 最小长度:1 最大长度:95
data.hitCount	uint32	规则匹配数 最小值:0

请求成功样例:

```
{
  "data": [
    {
      "name": "rule1",
      "hitCount": 0
    },
    {
      "name": "rule2",
      "hitCount": 0
    }
  ],
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.1.3.2 获取符合过滤条件的全部nat64

功能说明: 获取符合过滤条件的全部nat64

1) API请求数据

GET

https://{host}/api/v1/namespaces/@namespace/nat64s

2) 输入参数

- URL参数

参数	类型	是否必选	描述
_start	int32	否	起始位置, 从 0 开始 最小值:0 最大值:2000000
_search	string	否	模糊搜索关键字 最大长度:95
_select	string	否	选择字段, 字段用逗号分隔 最大长度:1000
position	int32	否	移动搜索位置参数项
movesearch	string	否	移动搜索参数项 最小长度:1 最大长度:95
transType	string	否	用于指定转换类型, 可选V4_TO_V6,V6_TO_V4 枚举值:[V4_TO_V6:转换类型为V4_TO_V6 V6_TO_V4:转换类型为V6_TO_V4]
_length	int32	否	获取最大长度, 限制最大值, 如果数据不够 可能返回小于 length 最小值:1 最大值:200 默认值:true

请求样例:

获取符合过滤条件的全部NAT64策略,查询条件: 起始位置为0,返回最多100条数据
GET https://192.168.1.1/api/v1/namespaces/public/nat64s?_start=0&_length=100

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	分页数据
data.totalItems	int32	总共多少项目
data.totalPages	int32	总共多少页
data.pageNumber	int32	当前页码, 从 1 开始
data.pageSize	int32	每页多大
data.itemsOffset	int32	当前条目偏移, 从 0 开始
data.itemLength	int32	数据列表长度
data.items	object[]	有效数据列表
data.items.uuid	string	唯一标识id

data.items.name	string	唯一标识名称 最小长度:1 最大长度:63
data.items.enable	bool	是否启用
data.items.position	int32	表示位置状态
data.items.natType	string	nat64策略类型 枚举值:[V4_TO_V6:转换类型为V4_TO_V6 V6_TO_V4:转换类型为V6_TO_V4]
data.items.description	string	描述 最小长度:0 最大长度:95
data.items.ignoreAcl	bool	是否忽略acl策略
data.items.nat64Hit	int64	nat64策略匹配次数
data.items.v4ToV6	object	IPv4转换为IPv6信息,二选一
data.items.v4ToV6.srcZones	string[]	源区域
data.items.v4ToV6.srcIpgroups	string[]	源ip组
data.items.v4ToV6.dstIip	string	目的IPv4网段,掩码格式
data.items.v4ToV6.natService	string[]	引用的服务组
data.items.v4ToV6.transfer	object	转换信息
data.items.v4ToV6.transfer.srcTransfer	object	源地址转换信息
data.items.v4ToV6.transfer.srcTransfer.srcTransferType	string	源地址转换类型 枚举值:[IP:单个IP转换 IP_RANGE:IP范围转换]
data.items.v4ToV6.transfer.srcTransfer.ipRange	object	IP范围,不支持bits以及mask 格式:ip-range
data.items.v4ToV6.transfer.srcTransfer.ipRange.start	string	ip范围的起始位置 格式:ip-address
data.items.v4ToV6.transfer.srcTransfer.ipRange.end	string	ip范围的终止位置 格式:ip-address
data.items.v4ToV6.transfer.srcTransfer.ipRange.bits	uint8	ip掩码简写

data.items.v4ToV6.transfer.srcTransfer.ipRange.mask	string	ip掩码 格式:ip-address
data.items.v4ToV6.transfer.srcTransfer.ipv6	string	IPv6地址 格式:ip6-address
data.items.v4ToV6.transfer.dstTransfer	object	目的地址转换信息
data.items.v4ToV6.transfer.dstTransfer.ipv6Segment	string	IPv6网段,掩码格式
data.items.v4ToV6.transfer.dstTransfer.portTransferType	string	端口转换类型 枚举值:[PORT:端口转换 NO_TRANS:不转换]
data.items.v4ToV6.transfer.dstTransfer.portTrans	object[]	端口号,单个端口只需配置start
data.items.v4ToV6.transfer.dstTransfer.portTrans.start	int32	网络端口起始值 最小值:0 最大值:65535
data.items.v4ToV6.transfer.dstTransfer.portTrans.end	int32	网络端口结束值 最小值:0 最大值:65535
data.items.v6ToV4	object	IPv6转换为IPv4信息,二选一
data.items.v6ToV4.srcZones	string[]	源区域
data.items.v6ToV4.srcIpgroups	string[]	源ip组
data.items.v6ToV4.dstIipv6	string	目的IPv6网段,掩码格式
data.items.v6ToV4.natService	string[]	引用的服务组
data.items.v6ToV4.transfer	object	转换信息
data.items.v6ToV4.transfer.srcTransfer	object	源地址转换信息
data.items.v6ToV4.transfer.srcTransfer.srcTransferType	string	源地址转换类型 枚举值:[IP:单个IP转换 IP_RANGE:IP范围转换]
data.items.v6ToV4.transfer.srcTransfer.ipRange	object	IP范围,不支持bits以及mask 格式:ip-range
data.items.v6ToV4.transfer.srcTransfer.ipRange.start	string	ip范围的起始位置 格式:ip-address

data.items.v6ToV4.transfer.srcTransfer.ipRange.end	string	ip范围的终止位置 格式:ip-address
data.items.v6ToV4.transfer.srcTransfer.ipRange.bits	uint8	ip掩码简写
data.items.v6ToV4.transfer.srcTransfer.ipRange.mask	string	ip掩码 格式:ip-address
data.items.v6ToV4.transfer.srcTransfer.ip	string	IPv4地址
data.items.v6ToV4.transfer.dstTransfer	object	目的地址转换信息
data.items.v6ToV4.transfer.dstTransfer.ipSegment	string	IPv4网段,掩码格式
data.items.v6ToV4.transfer.dstTransfer.portTransType	string	端口转换类型 枚举值: PORT:端口转换 NO_TRANS:不转换]
data.items.v6ToV4.transfer.dstTransfer.portTrans	object[]	端口号,单个端口只需配置start
data.items.v6ToV4.transfer.dstTransfer.portTrans.start	int32	网络端口起始值 最小值:0 最大值:65535
data.items.v6ToV4.transfer.dstTransfer.portTrans.end	int32	网络端口结束值 最小值:0 最大值:65535

请求成功样例:

```
{
  "data": {
    "itemsOffset": 0,
    "items": [
      {
        "enable": false,
        "position": 0,
        "natType": "V4_TO_V6",
        "uuid": "51CC5264DA384EAFA36F2110362E28ED",
        "ignoreAcl": true,
        "nat64Hit": 0,
        "v4ToV6": {
          "srcIpGroups": [
            "ip1",
            "私有网段"
          ],
          "natService": [
            "NAT_4to6-复制-复制_自动生成"
          ],
          "srcZones": [
            "L3_trust_B",
            "L3_trust_C"
          ],
          "dstIp": "2.2.2.0/24",
          "transfer": {
            "dstTransfer": {
              "portTrans": [

```

```

        {
            "start": 35
        }
    ],
    "ipv6Segment": "2001::abc/128",
    "portTransType": "PORT"
},
"srcTransfer": {
    "ipRange": {
        "start": "2001::abc",
        "end": "2001::abd"
    },
    "srcTransferType": "IP_RANGE"
}
},
"name": "test1",
"description": ""
},
{
    "enable": true,
    "position": 1,
    "natType": "V6_TO_V4",
    "uuid": "0A9D12DCE83E4F7C9B86AD21E6851E54",
    "ignoreAcl": true,
    "nat64Hit": 0,
    "v6ToV4": {
        "dstIpv6": "2001::abc/128",
        "srcIpGroups": [
            "全部"
        ],
        "natService": [
            "NAT_6th_自动生成"
        ],
        "srcZones": [
            "L3_trust_B",
            "L3_trust_C"
        ],
        "transfer": {
            "dstTransfer": {
                "portTransType": "NO_TRANS",
                "ipSegment": "2.2.3.0/24"
            },
            "srcTransfer": {
                "ip": "3.3.3.3",
                "srcTransferType": "IP"
            }
        }
    },
    "name": "6th",
    "description": ""
}
],
"totalItems": 2,
"pageNumber": 1,
"itemLength": 2,
"totalPages": 1,
"pageSize": 100
},
"code": 0,
"message": "成功"
}

```

请求失败样例:

```

{
    "data": "",
    "code": 22,
    "message": "无效参数"
}

```

4.1.3.3 获取单条nat64详细信息

功能说明：获取单条nat64详细信息

1) API请求数据

GET `https://{host}/api/v1/namespaces/@namespace/nat64s/@name`

2) 输入参数

请求样例：

获取单条NAT64策略详细信息

GET `https://192.168.1.1/api/v1/namespaces/public/nat64s/test`

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	nat64信息
data.uuid	string	唯一标识id
data.name	string	唯一标识名称 最小长度:1 最大长度:63
data.enable	bool	是否启用
data.position	int32	表示位置状态
data.natType	string	nat64策略类型 枚举值: V4_TO_V6:转换类型为V4_TO_V6 V6_TO_V4:转换类型为V6_TO_V4]
data.description	string	描述 最小长度:0 最大长度:95
data.ignoreAcl	bool	是否忽略acl策略
data.nat64Hit	int64	nat64策略匹配次数
data.v4ToV6	object	IPv4转换为IPv6信息,二选一
data.v4ToV6.srcZones	string[]	源区域
data.v4ToV6.srcIpGroups	string[]	源ip组
data.v4ToV6.dstIp	string	目的IPv4网段,掩码格式
data.v4ToV6.natService	string[]	引用的服务组
data.v4ToV6.transfer	object	转换信息
data.v4ToV6.transfer.srcTransfer	object	源地址转换信息

data.v4ToV6.transfer.srcTransfer.srcTransferType	string	源地址转换类型 枚举值: IP:单个IP转换 IP_RANGE:IP范围转换]
data.v4ToV6.transfer.srcTransfer.ipRange	object	IP范围,不支持bits以及mask 格式:ip-range
data.v4ToV6.transfer.srcTransfer.ipRange.start	string	ip范围的起始位置 格式:ip-address
data.v4ToV6.transfer.srcTransfer.ipRange.end	string	ip范围的终止位置 格式:ip-address
data.v4ToV6.transfer.srcTransfer.ipRange.bits	uint8	ip掩码简写
data.v4ToV6.transfer.srcTransfer.ipRange.mask	string	ip掩码 格式:ip-address
data.v4ToV6.transfer.srcTransfer.ipv6	string	IPv6地址 格式:ip6-address
data.v4ToV6.transfer.dstTransfer	object	目的地址转换信息
data.v4ToV6.transfer.dstTransfer.ipv6Segment	string	IPv6网段,掩码格式
data.v4ToV6.transfer.dstTransfer.portTransType	string	端口转换类型 枚举值: PORT:端口转换 NO_TRANS:不转换]
data.v4ToV6.transfer.dstTransfer.portTrans	object[]	端口号,单个端口只需配置start
data.v4ToV6.transfer.dstTransfer.portTrans.start	int32	网络端口起始值 最小值:0 最大值:65535
data.v4ToV6.transfer.dstTransfer.portTrans.end	int32	网络端口结束值 最小值:0 最大值:65535
data.v6ToV4	object	IPv6转换为IPv4信息,二选一
data.v6ToV4.srcZones	string[]	源区域
data.v6ToV4.srcIpGroups	string[]	源ip组
data.v6ToV4.dstIpv6	string	目的IPv6网段,掩码格式
data.v6ToV4.natService	string[]	引用的服务组
data.v6ToV4.transfer	object	转换信息
data.v6ToV4.transfer.srcTransfer	object	源地址转换信息

data.v6ToV4.transfer.srcTransfer.srcTransferType	string	源地址转换类型 枚举值:[IP:单个IP转换 IP_RANGE:IP范围转换]
data.v6ToV4.transfer.srcTransfer.ipRange	object	IP范围,不支持bits以及mask 格式:ip-range
data.v6ToV4.transfer.srcTransfer.ipRange.start	string	ip范围的起始位置 格式:ip-address
data.v6ToV4.transfer.srcTransfer.ipRange.end	string	ip范围的终止位置 格式:ip-address
data.v6ToV4.transfer.srcTransfer.ipRange.bits	uint8	ip掩码简写
data.v6ToV4.transfer.srcTransfer.ipRange.mask	string	ip掩码 格式:ip-address
data.v6ToV4.transfer.srcTransfer.ip	string	IPv4地址
data.v6ToV4.transfer.dstTransfer	object	目的地址转换信息
data.v6ToV4.transfer.dstTransfer.ipSegment	string	IPv4网段,掩码格式
data.v6ToV4.transfer.dstTransfer.portTransType	string	端口转换类型 枚举值:[PORT:端口转换 NO_TRANS:不转换]
data.v6ToV4.transfer.dstTransfer.portTrans	object[]	端口号,单个端口只需配置start
data.v6ToV4.transfer.dstTransfer.portTrans.start	int32	网络端口起始值 最小值:0 最大值:65535
data.v6ToV4.transfer.dstTransfer.portTrans.end	int32	网络端口结束值 最小值:0 最大值:65535

请求成功样例:

```
{
  "data": {
    "enable": true,
    "name": "test",
    "position": 0,
    "ignoreAcl": true,
    "v4ToV6": {
      "srcIpGroups": [
        "ip1",
        "ip2",
        "left"
      ],
      "natService": [
        "any"
      ],
      "srcZones": [
        "L3_trust_B",
        "L3_trust_C"
      ]
    }
  }
}
```

```

    ],
    "dstIp": "2.2.2.0/24",
    "transfer": {
      "dstTransfer": {
        "portTransType": "NO_TRANS",
        "ipv6Segment": "2001::abc/128"
      },
      "srcTransfer": {
        "ipv6": "2001::abc",
        "srcTransferType": "IP"
      }
    },
    "natType": "V4_TO_V6",
    "uuid": "61BAEA8EFE674654A7597D9AA3680A9C",
    "description": ""
  },
  "code": 0,
  "message": "成功"
}

```

请求失败样例：

```

{
  "data": "",
  "code": 22,
  "message": "无效参数"
}

```

4.1.3.4 添加nat64

功能说明：添加一条nat64策略

1) API请求数据

POST <https://{host}/api/v1/namespaces/@namespace/nat64s>

2) 输入参数

• URL参数

参数	类型	是否必选	描述
_where	string	否	指定位置, 用于移动/插入 枚举值:[top:顶部 bottom:底部 before:在目标位置之前 after:在目标位置之后]
_dest	string	否	目标位置关键字对应的值 最大长度:100
_key	string	否	目标位置关键字 枚举值:[position:指定目标位置关键字为 position name:指定目标位置关键字为 name]

• Body参数

参数	类型	是否必选	描述
obj	object	是	nat64信息

obj.name	string	是	唯一标识名称 最小长度:1 最大长度:63
obj.enable	bool	否	是否启用 默认值:true
obj.position	int32	否	表示位置状态
obj.natType	string	是	nat64策略类型 枚举值:[V4_TO_V6:转换类型为V4_TO_V6 V6_TO_V4:转换类型为V6_TO_V4]
obj.description	string	否	描述 最小长度:0 最大长度:95
obj.ignoreAcl	bool	否	是否忽略acl策略 默认值:true
obj.v4ToV6	object	是	IPv4转换为IPv6信息,二选一
obj.v4ToV6.srcZones	string[]	否	源区域
obj.v4ToV6.srcIpGroups	string[]	否	源ip组
obj.v4ToV6.dstIp	string	否	目的IPv4网段,掩码格式
obj.v4ToV6.natService	string[]	否	引用的服务组
obj.v4ToV6.transfer	object	是	转换信息
obj.v4ToV6.transfer.srcTransfer	object	是	源地址转换信息
obj.v4ToV6.transfer.srcTransfer.type	string	否	源地址转换类型 默认值:true 枚举值:[IP:单个IP转换 IP_RANGE:IP范围转换]
obj.v4ToV6.transfer.srcTransfer.ipRange	object	否	IP范围,不支持bits以及mask 格式:ip-range
obj.v4ToV6.transfer.srcTransfer.ipRange.start	string	是	ip范围的起始位置 格式:ip-address
obj.v4ToV6.transfer.srcTransfer.ipRange.end	string	否	ip范围的终止位置 格式:ip-address
obj.v4ToV6.transfer.srcTransfer.ipRange.bits	uint8	否	ip掩码简写
obj.v4ToV6.transfer.srcTransfer.ipRange.mask	string	否	ip掩码 格式:ip-address
obj.v4ToV6.transfer.srcTransfer.ipv6	string	否	IPv6地址 格式:ip6-address
obj.v4ToV6.transfer.dstTransfer	object	是	目的地址转换信息

obj.v4ToV6.transfer.dstTransfer.ipv6Segment	string	否	IPv6网段,掩码格式
obj.v4ToV6.transfer.dstTransfer.portTransType	string	否	端口转换类型 默认值:true 枚举值:[PORT:端口转换 NO_TRANS:不转换]
obj.v4ToV6.transfer.dstTransfer.portTrans	object[]	否	端口号,单个端口只需配置start
obj.v4ToV6.transfer.dstTransfer.portTrans.start	int32	是	网络端口起始值 最小值:0 最大值:65535
obj.v4ToV6.transfer.dstTransfer.portTrans.end	int32	否	网络端口结束值 最小值:0 最大值:65535
obj.v6ToV4	object	是	IPv6转换为IPv4信息,二选一
obj.v6ToV4.srcZones	string[]	否	源区域
obj.v6ToV4.srcIpGroups	string[]	否	源ip组
obj.v6ToV4.dstIpv6	string	否	目的IPv6网段,掩码格式
obj.v6ToV4.natService	string[]	否	引用的服务组
obj.v6ToV4.transfer	object	是	转换信息
obj.v6ToV4.transfer.srcTransfer	object	是	源地址转换信息
obj.v6ToV4.transfer.srcTransfer.srcTransferType	string	否	源地址转换类型 默认值:true 枚举值:[IP:单个IP转换 IP_RANGE:IP范围转换]
obj.v6ToV4.transfer.srcTransfer.ipRange	object	否	IP范围,不支持bits以及mask 格式:ip-range
obj.v6ToV4.transfer.srcTransfer.ipRange.start	string	是	ip范围的起始位置 格式:ip-address
obj.v6ToV4.transfer.srcTransfer.ipRange.end	string	否	ip范围的终止位置 格式:ip-address
obj.v6ToV4.transfer.srcTransfer.ipRange.bits	uint8	否	ip掩码简写
obj.v6ToV4.transfer.srcTransfer.ipRange.mask	string	否	ip掩码 格式:ip-address
obj.v6ToV4.transfer.srcTransfer.ip	string	否	IPv4地址
obj.v6ToV4.transfer.dstTransfer	object	是	目的地址转换信息

obj.v6ToV4.transfer.dstTransfer.ipSegment	string	否	IPv4网段,掩码格式
obj.v6ToV4.transfer.dstTransfer.portTransType	string	否	端口转换类型 默认值:true 枚举值:[PORT:端口转换 NO_TRANS:不转换]
obj.v6ToV4.transfer.dstTransfer.portTrans	object[]	否	端口号,单个端口只需配置start
obj.v6ToV4.transfer.dstTransfer.portTrans.start	int32	是	网络端口起始值 最小值:0 最大值:65535
obj.v6ToV4.transfer.dstTransfer.portTrans.end	int32	否	网络端口结束值 最小值:0 最大值:65535

请求样例:

```
添加NAT64策略
POST https://192.168.1.1/api/v1/namespaces/public/nat64s
{
  "enable": true,
  "position": 1,
  "ignoreAcl": true,
  "v4ToV6": {
    "srcIpGroups": [
      "ip1",
      "ip2",
      "left"
    ],
    "natService": [
      "any"
    ],
    "srcZones": [
      "L3_trust_B",
      "L3_trust_C"
    ],
    "dstIpf": "2.2.2.0/24",
    "transfer": {
      "dstTransfer": {
        "portTransType": "NO_TRANS",
        "ipv6Segment": "2001::abc/128"
      },
      "srcTransfer": {
        "ipv6": "2001::abc",
        "srcTransferType": "IP"
      }
    }
  },
  "natType": "V4_TO_V6",
  "name": "test1",
  "description": ""
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息

data	object	nat64信息
data.uuid	string	唯一标识id
data.name	string	唯一标识名称 最小长度:1 最大长度:63
data.enable	bool	是否启用
data.position	int32	表示位置状态
data.natType	string	nat64策略类型 枚举值:[V4_TO_V6:转换类型为V4_TO_V6 V6_TO_V4:转换类型为V6_TO_V4]
data.description	string	描述 最小长度:0 最大长度:95
data.ignoreAcl	bool	是否忽略acl策略
data.nat64Hit	int64	nat64策略匹配次数
data.v4ToV6	object	IPv4转换为IPv6信息,二选一
data.v4ToV6.srcZones	string[]	源区域
data.v4ToV6.srcIpGroups	string[]	源ip组
data.v4ToV6.dstIp	string	目的IPv4网段,掩码格式
data.v4ToV6.natService	string[]	引用的服务组
data.v4ToV6.transfer	object	转换信息
data.v4ToV6.transfer.srcTransfer	object	源地址转换信息
data.v4ToV6.transfer.srcTransfer.srcTransferType	string	源地址转换类型 枚举值:[IP:单个IP转换 IP_RANGE:IP范围转换]
data.v4ToV6.transfer.srcTransfer.ipRange	object	IP范围,不支持bits以及mask 格式:ip-range
data.v4ToV6.transfer.srcTransfer.ipRange.start	string	ip范围的起始位置 格式:ip-address
data.v4ToV6.transfer.srcTransfer.ipRange.end	string	ip范围的终止位置 格式:ip-address
data.v4ToV6.transfer.srcTransfer.ipRange.bits	uint8	ip掩码简写
data.v4ToV6.transfer.srcTransfer.ipRange.mask	string	ip掩码 格式:ip-address
data.v4ToV6.transfer.srcTransfer.ipv6	string	IPv6地址 格式:ip6-address

data.v4ToV6.transfer.dstTransfer	object	目的地址转换信息
data.v4ToV6.transfer.dstTransfer.ipv6Segment	string	IPv6网段,掩码格式
data.v4ToV6.transfer.dstTransfer.portTransType	string	端口转换类型 枚举值:[PORT:端口转换 NO_TRANS:不转换]
data.v4ToV6.transfer.dstTransfer.portTrans	object[]	端口号,单个端口只需配置start
data.v4ToV6.transfer.dstTransfer.portTrans.start	int32	网络端口起始值 最小值:0 最大值:65535
data.v4ToV6.transfer.dstTransfer.portTrans.end	int32	网络端口结束值 最小值:0 最大值:65535
data.v6ToV4	object	IPv6转换为IPv4信息,二选一
data.v6ToV4.srcZones	string[]	源区域
data.v6ToV4.srcIpGroups	string[]	源ip组
data.v6ToV4.dstIpv6	string	目的IPv6网段,掩码格式
data.v6ToV4.natService	string[]	引用的服务组
data.v6ToV4.transfer	object	转换信息
data.v6ToV4.transfer.srcTransfer	object	源地址转换信息
data.v6ToV4.transfer.srcTransfer.srcTransferType	string	源地址转换类型 枚举值:[IP:单个IP转换 IP_RANGE:IP范围转换]
data.v6ToV4.transfer.srcTransfer.ipRange	object	IP范围,不支持bits以及mask 格式:ip-range
data.v6ToV4.transfer.srcTransfer.ipRange.start	string	ip范围的起始位置 格式:ip-address
data.v6ToV4.transfer.srcTransfer.ipRange.end	string	ip范围的终止位置 格式:ip-address
data.v6ToV4.transfer.srcTransfer.ipRange.bits	uint8	ip掩码简写
data.v6ToV4.transfer.srcTransfer.ipRange.mask	string	ip掩码 格式:ip-address
data.v6ToV4.transfer.srcTransfer.ip	string	IPv4地址
data.v6ToV4.transfer.dstTransfer	object	目的地址转换信息

data.v6ToV4.transfer.dstTransfer.ipSegment	string	IPv4网段,掩码格式
data.v6ToV4.transfer.dstTransfer.portTransType	string	端口转换类型 枚举值:[PORT:端口转换 NO_TRANS:不转换]
data.v6ToV4.transfer.dstTransfer.portTrans	object[]	端口号,单个端口只需配置start
data.v6ToV4.transfer.dstTransfer.portTrans.start	int32	网络端口起始值 最小值:0 最大值:65535
data.v6ToV4.transfer.dstTransfer.portTrans.end	int32	网络端口结束值 最小值:0 最大值:65535

请求成功样例:

```
{
  "data": {
    "enable": true,
    "ignoreAcl": true,
    "name": "test1",
    "v4ToV6": {
      "srcIpGroups": [
        "ip1",
        "ip2",
        "left"
      ],
      "natService": [
        "any"
      ],
      "srcZones": [
        "L3_trust_B",
        "L3_trust_C"
      ],
      "dstIp": "2.2.2.0/24",
      "transfer": {
        "dstTransfer": {
          "portTransType": "NO_TRANS",
          "ipv6Segment": "2001::abc/128"
        },
        "srcTransfer": {
          "ipv6": "2001::abc",
          "srcTransferType": "IP"
        }
      }
    },
    "natType": "V4_TO_V6",
    "uuid": "36021D4AE5BF4EC59648E66AF29E9DB6",
    "description": ""
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.1.3.5 删除单条nat64策略

功能说明：删除单条nat64策略

1) API请求数据

DELETE

https://{host}/api/v1/namespaces/@namespace/nat64s/@name

2) 输入参数

请求样例：

删除单条NAT64策略

DELETE https://192.168.1.1/api/v1/namespaces/public/nat64s/test

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	nat64信息
data.uuid	string	唯一标识id
data.name	string	唯一标识名称 最小长度:1 最大长度:63
data.enable	bool	是否启用
data.position	int32	表示位置状态
data.natType	string	nat64策略类型 枚举值:[V4_TO_V6:转换类型为V4_TO_V6 V6_TO_V4:转换类型为V6_TO_V4]
data.description	string	描述 最小长度:0 最大长度:95
data.ignoreAcl	bool	是否忽略acl策略
data.nat64Hit	int64	nat64策略匹配次数
data.v4ToV6	object	IPv4转换为IPv6信息,二选一
data.v4ToV6.srcZones	string[]	源区域
data.v4ToV6.srcIpGroups	string[]	源ip组
data.v4ToV6.dstIp	string	目的IPv4网段,掩码格式
data.v4ToV6.natService	string[]	引用的服务组
data.v4ToV6.transfer	object	转换信息
data.v4ToV6.transfer.srcTransfer	object	源地址转换信息

data.v4ToV6.transfer.srcTransfer.type	string	源地址转换类型 枚举值: IP:单个IP转换 IP_RANGE:IP范围转换]
data.v4ToV6.transfer.srcTransfer.ipRange	object	IP范围,不支持bits以及mask 格式:ip-range
data.v4ToV6.transfer.srcTransfer.ipRange.start	string	ip范围的起始位置 格式:ip-address
data.v4ToV6.transfer.srcTransfer.ipRange.end	string	ip范围的终止位置 格式:ip-address
data.v4ToV6.transfer.srcTransfer.ipRange.bits	uint8	ip掩码简写
data.v4ToV6.transfer.srcTransfer.ipRange.mask	string	ip掩码 格式:ip-address
data.v4ToV6.transfer.srcTransfer.ipv6	string	IPv6地址 格式:ip6-address
data.v4ToV6.transfer.dstTransfer	object	目的地址转换信息
data.v4ToV6.transfer.dstTransfer.ipv6Segment	string	IPv6网段,掩码格式
data.v4ToV6.transfer.dstTransfer.portTransType	string	端口转换类型 枚举值: PORT:端口转换 NO_TRANS:不转换]
data.v4ToV6.transfer.dstTransfer.portTrans	object[]	端口号,单个端口只需配置start
data.v4ToV6.transfer.dstTransfer.portTrans.start	int32	网络端口起始值 最小值:0 最大值:65535
data.v4ToV6.transfer.dstTransfer.portTrans.end	int32	网络端口结束值 最小值:0 最大值:65535
data.v6ToV4	object	IPv6转换为IPv4信息,二选一
data.v6ToV4.srcZones	string[]	源区域
data.v6ToV4.srcIpGroups	string[]	源ip组
data.v6ToV4.dstIpv6	string	目的IPv6网段,掩码格式
data.v6ToV4.natService	string[]	引用的服务组
data.v6ToV4.transfer	object	转换信息
data.v6ToV4.transfer.srcTransfer	object	源地址转换信息

data.v6ToV4.transfer.srcTransfer.srcTransferType	string	源地址转换类型 枚举值:[IP:单个IP转换 IP_RANGE:IP范围转换]
data.v6ToV4.transfer.srcTransfer.ipRange	object	IP范围,不支持bits以及mask 格式:ip-range
data.v6ToV4.transfer.srcTransfer.ipRange.start	string	ip范围的起始位置 格式:ip-address
data.v6ToV4.transfer.srcTransfer.ipRange.end	string	ip范围的终止位置 格式:ip-address
data.v6ToV4.transfer.srcTransfer.ipRange.bits	uint8	ip掩码简写
data.v6ToV4.transfer.srcTransfer.ipRange.mask	string	ip掩码 格式:ip-address
data.v6ToV4.transfer.srcTransfer.ip	string	IPv4地址
data.v6ToV4.transfer.dstTransfer	object	目的地址转换信息
data.v6ToV4.transfer.dstTransfer.ipSegment	string	IPv4网段,掩码格式
data.v6ToV4.transfer.dstTransfer.portTransType	string	端口转换类型 枚举值:[PORT:端口转换 NO_TRANS:不转换]
data.v6ToV4.transfer.dstTransfer.portTrans	object[]	端口号,单个端口只需配置start
data.v6ToV4.transfer.dstTransfer.portTrans.start	int32	网络端口起始值 最小值:0 最大值:65535
data.v6ToV4.transfer.dstTransfer.portTrans.end	int32	网络端口结束值 最小值:0 最大值:65535

请求成功样例:

```
{
  "data": {
    "enable": false,
    "ignoreAcl": true,
    "name": "test",
    "v4ToV6": {
      "srcIpGroups": [
        "私有网段",
        "ip1"
      ],
      "natService": [
        "NAT_4to6-复制_自动生成"
      ],
      "srcZones": [
        "L3_trust_B",
        "L3_trust_C"
      ],
      "dstIp": "2.2.2.0/24",
    }
  }
}
```

```
{
  "transfer": {
    "dstTransfer": {
      "portTrans": [
        {
          "start": 35
        }
      ],
      "ipv6Segment": "2001::abc/128",
      "portTransType": "PORT"
    },
    "srcTransfer": {
      "ipv6": "2001::abc",
      "srcTransferType": "IP"
    }
  },
  "natType": "V4_TO_V6",
  "uuid": "3C6CE90E9E524B358E8316ADEFFB67A0",
  "description": "asd112"
},
"code": 0,
"message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.1.3.6 批量删除nat64

功能说明: 批量删除nat64

1) API请求数据

POST

https://{host}/api/batch/v1/namespaces/@namespace/nat64s?_method=delete

2) 输入参数

• Body参数

参数	类型	是否必选	描述
obj	object[]	是	nat64列表
obj.name	string	是	唯一标识名称 最小长度:1 最大长度:63
obj.enable	bool	否	是否启用 默认值:true
obj.position	int32	否	表示位置状态
obj.natType	string	是	nat64策略类型 枚举值:[V4_TO_V6:转换类型为V4_TO_V6 V6_TO_V4:转换类型为V6_TO_V4]
obj.description	string	否	描述 最小长度:0 最大长度:95
obj.ignoreAcl	bool	否	是否忽略acl策略 默认值:true

obj.v4ToV6	object	是	IPv4转换为IPv6信息,二选一
obj.v4ToV6.srcZones	string[]	否	源区域
obj.v4ToV6.srcIpGroups	string[]	否	源ip组
obj.v4ToV6.dstIp	string	否	目的IPv4网段,掩码格式
obj.v4ToV6.natService	string[]	否	引用的服务组
obj.v4ToV6.transfer	object	是	转换信息
obj.v4ToV6.transfer.srcTransfer	object	是	源地址转换信息
obj.v4ToV6.transfer.srcTransfer.type	string	否	源地址转换类型 默认值:true 枚举值:[IP:单个IP转换 IP_RANGE:IP范围转换]
obj.v4ToV6.transfer.srcTransfer.ipRange	object	否	IP范围,不支持bits以及mask 格式:ip-range
obj.v4ToV6.transfer.srcTransfer.ipRange.start	string	是	ip范围的起始位置 格式:ip-address
obj.v4ToV6.transfer.srcTransfer.ipRange.end	string	否	ip范围的终止位置 格式:ip-address
obj.v4ToV6.transfer.srcTransfer.ipRange.bits	uint8	否	ip掩码简写
obj.v4ToV6.transfer.srcTransfer.ipRange.mask	string	否	ip掩码 格式:ip-address
obj.v4ToV6.transfer.srcTransfer.ipv6	string	否	IPv6地址 格式:ip6-address
obj.v4ToV6.transfer.dstTransfer	object	是	目的地址转换信息
obj.v4ToV6.transfer.dstTransfer.ipv6Segment	string	否	IPv6网段,掩码格式
obj.v4ToV6.transfer.dstTransfer.portTransType	string	否	端口转换类型 默认值:true 枚举值:[PORT:端口转换 NO_TRANS:不转换]
obj.v4ToV6.transfer.dstTransfer.portTrans	object[]	否	端口号,单个端口只需配置start
obj.v4ToV6.transfer.dstTransfer.portTrans.start	int32	是	网络端口起始值 最小值:0 最大值:65535
obj.v4ToV6.transfer.dstTransfer.portTrans.end	int32	否	网络端口结束值 最小值:0 最大值:65535
obj.v6ToV4	object	是	IPv6转换为IPv4信息,二选一

obj.v6ToV4.srcZones	string[]	否	源区域
obj.v6ToV4.srcIpGroups	string[]	否	源ip组
obj.v6ToV4.dstIpv6	string	否	目的IPv6网段,掩码格式
obj.v6ToV4.natService	string[]	否	引用的服务组
obj.v6ToV4.transfer	object	是	转换信息
obj.v6ToV4.transfer.srcTransfer	object	是	源地址转换信息
obj.v6ToV4.transfer.srcTransferType	string	否	源地址转换类型 默认值:true 枚举值:[IP:单个IP转换 IP_RANGE:IP范围转换]
obj.v6ToV4.transfer.srcTransfer.ipRange	object	否	IP范围,不支持bits以及mask 格式:ip-range
obj.v6ToV4.transfer.srcTransfer.ipRange.start	string	是	ip范围的起始位置 格式:ip-address
obj.v6ToV4.transfer.srcTransfer.ipRange.end	string	否	ip范围的终止位置 格式:ip-address
obj.v6ToV4.transfer.srcTransfer.ipRange.bits	uint8	否	ip掩码简写
obj.v6ToV4.transfer.srcTransfer.ipRange.mask	string	否	ip掩码 格式:ip-address
obj.v6ToV4.transfer.srcTransfer.ip	string	否	IPv4地址
obj.v6ToV4.transfer.dstTransfer	object	是	目的地址转换信息
obj.v6ToV4.transfer.dstTransfer.ipSegment	string	否	IPv4网段,掩码格式
obj.v6ToV4.transfer.dstTransfer.portTransType	string	否	端口转换类型 默认值:true 枚举值:[PORT:端口转换 NO_TRANS:不转换]
obj.v6ToV4.transfer.dstTransfer.portTrans	object[]	否	端口号,单个端口只需配置start
obj.v6ToV4.transfer.dstTransfer.portTrans.start	int32	是	网络端口起始值 最小值:0 最大值:65535
obj.v6ToV4.transfer.dstTransfer.portTrans.end	int32	否	网络端口结束值 最小值:0 最大值:65535

请求样例:

批量删除NAT64策略

POST https://192.168.1.1/api/batch/v1/namespaces/public/nat64s?_method=DELETE

```
[
  {
    "name": "test1"
  },
  {
    "name": "test"
  }
]
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	nat64列表
data.uuid	string	唯一标识id
data.name	string	唯一标识名称 最小长度:1 最大长度:63
data.enable	bool	是否启用
data.position	int32	表示位置状态
data.natType	string	nat64策略类型 枚举值:[V4_TO_V6:转换类型为V4_TO_V6 V6_TO_V4:转换类型为V6_TO_V4]
data.description	string	描述 最小长度:0 最大长度:95
data.ignoreAcl	bool	是否忽略acl策略
data.nat64Hit	int64	nat64策略匹配次数
data.v4ToV6	object	IPv4转换为IPv6信息,二选一
data.v4ToV6.srcZones	string[]	源区域
data.v4ToV6.srcIpGroups	string[]	源ip组
data.v4ToV6.dstIp	string	目的IPv4网段,掩码格式
data.v4ToV6.natService	string[]	引用的服务组
data.v4ToV6.transfer	object	转换信息
data.v4ToV6.transfer.srcTransfer	object	源地址转换信息
data.v4ToV6.transfer.srcTransferType	string	源地址转换类型 枚举值:[IP:单个IP转换 IP_RANGE:IP范围转换]

data.v4ToV6.transfer.srcTransfer.ipRange	object	IP范围,不支持bits以及mask 格式:ip-range
data.v4ToV6.transfer.srcTransfer.ipRange.start	string	ip范围的起始位置 格式:ip-address
data.v4ToV6.transfer.srcTransfer.ipRange.end	string	ip范围的终止位置 格式:ip-address
data.v4ToV6.transfer.srcTransfer.ipRange.bits	uint8	ip掩码简写
data.v4ToV6.transfer.srcTransfer.ipRange.mask	string	ip掩码 格式:ip-address
data.v4ToV6.transfer.srcTransfer.ipv6	string	IPv6地址 格式:ip6-address
data.v4ToV6.transfer.dstTransfer	object	目的地址转换信息
data.v4ToV6.transfer.dstTransfer.ipv6Segment	string	IPv6网段,掩码格式
data.v4ToV6.transfer.dstTransfer.portTransType	string	端口转换类型 枚举值:[PORT:端口转换 NO_TRANS:不转换]
data.v4ToV6.transfer.dstTransfer.portTrans	object[]	端口号,单个端口只需配置start
data.v4ToV6.transfer.dstTransfer.portTrans.start	int32	网络端口起始值 最小值:0 最大值:65535
data.v4ToV6.transfer.dstTransfer.portTrans.end	int32	网络端口结束值 最小值:0 最大值:65535
data.v6ToV4	object	IPv6转换为IPv4信息,二选一
data.v6ToV4.srcZones	string[]	源区域
data.v6ToV4.srcIpGroups	string[]	源ip组
data.v6ToV4.dstIpv6	string	目的IPv6网段,掩码格式
data.v6ToV4.natService	string[]	引用的服务组
data.v6ToV4.transfer	object	转换信息
data.v6ToV4.transfer.srcTransfer	object	源地址转换信息
data.v6ToV4.transfer.srcTransfer.srcTransferType	string	源地址转换类型 枚举值:[IP:单个IP转换 IP_RANGE:IP范围转换]
data.v6ToV4.transfer.srcTransfer.ipRange	object	IP范围,不支持bits以及mask 格式:ip-range

data.v6ToV4.transfer.srcTransfer.ipRange.start	string	ip范围的起始位置 格式:ip-address
data.v6ToV4.transfer.srcTransfer.ipRange.end	string	ip范围的终止位置 格式:ip-address
data.v6ToV4.transfer.srcTransfer.ipRange.bits	uint8	ip掩码简写
data.v6ToV4.transfer.srcTransfer.ipRange.mask	string	ip掩码 格式:ip-address
data.v6ToV4.transfer.srcTransfer.ip	string	IPv4地址
data.v6ToV4.transfer.dstTransfer	object	目的地址转换信息
data.v6ToV4.transfer.dstTransfer.ipSegment	string	IPv4网段,掩码格式
data.v6ToV4.transfer.dstTransfer.portTransType	string	端口转换类型 枚举值:[PORT:端口转换 NO_TRANS:不转换]
data.v6ToV4.transfer.dstTransfer.portTrans	object[]	端口号,单个端口只需配置start
data.v6ToV4.transfer.dstTransfer.portTrans.start	int32	网络端口起始值 最小值:0 最大值:65535
data.v6ToV4.transfer.dstTransfer.portTrans.end	int32	网络端口结束值 最小值:0 最大值:65535

请求成功样例:

```
{
  "data": [
    {
      "enable": true,
      "ignoreAcl": true,
      "name": "test1",
      "v4ToV6": {
        "srcIpGroups": [
          "ip1",
          "ip2",
          "left"
        ],
        "natService": [
          "any"
        ],
        "srcZones": [
          "L3_trust_B",
          "L3_trust_C"
        ],
        "dstIp": "2.2.2.0/24",
        "transfer": {
          "dstTransfer": {
            "portTransType": "NO_TRANS",
            "ipv6Segment": "2001::abc/128"
          },
          "srcTransfer": {
            "ipv6": "2001::abc",
            "srcTransferType": "IP"
          }
        }
      }
    }
  ]
}
```

```
    }
  },
  "natType": "V4_TO_V6",
  "uuid": "36021D4AE5BF4EC59648E66AF29E9DB6",
  "description": ""
},
{
  "enable": true,
  "ignoreAcl": true,
  "name": "test",
  "v4ToV6": {
    "srcIpGroups": [
      "ip1",
      "ip2",
      "left"
    ],
    "natService": [
      "any"
    ],
    "srcZones": [
      "L3_trust_B",
      "L3_trust_C"
    ],
    "dstIp": "2.2.2.0/24",
    "transfer": {
      "dstTransfer": {
        "portTransType": "NO_TRANS",
        "ipv6Segment": "2001::abc/128"
      },
      "srcTransfer": {
        "ipv6": "2001::abc",
        "srcTransferType": "IP"
      }
    }
  },
  "natType": "V4_TO_V6",
  "uuid": "61BAEA8EFE674654A7597D9AA3680A9C",
  "description": ""
}
},
"code": 0,
"message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.1.3.7 全量修改nat64

功能说明：全量修改单条nat64

1) API请求数据

PUT

https://{host}/api/v1/namespaces/@namespace/nat64s/@name

2) 输入参数

• Body参数

参数	类型	是否必选	描述
obj	object	是	nat64信息
obj.name	string	是	唯一标识名称 最小长度:1 最大长度:63

obj.enable	bool	否	是否启用 默认值:true
obj.position	int32	否	表示位置状态
obj.natType	string	是	nat64策略类型 枚举值:[V4_TO_V6:转换类型为V4_TO_V6 V6_TO_V4:转换类型为V6_TO_V4]
obj.description	string	否	描述 最小长度:0 最大长度:95
obj.ignoreAcl	bool	否	是否忽略acl策略 默认值:true
obj.v4ToV6	object	是	IPv4转换为IPv6信息,二选一
obj.v4ToV6.srcZones	string[]	否	源区域
obj.v4ToV6.srcIpGroups	string[]	否	源ip组
obj.v4ToV6.dstIp	string	否	目的IPv4网段,掩码格式
obj.v4ToV6.natService	string[]	否	引用的服务组
obj.v4ToV6.transfer	object	是	转换信息
obj.v4ToV6.transfer.srcTransfer	object	是	源地址转换信息
obj.v4ToV6.transfer.srcTransfer.type	string	否	源地址转换类型 默认值:true 枚举值:[IP:单个IP转换 IP_RANGE:IP范围转换]
obj.v4ToV6.transfer.srcTransfer.ipRange	object	否	IP范围,不支持bits以及mask 格式:ip-range
obj.v4ToV6.transfer.srcTransfer.ipRange.start	string	是	ip范围的起始位置 格式:ip-address
obj.v4ToV6.transfer.srcTransfer.ipRange.end	string	否	ip范围的终止位置 格式:ip-address
obj.v4ToV6.transfer.srcTransfer.ipRange.bits	uint8	否	ip掩码简写
obj.v4ToV6.transfer.srcTransfer.ipRange.mask	string	否	ip掩码 格式:ip-address
obj.v4ToV6.transfer.srcTransfer.ipv6	string	否	IPv6地址 格式:ip6-address
obj.v4ToV6.transfer.dstTransfer	object	是	目的地址转换信息
obj.v4ToV6.transfer.dstTransfer.ipv6Segment	string	否	IPv6网段,掩码格式

obj.v4ToV6.transfer.dstTransfer.portTransType	string	否	端口转换类型 默认值:true 枚举值:[PORT:端口转换 NO_TRANS:不转换]
obj.v4ToV6.transfer.dstTransfer.portTrans	object[]	否	端口号,单个端口只需配置start
obj.v4ToV6.transfer.dstTransfer.portTrans.start	int32	是	网络端口起始值 最小值:0 最大值:65535
obj.v4ToV6.transfer.dstTransfer.portTrans.end	int32	否	网络端口结束值 最小值:0 最大值:65535
obj.v6ToV4	object	是	IPv6转换为IPv4信息,二选一
obj.v6ToV4.srcZones	string[]	否	源区域
obj.v6ToV4.srcIpGroups	string[]	否	源ip组
obj.v6ToV4.dstIpv6	string	否	目的IPv6网段,掩码格式
obj.v6ToV4.natService	string[]	否	引用的服务组
obj.v6ToV4.transfer	object	是	转换信息
obj.v6ToV4.transfer.srcTransfer	object	是	源地址转换信息
obj.v6ToV4.transfer.srcTransfer.srcTransferType	string	否	源地址转换类型 默认值:true 枚举值:[IP:单个IP转换 IP_RANGE:IP范围转换]
obj.v6ToV4.transfer.srcTransfer.ipRange	object	否	IP范围,不支持bits以及mask 格式:ip-range
obj.v6ToV4.transfer.srcTransfer.ipRange.start	string	是	ip范围的起始位置 格式:ip-address
obj.v6ToV4.transfer.srcTransfer.ipRange.end	string	否	ip范围的终止位置 格式:ip-address
obj.v6ToV4.transfer.srcTransfer.ipRange.bits	uint8	否	ip掩码简写
obj.v6ToV4.transfer.srcTransfer.ipRange.mask	string	否	ip掩码 格式:ip-address
obj.v6ToV4.transfer.srcTransfer.ip	string	否	IPv4地址
obj.v6ToV4.transfer.dstTransfer	object	是	目的地址转换信息
obj.v6ToV4.transfer.dstTransfer.ipSegment	string	否	IPv4网段,掩码格式

obj.v6ToV4.transfer.dstTransfer.portTransType	string	否	端口转换类型 默认值:true 枚举值:[PORT:端口转换 NO_TRANS:不转换]
obj.v6ToV4.transfer.dstTransfer.portTrans	object[]	否	端口号,单个端口只需配置start
obj.v6ToV4.transfer.dstTransfer.portTrans.start	int32	是	网络端口起始值 最小值:0 最大值:65535
obj.v6ToV4.transfer.dstTransfer.portTrans.end	int32	否	网络端口结束值 最小值:0 最大值:65535

请求样例:

```
全量修改NAT64策略
PUT https://192.168.1.1/api/v1/namespaces/public/nat64s/test
{
  "enable": true,
  "position": 2,
  "ignoreAcl": true,
  "v4ToV6": {
    "srcIpGroups": [
      "ip1",
      "私有网段"
    ],
    "natService": [
      "NAT_4to6-复制-复制_自动生成"
    ],
    "srcZones": [
      "L3_trust_B",
      "L3_trust_C"
    ],
    "dstIip": "2.2.2.0/24",
    "transfer": {
      "dstTransfer": {
        "portTrans": [
          {
            "start": 35
          }
        ],
        "ipv6Segment": "2001::abc/128",
        "portTransType": "PORT"
      },
      "srcTransfer": {
        "ipRange": {
          "start": "2001::abc",
          "end": "2001::abd"
        },
        "srcTransferType": "IP_RANGE"
      }
    }
  },
  "natType": "V4_TO_V6",
  "name": "test1",
  "description": ""
}
```

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码

message	string	错误信息
data	object	nat64信息
data.uuid	string	唯一标识id
data.name	string	唯一标识名称 最小长度:1 最大长度:63
data.enable	bool	是否启用
data.position	int32	表示位置状态
data.natType	string	nat64策略类型 枚举值:[V4_TO_V6:转换类型为V4_TO_V6 V6_TO_V4:转换类型为V6_TO_V4]
data.description	string	描述 最小长度:0 最大长度:95
data.ignoreAcl	bool	是否忽略acl策略
data.nat64Hit	int64	nat64策略匹配次数
data.v4ToV6	object	IPv4转换为IPv6信息,二选一
data.v4ToV6.srcZones	string[]	源区域
data.v4ToV6.srcIpGroups	string[]	源ip组
data.v4ToV6.dstIp	string	目的IPv4网段,掩码格式
data.v4ToV6.natService	string[]	引用的服务组
data.v4ToV6.transfer	object	转换信息
data.v4ToV6.transfer.srcTransfer	object	源地址转换信息
data.v4ToV6.transfer.srcTransfer.type	string	源地址转换类型 枚举值:[IP:单个IP转换 IP_RANGE:IP范围转换]
data.v4ToV6.transfer.srcTransfer.ipRange	object	IP范围,不支持bits以及mask 格式:ip-range
data.v4ToV6.transfer.srcTransfer.ipRange.start	string	ip范围的起始位置 格式:ip-address
data.v4ToV6.transfer.srcTransfer.ipRange.end	string	ip范围的终止位置 格式:ip-address
data.v4ToV6.transfer.srcTransfer.ipRange.bits	uint8	ip掩码简写
data.v4ToV6.transfer.srcTransfer.ipRange.mask	string	ip掩码 格式:ip-address
data.v4ToV6.transfer.srcTransfer.ipv6	string	IPv6地址 格式:ip6-address

data.v4ToV6.transfer.dstTransfer	object	目的地址转换信息
data.v4ToV6.transfer.dstTransfer.ipv6Segment	string	IPv6网段,掩码格式
data.v4ToV6.transfer.dstTransfer.portTransType	string	端口转换类型 枚举值:[PORT:端口转换 NO_TRANS:不转换]
data.v4ToV6.transfer.dstTransfer.portTrans	object[]	端口号,单个端口只需配置start
data.v4ToV6.transfer.dstTransfer.portTrans.start	int32	网络端口起始值 最小值:0 最大值:65535
data.v4ToV6.transfer.dstTransfer.portTrans.end	int32	网络端口结束值 最小值:0 最大值:65535
data.v6ToV4	object	IPv6转换为IPv4信息,二选一
data.v6ToV4.srcZones	string[]	源区域
data.v6ToV4.srcIpGroups	string[]	源ip组
data.v6ToV4.dstIpv6	string	目的IPv6网段,掩码格式
data.v6ToV4.natService	string[]	引用的服务组
data.v6ToV4.transfer	object	转换信息
data.v6ToV4.transfer.srcTransfer	object	源地址转换信息
data.v6ToV4.transfer.srcTransfer.srcTransferType	string	源地址转换类型 枚举值:[IP:单个IP转换 IP_RANGE:IP范围转换]
data.v6ToV4.transfer.srcTransfer.ipRange	object	IP范围,不支持bits以及mask 格式:ip-range
data.v6ToV4.transfer.srcTransfer.ipRange.start	string	ip范围的起始位置 格式:ip-address
data.v6ToV4.transfer.srcTransfer.ipRange.end	string	ip范围的终止位置 格式:ip-address
data.v6ToV4.transfer.srcTransfer.ipRange.bits	uint8	ip掩码简写
data.v6ToV4.transfer.srcTransfer.ipRange.mask	string	ip掩码 格式:ip-address
data.v6ToV4.transfer.srcTransfer.ip	string	IPv4地址
data.v6ToV4.transfer.dstTransfer	object	目的地址转换信息

data.v6ToV4.transfer.dstTransfer.ipSegment	string	IPv4网段,掩码格式
data.v6ToV4.transfer.dstTransfer.portTransType	string	端口转换类型 枚举值:[PORT:端口转换 NO_TRANS:不转换]
data.v6ToV4.transfer.dstTransfer.portTrans	object[]	端口号,单个端口只需配置start
data.v6ToV4.transfer.dstTransfer.portTrans.start	int32	网络端口起始值 最小值:0 最大值:65535
data.v6ToV4.transfer.dstTransfer.portTrans.end	int32	网络端口结束值 最小值:0 最大值:65535

请求成功样例:

```
{
  "data": {
    "enable": true,
    "ignoreAcl": true,
    "name": "test1",
    "v4ToV6": {
      "srcIpGroups": [
        "ip1",
        "私有网段"
      ],
      "natService": [
        "NAT_4to6-复制-复制_自动生成"
      ],
      "srcZones": [
        "L3_trust_B",
        "L3_trust_C"
      ],
      "dstIp": "2.2.2.0/24",
      "transfer": {
        "dstTransfer": {
          "portTrans": [
            {
              "start": 35
            }
          ],
          "ipv6Segment": "2001::abc/128",
          "portTransType": "PORT"
        },
        "srcTransfer": {
          "ipRange": {
            "start": "2001::abc",
            "end": "2001::abd"
          },
          "srcTransferType": "IP_RANGE"
        }
      }
    },
    "natType": "V4_TO_V6",
    "uuid": "51CC5264DA384EAFA36F2110362E28ED",
    "description": ""
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
```

```
    "message": "无效参数"
  }
}
```

4.1.3.8 增量修改nat64

功能说明：增量修改nat64策略，主要用于移动启禁用

1) API请求数据

PATCH <https://{host}/api/v1/namespaces/@namespace/nat64s/@name>

2) 输入参数

• URL参数

参数	类型	是否必选	描述
_arrayop	string	否	对象属性是数组的操作控制 枚举值:[add:添加对应的数组元素 remove:删除对应的数组元素]
_key	string	否	目标位置关键字 枚举值:[position:指定目标位置关键字为 position name:指定目标位置关键字为 name]
_dest	string	否	目标位置关键字对应的值 最大长度:100
_where	string	否	指定位置, 用于移动/插入 枚举值:[top:顶部 bottom:底部 before:在目标位置之前 after:在目标位置之后]

• Body参数

参数	类型	是否必选	描述
obj	object	是	nat64信息
obj.name	string	是	唯一标识名称 最小长度:1 最大长度:63
obj.enable	bool	否	是否启用 默认值:true
obj.position	int32	否	表示位置状态
obj.natType	string	是	nat64策略类型 枚举值:[V4_TO_V6:转换类型为V4_TO_V6 V6_TO_V4:转换类型为V6_TO_V4]
obj.description	string	否	描述 最小长度:0 最大长度:95

obj.ignoreAcl	bool	否	是否忽略acl策略 默认值:true
obj.v4ToV6	object	是	IPv4转换为IPv6信息,二选一
obj.v4ToV6.srcZones	string[]	否	源区域
obj.v4ToV6.srcIpGroups	string[]	否	源ip组
obj.v4ToV6.dstIp	string	否	目的IPv4网段,掩码格式
obj.v4ToV6.natService	string[]	否	引用的服务组
obj.v4ToV6.transfer	object	是	转换信息
obj.v4ToV6.transfer.srcTransfer	object	是	源地址转换信息
obj.v4ToV6.transfer.srcTransferType	string	否	源地址转换类型 默认值:true 枚举值:[IP:单个IP转换 IP_RANGE:IP范围转换]
obj.v4ToV6.transfer.srcTransfer.ipRange	object	否	IP范围,不支持bits以及mask 格式:ip-range
obj.v4ToV6.transfer.srcTransfer.ipRange.start	string	是	ip范围的起始位置 格式:ip-address
obj.v4ToV6.transfer.srcTransfer.ipRange.end	string	否	ip范围的终止位置 格式:ip-address
obj.v4ToV6.transfer.srcTransfer.ipRange.bits	uint8	否	ip掩码简写
obj.v4ToV6.transfer.srcTransfer.ipRange.mask	string	否	ip掩码 格式:ip-address
obj.v4ToV6.transfer.srcTransfer.ipv6	string	否	IPv6地址 格式:ip6-address
obj.v4ToV6.transfer.dstTransfer	object	是	目的地址转换信息
obj.v4ToV6.transfer.dstTransfer.ipv6Segment	string	否	IPv6网段,掩码格式
obj.v4ToV6.transfer.dstTransfer.portTransType	string	否	端口转换类型 默认值:true 枚举值:[PORT:端口转换 NO_TRANS:不转换]
obj.v4ToV6.transfer.dstTransfer.portTrans	object[]	否	端口号,单个端口只需配置start
obj.v4ToV6.transfer.dstTransfer.portTrans.start	int32	是	网络端口起始值 最小值:0 最大值:65535

obj.v4ToV6.transfer.dstTransfer.portTrans.end	int32	否	网络端口结束值 最小值:0 最大值:65535
obj.v6ToV4	object	是	IPv6转换为IPv4信息,二选一
obj.v6ToV4.srcZones	string[]	否	源区域
obj.v6ToV4.srcIpGroups	string[]	否	源ip组
obj.v6ToV4.dstIpv6	string	否	目的IPv6网段,掩码格式
obj.v6ToV4.natService	string[]	否	引用的服务组
obj.v6ToV4.transfer	object	是	转换信息
obj.v6ToV4.transfer.srcTransfer	object	是	源地址转换信息
obj.v6ToV4.transfer.srcTransferType	string	否	源地址转换类型 默认值:true 枚举值:[IP:单个IP转换 IP_RANGE:IP范围转换]
obj.v6ToV4.transfer.srcTransfer.ipRange	object	否	IP范围,不支持bits以及mask 格式:ip-range
obj.v6ToV4.transfer.srcTransfer.ipRange.start	string	是	ip范围的起始位置 格式:ip-address
obj.v6ToV4.transfer.srcTransfer.ipRange.end	string	否	ip范围的终止位置 格式:ip-address
obj.v6ToV4.transfer.srcTransfer.ipRange.bits	uint8	否	ip掩码简写
obj.v6ToV4.transfer.srcTransfer.ipRange.mask	string	否	ip掩码 格式:ip-address
obj.v6ToV4.transfer.srcTransfer.ip	string	否	IPv4地址
obj.v6ToV4.transfer.dstTransfer	object	是	目的地址转换信息
obj.v6ToV4.transfer.dstTransfer.ipSegment	string	否	IPv4网段,掩码格式
obj.v6ToV4.transfer.dstTransfer.portTransType	string	否	端口转换类型 默认值:true 枚举值:[PORT:端口转换 NO_TRANS:不转换]
obj.v6ToV4.transfer.dstTransfer.portTrans	object[]	否	端口号,单个端口只需配置start
obj.v6ToV4.transfer.dstTransfer.portTrans.start	int32	是	网络端口起始值 最小值:0 最大值:65535

obj.v6ToV4.transfer.dstTransfer.portTrans.end	int32	否	网络端口结束值 最小值:0 最大值:65535
---	-------	---	-------------------------------

请求样例:

增量修改NAT64策略,将策略启禁用

```
PATCH https://192.168.1.1/api/v1/namespaces/public/nat64s/test
{
  "enable": false
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	nat64信息
data.uuid	string	唯一标识id
data.name	string	唯一标识名称 最小长度:1 最大长度:63
data.enable	bool	是否启用
data.position	int32	表示位置状态
data.natType	string	nat64策略类型 枚举值: V4_TO_V6:转换类型为V4_TO_V6 V6_TO_V4:转换类型为V6_TO_V4]
data.description	string	描述 最小长度:0 最大长度:95
data.ignoreAcl	bool	是否忽略acl策略
data.nat64Hit	int64	nat64策略匹配次数
data.v4ToV6	object	IPv4转换为IPv6信息,二选一
data.v4ToV6.srcZones	string[]	源区域
data.v4ToV6.srcIpGroups	string[]	源ip组
data.v4ToV6.dstIp	string	目的IPv4网段,掩码格式
data.v4ToV6.natService	string[]	引用的服务组
data.v4ToV6.transfer	object	转换信息
data.v4ToV6.transfer.srcTransfer	object	源地址转换信息

data.v4ToV6.transfer.srcTransfer.type	string	源地址转换类型 枚举值: IP:单个IP转换 IP_RANGE:IP范围转换]
data.v4ToV6.transfer.srcTransfer.ipRange	object	IP范围,不支持bits以及mask 格式:ip-range
data.v4ToV6.transfer.srcTransfer.ipRange.start	string	ip范围的起始位置 格式:ip-address
data.v4ToV6.transfer.srcTransfer.ipRange.end	string	ip范围的终止位置 格式:ip-address
data.v4ToV6.transfer.srcTransfer.ipRange.bits	uint8	ip掩码简写
data.v4ToV6.transfer.srcTransfer.ipRange.mask	string	ip掩码 格式:ip-address
data.v4ToV6.transfer.srcTransfer.ipv6	string	IPv6地址 格式:ip6-address
data.v4ToV6.transfer.dstTransfer	object	目的地址转换信息
data.v4ToV6.transfer.dstTransfer.ipv6Segment	string	IPv6网段,掩码格式
data.v4ToV6.transfer.dstTransfer.portTransType	string	端口转换类型 枚举值: PORT:端口转换 NO_TRANS:不转换]
data.v4ToV6.transfer.dstTransfer.portTrans	object[]	端口号,单个端口只需配置start
data.v4ToV6.transfer.dstTransfer.portTrans.start	int32	网络端口起始值 最小值:0 最大值:65535
data.v4ToV6.transfer.dstTransfer.portTrans.end	int32	网络端口结束值 最小值:0 最大值:65535
data.v6ToV4	object	IPv6转换为IPv4信息,二选一
data.v6ToV4.srcZones	string[]	源区域
data.v6ToV4.srcIpGroups	string[]	源ip组
data.v6ToV4.dstIpv6	string	目的IPv6网段,掩码格式
data.v6ToV4.natService	string[]	引用的服务组
data.v6ToV4.transfer	object	转换信息
data.v6ToV4.transfer.srcTransfer	object	源地址转换信息

data.v6ToV4.transfer.srcTransfer.srcTransferType	string	源地址转换类型 枚举值:[IP:单个IP转换 IP_RANGE:IP范围转换]
data.v6ToV4.transfer.srcTransfer.ipRange	object	IP范围,不支持bits以及mask 格式:ip-range
data.v6ToV4.transfer.srcTransfer.ipRange.start	string	ip范围的起始位置 格式:ip-address
data.v6ToV4.transfer.srcTransfer.ipRange.end	string	ip范围的终止位置 格式:ip-address
data.v6ToV4.transfer.srcTransfer.ipRange.bits	uint8	ip掩码简写
data.v6ToV4.transfer.srcTransfer.ipRange.mask	string	ip掩码 格式:ip-address
data.v6ToV4.transfer.srcTransfer.ip	string	IPv4地址
data.v6ToV4.transfer.dstTransfer	object	目的地址转换信息
data.v6ToV4.transfer.dstTransfer.ipSegment	string	IPv4网段,掩码格式
data.v6ToV4.transfer.dstTransfer.portTransType	string	端口转换类型 枚举值:[PORT:端口转换 NO_TRANS:不转换]
data.v6ToV4.transfer.dstTransfer.portTrans	object[]	端口号,单个端口只需配置start
data.v6ToV4.transfer.dstTransfer.portTrans.start	int32	网络端口起始值 最小值:0 最大值:65535
data.v6ToV4.transfer.dstTransfer.portTrans.end	int32	网络端口结束值 最小值:0 最大值:65535

请求成功样例:

```
{
  "data": {
    "enable": false,
    "ignoreAcl": true,
    "name": "test",
    "v4ToV6": {
      "srcIpGroups": [
        "私有网段",
        "ip1"
      ],
      "natService": [
        "NAT_4to6-复制_自动生成"
      ],
      "srcZones": [
        "L3_trust_B",
        "L3_trust_C"
      ],
      "dstIp": "2.2.2.0/24",
    }
  }
}
```

```
    "transfer": {
      "dstTransfer": {
        "portTrans": [
          {
            "start": 35
          }
        ],
        "ipv6Segment": "2001::abc/128",
        "portTransType": "PORT"
      },
      "srcTransfer": {
        "ipv6": "2001::abc",
        "srcTransferType": "IP"
      }
    },
    "natType": "V4_TO_V6",
    "uuid": "3C6CE90E9E524B358E8316ADEFFB67A0",
    "description": "asd112"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.1.3.9 批量修改nat64

功能说明: 批量增量修改nat64策略

1) API请求数据

PATCH

https://{host}/api/batch/v1/namespaces/@namespace/nat64s

2) 输入参数

• URL参数

参数	类型	是否必选	描述
_where	string	否	指定位置, 用于移动/插入 枚举值:[top:顶部 bottom:底部 before:在目标位置之前 after:在目标位置之后]
_dest	string	否	目标位置关键字对应的值 最大长度:100
_key	string	否	目标位置关键字 枚举值:[position:指定目标位置关键字为 position name:指定目标位置关键字为 name]

• Body参数

参数	类型	是否必选	描述
obj	object[]	是	nat64列表

obj.name	string	是	唯一标识名称 最小长度:1 最大长度:63
obj.enable	bool	否	是否启用 默认值:true
obj.position	int32	否	表示位置状态
obj.natType	string	是	nat64策略类型 枚举值:[V4_TO_V6:转换类型为V4_TO_V6 V6_TO_V4:转换类型为V6_TO_V4]
obj.description	string	否	描述 最小长度:0 最大长度:95
obj.ignoreAcl	bool	否	是否忽略acl策略 默认值:true
obj.v4ToV6	object	是	IPv4转换为IPv6信息,二选一
obj.v4ToV6.srcZones	string[]	否	源区域
obj.v4ToV6.srcIpGroups	string[]	否	源ip组
obj.v4ToV6.dstIp	string	否	目的IPv4网段,掩码格式
obj.v4ToV6.natService	string[]	否	引用的服务组
obj.v4ToV6.transfer	object	是	转换信息
obj.v4ToV6.transfer.srcTransfer	object	是	源地址转换信息
obj.v4ToV6.transfer.srcTransfer.type	string	否	源地址转换类型 默认值:true 枚举值:[IP:单个IP转换 IP_RANGE:IP范围转换]
obj.v4ToV6.transfer.srcTransfer.ipRange	object	否	IP范围,不支持bits以及mask 格式:ip-range
obj.v4ToV6.transfer.srcTransfer.ipRange.start	string	是	ip范围的起始位置 格式:ip-address
obj.v4ToV6.transfer.srcTransfer.ipRange.end	string	否	ip范围的终止位置 格式:ip-address
obj.v4ToV6.transfer.srcTransfer.ipRange.bits	uint8	否	ip掩码简写
obj.v4ToV6.transfer.srcTransfer.ipRange.mask	string	否	ip掩码 格式:ip-address
obj.v4ToV6.transfer.srcTransfer.ipv6	string	否	IPv6地址 格式:ip6-address
obj.v4ToV6.transfer.dstTransfer	object	是	目的地址转换信息

obj.v4ToV6.transfer.dstTransfer.ipv6Segment	string	否	IPv6网段,掩码格式
obj.v4ToV6.transfer.dstTransfer.portTransType	string	否	端口转换类型 默认值:true 枚举值:[PORT:端口转换 NO_TRANS:不转换]
obj.v4ToV6.transfer.dstTransfer.portTrans	object[]	否	端口号,单个端口只需配置start
obj.v4ToV6.transfer.dstTransfer.portTrans.start	int32	是	网络端口起始值 最小值:0 最大值:65535
obj.v4ToV6.transfer.dstTransfer.portTrans.end	int32	否	网络端口结束值 最小值:0 最大值:65535
obj.v6ToV4	object	是	IPv6转换为IPv4信息,二选一
obj.v6ToV4.srcZones	string[]	否	源区域
obj.v6ToV4.srcIpGroups	string[]	否	源ip组
obj.v6ToV4.dstIpv6	string	否	目的IPv6网段,掩码格式
obj.v6ToV4.natService	string[]	否	引用的服务组
obj.v6ToV4.transfer	object	是	转换信息
obj.v6ToV4.transfer.srcTransfer	object	是	源地址转换信息
obj.v6ToV4.transfer.srcTransfer.srcTransferType	string	否	源地址转换类型 默认值:true 枚举值:[IP:单个IP转换 IP_RANGE:IP范围转换]
obj.v6ToV4.transfer.srcTransfer.ipRange	object	否	IP范围,不支持bits以及mask 格式:ip-range
obj.v6ToV4.transfer.srcTransfer.ipRange.start	string	是	ip范围的起始位置 格式:ip-address
obj.v6ToV4.transfer.srcTransfer.ipRange.end	string	否	ip范围的终止位置 格式:ip-address
obj.v6ToV4.transfer.srcTransfer.ipRange.bits	uint8	否	ip掩码简写
obj.v6ToV4.transfer.srcTransfer.ipRange.mask	string	否	ip掩码 格式:ip-address
obj.v6ToV4.transfer.srcTransfer.ip	string	否	IPv4地址
obj.v6ToV4.transfer.dstTransfer	object	是	目的地址转换信息

obj.v6ToV4.transfer.dst Transfer.ipSegment	string	否	IPv4网段,掩码格式
obj.v6ToV4.transfer.dst Transfer.portTransType	string	否	端口转换类型 默认值:true 枚举值:[PORT:端口转换 NO_TRANS:不转换]
obj.v6ToV4.transfer.dst Transfer.portTrans	object[]	否	端口号,单个端口只需配置start
obj.v6ToV4.transfer.dst Transfer.portTrans.start	int32	是	网络端口起始值 最小值:0 最大值:65535
obj.v6ToV4.transfer.dst Transfer.portTrans.end	int32	否	网络端口结束值 最小值:0 最大值:65535

请求样例:

```
批量修改NAT64策略
PATCH https://192.168.1.1/api/batch/v1/namespaces/public/nat64s
[
  {
    "name": "test",
    "enable": false
  },
  {
    "name": "test1",
    "enable": false
  }
]
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	nat64列表
data.uuid	string	唯一标识id
data.name	string	唯一标识名称 最小长度:1 最大长度:63
data.enable	bool	是否启用
data.position	int32	表示位置状态
data.natType	string	nat64策略类型 枚举值:[V4_TO_V6:转换类型为V4_TO_V6 V6_TO_V4:转换类型为V6_TO_V4]

data.description	string	描述 最小长度:0 最大长度:95
data.ignoreAcl	bool	是否忽略acl策略
data.nat64Hit	int64	nat64策略匹配次数
data.v4ToV6	object	IPv4转换为IPv6信息,二选一
data.v4ToV6.srcZones	string[]	源区域
data.v4ToV6.srcIpGroups	string[]	源ip组
data.v4ToV6.dstIp	string	目的IPv4网段,掩码格式
data.v4ToV6.natService	string[]	引用的服务组
data.v4ToV6.transfer	object	转换信息
data.v4ToV6.transfer.srcTransfer	object	源地址转换信息
data.v4ToV6.transfer.srcTransfer.srcTransferType	string	源地址转换类型 枚举值:[IP:单个IP转换 IP_RANGE:IP范围转换]
data.v4ToV6.transfer.srcTransfer.ipRange	object	IP范围,不支持bits以及mask 格式:ip-range
data.v4ToV6.transfer.srcTransfer.ipRange.start	string	ip范围的起始位置 格式:ip-address
data.v4ToV6.transfer.srcTransfer.ipRange.end	string	ip范围的终止位置 格式:ip-address
data.v4ToV6.transfer.srcTransfer.ipRange.bits	uint8	ip掩码简写
data.v4ToV6.transfer.srcTransfer.ipRange.mask	string	ip掩码 格式:ip-address
data.v4ToV6.transfer.srcTransfer.ipv6	string	IPv6地址 格式:ip6-address
data.v4ToV6.transfer.dstTransfer	object	目的地址转换信息
data.v4ToV6.transfer.dstTransfer.ipv6Segment	string	IPv6网段,掩码格式
data.v4ToV6.transfer.dstTransfer.portTransType	string	端口转换类型 枚举值:[PORT:端口转换 NO_TRANS:不转换]
data.v4ToV6.transfer.dstTransfer.portTrans	object[]	端口号,单个端口只需配置start
data.v4ToV6.transfer.dstTransfer.portTrans.start	int32	网络端口起始值 最小值:0 最大值:65535

data.v4ToV6.transfer.dstTransfer.portTrans.end	int32	网络端口结束值 最小值:0 最大值:65535
data.v6ToV4	object	IPv6转换为IPv4信息,二选一
data.v6ToV4.srcZones	string[]	源区域
data.v6ToV4.srcIpGroups	string[]	源ip组
data.v6ToV4.dstIpv6	string	目的IPv6网段,掩码格式
data.v6ToV4.natService	string[]	引用的服务组
data.v6ToV4.transfer	object	转换信息
data.v6ToV4.transfer.srcTransfer	object	源地址转换信息
data.v6ToV4.transfer.srcTransfer.srcTransferType	string	源地址转换类型 枚举值:[IP:单个IP转换 IP_RANGE:IP范围转换]
data.v6ToV4.transfer.srcTransfer.ipRange	object	IP范围,不支持bits以及mask 格式:ip-range
data.v6ToV4.transfer.srcTransfer.ipRange.start	string	ip范围的起始位置 格式:ip-address
data.v6ToV4.transfer.srcTransfer.ipRange.end	string	ip范围的终止位置 格式:ip-address
data.v6ToV4.transfer.srcTransfer.ipRange.bits	uint8	ip掩码简写
data.v6ToV4.transfer.srcTransfer.ipRange.mask	string	ip掩码 格式:ip-address
data.v6ToV4.transfer.srcTransfer.ip	string	IPv4地址
data.v6ToV4.transfer.dstTransfer	object	目的地址转换信息
data.v6ToV4.transfer.dstTransfer.ipSegment	string	IPv4网段,掩码格式
data.v6ToV4.transfer.dstTransfer.portTransType	string	端口转换类型 枚举值:[PORT:端口转换 NO_TRANS:不转换]
data.v6ToV4.transfer.dstTransfer.portTrans	object[]	端口号,单个端口只需配置start
data.v6ToV4.transfer.dstTransfer.portTrans.start	int32	网络端口起始值 最小值:0 最大值:65535

data.v6ToV4.transfer.dstTransfer.portTrans.end	int32	网络端口结束值 最小值:0 最大值:65535
--	-------	-------------------------------

请求成功样例:

```
{
  "data": [
    {
      "enable": false,
      "ignoreAcl": true,
      "name": "test",
      "v4ToV6": {
        "srcIpGroups": [
          "私有网段",
          "ip1"
        ],
        "natService": [
          "NAT_4to6-复制_自动生成"
        ],
        "srcZones": [
          "L3_trust_B",
          "L3_trust_C"
        ],
        "dstIp": "2.2.2.0/24",
        "transfer": {
          "dstTransfer": {
            "portTrans": [
              {
                "start": 35
              }
            ],
            "ipv6Segment": "2001::abc/128",
            "portTransType": "PORT"
          },
          "srcTransfer": {
            "ipv6": "2001::abc",
            "srcTransferType": "IP"
          }
        }
      },
      "natType": "V4_TO_V6",
      "uuid": "3C6CE90E9E524B358E8316ADEFFB67A0",
      "description": ""
    },
    {
      "enable": false,
      "ignoreAcl": true,
      "name": "test1",
      "v4ToV6": {
        "srcIpGroups": [
          "ip1",
          "私有网段"
        ],
        "natService": [
          "NAT_4to6-复制-复制_自动生成"
        ],
        "srcZones": [
          "L3_trust_B",
          "L3_trust_C"
        ],
        "dstIp": "2.2.2.0/24",
        "transfer": {
          "dstTransfer": {
            "portTrans": [
              {
                "start": 35
              }
            ],
            "ipv6Segment": "2001::abc/128",
            "portTransType": "PORT"
          },
          "srcTransfer": {
            "ipRange": {
              "start": "2001::abc",
              "end": "2001::abd"
            }
          }
        }
      }
    }
  ]
}
```

```
    },
    "srcTransferType": "IP_RANGE"
  }
},
},
},
"natType": "V4_TO_V6",
"uuid": "51CC5264DA384EAFA36F2110362E28ED",
"description": ""
}
},
"code": 0,
"message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.1.4 dnsmmap

4.1.4.1 获取匹配上的dnsmmap数据

功能说明: 获取匹配上的dnsmmap数据

1) API请求数据

GET <https://{host}/api/v1/namespaces/@namespace/dnsmappings>

2) 输入参数

- URL参数

参数	类型	是否必选	描述
_start	int32	否	起始位置, 从 0 开始 最小值:0 最大值:2000000
_length	int32	否	获取最大长度, 限制最大值, 如果数据不够 可能返回小于 length 最小值:1 最大值:200 默认值:true
_select	string	否	选择字段, 字段用逗号分隔 最大长度:1000

请求样例:

获取所有的DNS-mappings数据

GET <https://192.168.1.1/api/v1/namespaces/public/dnsmappings>

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	分页数据

data.totalItems	int32	总共多少项目
data.totalPages	int32	总共多少页
data.pageNumber	int32	当前页码, 从 1 开始
data.pageSize	int32	每页多大
data.itemsOffset	int32	当前条目偏移, 从 0 开始
data.itemLength	int32	数据列表长度
data.items	object[]	有效数据列表
data.items.uuid	string	唯一标识id
data.items.domains	string[]	域名
data.items.internetIp	string	单个IPv4 公网ip
data.items.internallp	string	单个IPv4 内网ip

请求成功样例:

```
{
  "data": {
    "itemsOffset": 0,
    "items": [
      {
        "internallp": "2.2.2.2",
        "internetIp": "1.1.1.1",
        "uuid": "6D4A5E5690654A5798E3A8D8B29DA5E5",
        "domains": [
          "www.google.com"
        ]
      }
    ],
    "totalItems": 1,
    "pageNumber": 1,
    "itemLength": 1,
    "totalPages": 1,
    "pageSize": 100
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.1.4.2 获取单条Dnsmapping详细信息

功能说明: 获取单条Dnsmapping详细信息

1) API请求数据

GET <https://{host}/api/v1/namespaces/@namespace/dnsmappings/@uuid>

2) 输入参数

请求样例:

获取单条DNS-mappings数据详细信息,使用uuid索引

GET <https://192.168.1.1/api/v1/namespaces/public/dnsmappings/6D4A5E5690654A5798E3A8D8B29DA5E5>

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	dnsmapping信息
data.uuid	string	唯一标识id
data.domains	string[]	域名
data.internetIp	string	单个IPv4 公网ip
data.internalIp	string	单个IPv4 内网ip

请求成功样例:

```
{
  "data": {
    "internalIp": "2.2.2.2",
    "internetIp": "1.1.1.1",
    "uuid": "6D4A5E5690654A5798E3A8D8B29DA5E5",
    "domains": [
      "www.google.com"
    ]
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.1.4.3 使用整个数据删除单条Dnsmapping

功能说明: 使用整个数据删除单条Dnsmapping

1) API请求数据

POST

https://{host}/api/v1/namespaces/@namespace/dns mappings?_method=delete

2) 输入参数

- Body参数

参数	类型	是否必选	描述
obj	object	是	dnsmapping信息
obj.domains	string[]	是	域名
obj.internetIp	string	是	单个IPv4 公网ip
obj.internalIp	string	是	单个IPv4 内网ip

请求样例:

使用具体数据删除单条DNS-mappings数据

POST https://192.168.1.1/api/v1/namespaces/public/dns mappings?_method=DELETE

```
{
  "internalip": "192.168.1.1",
  "internetip": "192.168.1.1",
  "domains": [
    "www.sangfor.com"
  ]
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	dnsmap信息
data.uuid	string	唯一标识id
data.domains	string[]	域名
data.internetip	string	单个IPv4 公网ip
data.internalip	string	单个IPv4 内网ip

请求成功样例:

```
{
  "data": {
    "uuid": "07150a520899461e956bd9f0a69ea82d",
    "domains": [
      "www.sangfor.com"
    ],
    "internetip": "192.168.1.1",
    "internalip": "192.168.1.1"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.1.4.4 添加Dnsmapping

功能说明: 添加Dnsmapping

1) API请求数据

POST <https://{host}/api/v1/namespaces/@namespace/dns mappings>

2) 输入参数

- Body参数

参数	类型	是否必选	描述
obj	object	是	dnsmap信息
obj.domains	string[]	是	域名
obj.internetip	string	是	单个IPv4 公网ip
obj.internalip	string	是	单个IPv4 内网ip

请求样例:

```
添加一条DNS-mappings数据

POST https://192.168.1.1/api/v1/namespaces/public/dns mappings
{
  "internalIp": "2.2.2.2",
  "domains": [
    "www.google.com"
  ],
  "internetIp": "1.1.1.1"
}
```

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	dnsmap信息
data.uuid	string	唯一标识id
data.domains	string[]	域名
data.internetIp	string	单个IPv4 公网ip
data.internalIp	string	单个IPv4 内网ip

请求成功样例:

```
{
  "data": {
    "itemsOffset": 0,
    "items": [
      {
        "internalIp": "2.2.2.2",
        "internetIp": "1.1.1.1",
        "uuid": "6D4A5E5690654A5798E3A8D8B29DA5E5",
        "domains": [
          "www.google.com"
        ]
      }
    ],
    "totalItems": 1,
    "pageNumber": 1,
    "itemLength": 1,
    "totalPages": 1,
    "pageSize": 100
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.1.4.5 批量删除Dnsmapping

功能说明: 批量删除Dnsmapping

1) API请求数据

POST

`https://{host}/api/batch/v1/namespaces/@namespace/dns mappings?_method=delete`

2) 输入参数

• Body参数

参数	类型	是否必选	描述
obj	object[]	是	dnsmapping配置数据
obj.domains	string[]	是	域名
obj.internetIp	string	是	单个IPv4 公网ip
obj.internallp	string	是	单个IPv4 内网ip

请求样例:

批量删除DNS-mappings数据,使用被删除策略的uuid

POST `https://192.168.1.1/api/batch/v1/namespaces/public/dns mappings?_method=DELETE`

```
[
  {
    "uuid": "6D4A5E5690654A5798E3A8D8B29DA5E5"
  },
  {
    "uuid": "4572AFE5B7A5461A9E27416B7A30DCA0"
  }
]
```

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	dnsmapping配置数据
data.uuid	string	唯一标识id
data.domains	string[]	域名
data.internetIp	string	单个IPv4 公网ip
data.internallp	string	单个IPv4 内网ip

请求成功样例:

```
{
  "data": [
    {
      "internallp": "2.2.2.2",
      "internetIp": "1.1.1.1",
      "uuid": "6D4A5E5690654A5798E3A8D8B29DA5E5",
      "domains": [
        "www.google.com"
      ]
    },
    {
      "internallp": "3.3.32.45",
      "internetIp": "2.2.2.2",
      "uuid": "4572AFE5B7A5461A9E27416B7A30DCA0",
      "domains": [
        "asd.com"
      ]
    }
  ]
}
```

```
}
},
"code": 0,
"message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.1.4.6 全量修改Dnsmapping

功能说明: 全量修改Dnsmapping

1) API请求数据

PUT <https://{host}/api/v1/namespaces/@namespace/dns mappings/@uuid>

2) 输入参数

• Body参数

参数	类型	是否必选	描述
obj	object	是	dnsmap信息
obj.domains	string[]	是	域名
obj.internetIp	string	是	单个IPv4 公网ip
obj.internallp	string	是	单个IPv4 内网ip

请求样例:

全量修改DNS-mappings数据

```
PUT https://192.168.1.1/api/v1/namespaces/public/dns mappings/F767E3A488AA4337909C9247B85F
D0C7
{
  "internallp": "3.3.3.3",
  "domains": [
    "asd.com"
  ],
  "internetIp": "2.2.2.21"
}
```

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	dnsmap信息
data.uuid	string	唯一标识id
data.domains	string[]	域名
data.internetIp	string	单个IPv4 公网ip
data.internallp	string	单个IPv4 内网ip

请求成功样例:


```
{
  "data": {
    "internallp": "3.3.3.3",
    "internetlp": "2.2.2.21",
    "uuid": "F767E3A488AA4337909C9247B85FD0C7",
    "domains": [
      "asd.com"
    ]
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.1.4.7 删除单条Dnsmapping

功能说明: 删除单条Dnsmapping

1) API请求数据

DELETE

https://{host}/api/v1/namespaces/@namespace/dnsmappings/@uuid

2) 输入参数

请求样例:

删除单条DNS-mappings数据

DELETE https://192.168.1.1/api/v1/namespaces/public/dnsmappings/F767E3A488AA4337909C9247B85FD0C7

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	dnsmap信息
data.uuid	string	唯一标识id
data.domains	string[]	域名
data.internetlp	string	单个IPv4 公网ip
data.internallp	string	单个IPv4 内网ip

请求成功样例:

```
{
  "data": {
    "internallp": "3.3.3.3",
    "internetlp": "2.2.2.21",
    "uuid": "F767E3A488AA4337909C9247B85FD0C7",
    "domains": [
      "asd.com"
    ]
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.2 访问控制

4.2.1 应用控制策略相关

4.2.1.1 应用控制策略

4.2.1.1.1 获取策略命中次数

功能说明：获取策略命中次数

1) API请求数据

GET https://{host}/api/v1/namespaces/@namespace/appcontrols/hits/@name

2) 输入参数

请求样例：

获取单条应用控制策略匹配命中次数，获取应用控制策略名为test的匹配命中信息

GET https://192.168.1.1/api/v1/namespaces/public/appcontrols/hits/test

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	策略命中次数信息
data.name	string	应用控制策略名称 最小长度:1 最大长度:95
data.hitCount	uint32	策略命中次数
data.lastHitTime	string	策略最近一次匹配命中时间
data.notHitDay	int32	策略持续未被匹配到的天数

请求成功样例：

```
{
  "data": {
    "hitCount": 36,
    "notHitDay": 0,
    "name": "test",
    "lastHitTime": "2020-07-09 13:33:20"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.2.1.1.2 批量获取策略命中次数

功能说明：批量获取策略命中次数

1) API请求数据

POST

https://{host}/api/batch/v1/namespaces/@namespace/appcontrols/hits?_method=get

2) 输入参数

• Body参数

参数	类型	是否必选	描述
objs	object[]	是	策略名称列表
objs.name	string	是	应用控制策略名称 最小长度:1 最大长度:95

请求样例：

获取多条指定应用控制策略匹配命中次数，获取应用控制策略名为test的匹配命中信息

```
POST https://192.168.1.1/api/batch/v1/namespaces/public/appcontrols/hits?_method=GET
[
  {
    "name": "test"
  }
]
```

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	策略命中次数信息列表
data.name	string	应用控制策略名称 最小长度:1 最大长度:95
data.hitCount	uint32	策略命中次数
data.lastHitTime	string	策略最近一次匹配命中时间
data.notHitDay	int32	策略持续未被匹配到的天数

请求成功样例：

```
{
  "data": [
    {
      "hitCount": 36,
      "notHitDay": 1339040416,
      "name": "test",
      "lastHitTime": "2020-07-09 13:33:20"
    }
  ]
}
```

```
    },
    "code": 0,
    "message": "成功"
  }
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.2.1.1.3 清除策略命中次数

功能说明: 清除策略命中次数

1) API请求数据

PATCH https://{host}/api/v1/namespaces/@namespace/appcontrols/hits/@name

2) 输入参数

• Body参数

参数	类型	是否必选	描述
obj	object	是	清除单个策略匹配数请求参数
obj.hitCount	uint8	否	策略命中次数 最大值:0
obj.isDefaultPolicy	bool	否	是否为默认策略

请求样例:

删除单条应用控制策略匹配命中次数, 清空应用控制策略名为test的匹配命中次数

```
PATCH https://192.168.1.1/api/v1/namespaces/public/appcontrols/hits/test
{
  "hitCount": 0
}
```

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	bool	

请求成功样例:

```
{
  "data": true,
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.2.1.1.4 批量清除策略命中次数

功能说明：批量清除策略命中次数

1) API请求数据

PATCH

https://{host}/api/batch/v1/namespaces/@namespace/appcontrols/hits

2) 输入参数

- Body参数

参数	类型	是否必选	描述
objs	object[]	是	批量清除策略的匹配数请求参数
objs.name	string	是	应用控制策略名称 最小长度:1 最大长度:95
objs.hitCount	uint8	否	策略命中次数 最大值:0
objs.isDefaultPolicy	bool	否	是否为默认策略

请求样例：

删除多条指定应用控制策略匹配命中次数，清空应用控制策略名为test的匹配命中次数

```
PATCH https://192.168.1.1/api/batch/v1/namespaces/public/appcontrols/hits
[
  {
    "name": "test",
    "hitCount": 0
  }
]
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	bool	

请求成功样例：

```
{
  "data": true,
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.2.1.1.5 开始同步策略冲突检查

功能说明：开始同步策略冲突检查

1) API请求数据

GET https://{host}/api/v1/namespaces/@namespace/appcontrols/policycheck

2) 输入参数

请求样例：

GET https://192.168.1.1/api/v1/namespaces/public/appcontrols/policycheck

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	bool	

请求成功样例：

```
{
  "message": "",
  "data": true,
  "code": 0
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.2.1.1.6 开始异步策略冲突检查

功能说明：开始异步策略冲突检查

1) API请求数据

POST https://{host}/api/v1/namespaces/@namespace/appcontrols/checks

2) 输入参数

- URL参数

参数	类型	是否必选	描述
sync	bool	否	是否为同步请求

- Body参数

参数	类型	是否必选	描述
obj	object	是	策略检查状态信息

obj.status	uint32	否	策略检查状态 枚举值:[0:未开始状态 1:运行状态 2:已完成状态]
obj.rate	uint32	否	策略检查进度百分比 最小值:0 最大值:100

请求样例:

```
执行应用控制策略失败检查
POST https://192.168.1.1/api/v1/namespaces/public/appcontrols/checks
[]
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	策略检查状态信息
data.status	uint32	策略检查状态 枚举值:[0:未开始状态 1:运行状态 2:已完成状态]
data.rate	uint32	策略检查进度百分比 最小值:0 最大值:100

请求成功样例:

```
{
  "data": {
    "status": 1,
    "rate": 0
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.2.1.1.7 获取策略冲突检查状态

功能说明: 获取策略冲突检查状态

1) API请求数据

GET

https://{host}/api/v1/namespaces/@namespace/appcontrols/checks

2) 输入参数

请求样例:

获取应用控制策略失败检查及进度
GET https://192.168.1.1/api/v1/namespaces/public/appcontrols/checks

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	策略检查状态信息
data.status	uint32	策略检查状态 枚举值: 0:未开始状态 1:运行状态 2:已完成状态]
data.rate	uint32	策略检查进度百分比 最小值:0 最大值:100

请求成功样例:

```
{
  "data": {
    "status": 2,
    "rate": 0
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.2.1.1.8 取消策略冲突检查

功能说明: 取消策略冲突检查

1) API请求数据

DELETE

https://{host}/api/v1/namespaces/@namespace/appcontrols/checks

2) 输入参数

请求样例:

取消应用控制策略失败
DELETE https://192.168.1.1/api/v1/namespaces/public/appcontrols/checks

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	策略检查状态信息
data.status	uint32	策略检查状态 枚举值: 0:未开始状态 1:运行状态 2:已完成状态]
data.rate	uint32	策略检查进度百分比 最小值:0 最大值:100

请求成功样例:

```
{
  "data": {
    "status": 1,
    "rate": 2
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.2.1.1.9 获取策略冲突列表

功能说明: 获取策略冲突列表

1) API请求数据

GET	https://{host}/api/v1/namespaces/@namespace/appcontrols/conflicts
-----	---

2) 输入参数

- URL 参数

参数	类型	是否必选	描述
_start	int32	否	起始位置, 从 0 开始 最小值:0 最大值:2000000
_search	string	否	模糊搜索关键字 最大长度:95

_length	int32	否	获取最大长度，限制最大值，如果数据不够可能返回小于 length 最小值:1 最大值:200 默认值:true
---------	-------	---	--

请求样例:

获取多条应用控制策略冲突配置信息，查询条件：起始位置为0，最多展示100条

GET https://192.168.1.1/api/v1/namespaces/public/appcontrols/conflicts?_start=0&_length=100

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	分页数据
data.totalItems	int32	总共多少项目
data.totalPages	int32	总共多少页
data.pageNumber	int32	当前页码，从 1 开始
data.pageSize	int32	每页多大
data.itemsOffset	int32	当前条目偏移, 从 0 开始
data.itemLength	int32	数据列表长度
data.items	object[]	有效数据列表
data.items.uuid	string	应用控制策略UUID
data.items.name	string	应用控制策略名称 最小长度:1 最大长度:95
data.items.conflictAcl	object[]	策略冲突列表
data.items.conflictAcl.uuid	string	应用控制策略UUID
data.items.conflictAcl.name	string	应用控制策略名称 最小长度:1 最大长度:95
data.items.conflictNat	object[]	与NAT策略冲突的策略列表
data.items.conflictNat.uuid	string	应用控制策略UUID
data.items.conflictNat.name	string	应用控制策略名称 最小长度:1 最大长度:95

data.items.healthStatus	string	应用控制策略健康状态 枚举值: APPCON_NORMAL:正常启用 APPCON_FORBID:禁用 APPCON_OUTTIME:时间过期 APPCON_NOT_MATCH_IN_30_DAYS:超过30天没有匹配 APPCON_NOT_MATCH_IN_7_DAYS:超过7天没有匹配 APPCON_CONFLICT_WITH_NAT:与nat策略冲突 APPCON_OTHERS_CONFLICT:与其他应用控制策略冲突 APPCON_SRC_INTERFACE_DOWN:源区域down APPCON_DST_INTERFACE_DOWN:目的区域down APPCON_ALL_INTERFACE_DOWN:源和目的区域都down APPCON_PARTIAL:策略健康状态为不完整策略 APPCON_APPLICATION_NOT_FOUND:策略健康状态为应用识别未找到 APPCON_DEFAULT_STATUS:默认策略状态]
-------------------------	--------	---

请求成功样例：

```
{
  "data": {
    "itemsOffset": 0,
    "items": [
      {
        "healthStatus": "APPCON_OTHERS_CONFLICT",
        "conflictNat": [],
        "name": "test",
        "uuid": "89FDEA5E61114BAF9C0B08DEE5B1699A",
        "conflictAcl": [
          {
            "uuid": "5BCC386ED6B240A8996D5095A2928E17",
            "name": "test_1"
          }
        ]
      }
    ],
    "totalItems": 1,
    "pageNumber": 1,
    "itemLength": 1,
    "totalPages": 1,
    "pageSize": 100
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.2.1.1.10 获取指定的策略冲突信息

功能说明：获取指定的策略冲突信息

1) API请求数据

GET

https://{host}/api/v1/namespaces/@namespace/appcontrols/conflicts/@name

2) 输入参数

请求样例：

获取单条应用控制策略冲突信息，获取应用控制策略名为test的冲突信息

GET https://192.168.1.1/api/v1/namespaces/public/appcontrols/conflicts/test

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	应用控制策略冲突列表
data.uuid	string	应用控制策略UUID
data.name	string	应用控制策略名称 最小长度:1 最大长度:95
data.conflictAcl	object[]	策略冲突列表
data.conflictAcl.uuid	string	应用控制策略UUID
data.conflictAcl.name	string	应用控制策略名称 最小长度:1 最大长度:95
data.conflictNat	object[]	与NAT策略冲突的策略列表
data.conflictNat.uuid	string	应用控制策略UUID
data.conflictNat.name	string	应用控制策略名称 最小长度:1 最大长度:95

data.healthStatus	string	应用控制策略健康状态 枚举值:[APPCON_NORMAL:正常启用 APPCON_FORBID:禁用 APPCON_OUTTIME:时间过期 APPCON_NOT_MATCH_IN_30_DAYS:超过30天没有匹配 APPCON_NOT_MATCH_IN_7_DAYS:超过7天没有匹配 APPCON_CONFLICT_WITH_NAT:与nat策略冲突 APPCON_OTHERS_CONFLICT:与其他应用控制策略冲突 APPCON_SRC_INTERFACE_DOWN:源区域down APPCON_DST_INTERFACE_DOWN:目的区域down APPCON_ALL_INTERFACE_DOWN:源和目的区域都down APPCON_PARTIAL:策略健康状态为不完整策略 APPCON_APPLICATION_NOT_FOUND:策略健康状态为应用识别未找到 APPCON_DEFAULT_STATUS:默认策略状态]
-------------------	--------	--

请求成功样例：

```
{
  "data": {
    "healthStatus": "APPCON_OTHERS_CONFLICT",
    "conflictNat": [],
    "name": "test",
    "uuid": "89FDEA5E61114BAF9C0B08DEE5B1699A",
    "conflictAcl": [
      {
        "uuid": "5BCC386ED6B240A8996D5095A2928E17",
        "name": "test_1"
      }
    ]
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.2.1.1.11 批量获取指定的策略冲突信息

功能说明：批量获取指定的策略冲突信息

1) API请求数据

POST	https://{host}/api/batch/v1/namespaces/@namespace/appcontrols/conflicts?_method=get
------	---

2) 输入参数

• Body参数

参数	类型	是否必选	描述
objs	object[]	是	应用控制策略冲突配置
objs.name	string	是	应用控制策略名称 最小长度:1 最大长度:95
objs.conflictAcl	object[]	是	策略冲突列表
objs.conflictAcl.name	string	是	应用控制策略名称 最小长度:1 最大长度:95
objs.conflictNat	object[]	是	与NAT策略冲突的策略列表
objs.conflictNat.name	string	是	应用控制策略名称 最小长度:1 最大长度:95
objs.healthStatus	string	否	应用控制策略健康状态 默认值:true 枚举值:[APPCON_NORMAL:正常启用 APPCON_FORBID:禁用 APPCON_OUTTIME:时间过期 APPCON_NOT_MATCH_IN_30_DAYS:超过30天没有匹配 APPCON_NOT_MATCH_IN_7_DAYS:超过7天没有匹配 APPCON_CONFLICT_WITH_NAT:与nat策略冲突 APPCON_OTHERS_CONFLICT:与其他应用控制策略冲突 APPCON_SRC_INTERFACE_DOWN:源区域down APPCON_DST_INTERFACE_DOWN:目的区域down APPCON_ALL_INTERFACE_DOWN:源和目的区域都down APPCON_PARTIAL:策略健康状态为不完整策略 APPCON_APPLICATION_NOT_FOUND:策略健康状态为应用识别未找到 APPCON_DEFAULT_STATUS:默认策略状态]

请求样例:

获取指定的多条应用控制策略冲突信息，获取应用控制策略名为test的冲突信息

```
POST https://192.168.1.1/api/batch/v1/namespaces/public/appcontrols/conflicts?_method=GET
[
  {
    "name": "test"
  }
]
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	应用控制策略冲突配置
data.uuid	string	应用控制策略UUID
data.name	string	应用控制策略名称 最小长度:1 最大长度:95
data.conflictAcl	object[]	策略冲突列表
data.conflictAcl.uuid	string	应用控制策略UUID
data.conflictAcl.name	string	应用控制策略名称 最小长度:1 最大长度:95
data.conflictNat	object[]	与NAT策略冲突的策略列表
data.conflictNat.uuid	string	应用控制策略UUID
data.conflictNat.name	string	应用控制策略名称 最小长度:1 最大长度:95
data.healthStatus	string	应用控制策略健康状态 枚举值: APPCON_NORMAL:正常启用 APPCON_FORBID:禁用 APPCON_OUTTIME:时间过期 APPCON_NOT_MATCH_IN_30_DAYS:超过30天没有匹配 APPCON_NOT_MATCH_IN_7_DAYS:超过7天没有匹配 APPCON_CONFLICT_WITH_NAT:与nat策略冲突 APPCON_OTHERS_CONFLICT:与其他应用控制策略冲突 APPCON_SRC_INTERFACE_DOWN:源区域down APPCON_DST_INTERFACE_DOWN:目的区域down APPCON_ALL_INTERFACE_DOWN:源和目的区域都down APPCON_PARTIAL:策略健康状态为不完整策略 APPCON_APPLICATION_NOT_FOUND:策略健康状态为应用识别未找到 APPCON_DEFAULT_STATUS:默认策略状态]

请求成功样例:

```
{
  "data": [
    {
      "healthStatus": "APPCON_OTHERS_CONFLICT",
      "conflictNat": [],

```

```
{
  "name": "test",
  "uuid": "89FDEA5E61114BAF9C0B08DEE5B1699A",
  "conflictAcl": [
    {
      "uuid": "5BCC386ED6B240A8996D5095A2928E17",
      "name": "test_1"
    }
  ]
},
{
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.2.1.1.12 获取应用控制策略列表

功能说明：获取应用控制策略列表

1) API请求数据

GET

https://{host}/api/v1/namespaces/@namespace/appcontrols/policys

2) 输入参数

- URL 参数

参数	类型	是否必选	描述
srcIp	string	否	匹配源ip地址
id	uint32	否	过滤id
policyGroup	string	否	过滤策略组 最小长度:1 最大长度:95
uuid	string	否	资源全局唯一ID 最大长度:32
_sortBy	string	否	指定排序字段 最大长度:100
label	string	否	过滤标签 最小长度:1 最大长度:95
actionTime	string	否	匹配时间
action	uint32	否	过滤动作 枚举值:[1:动作允许 0:动作拒绝]
dstIp	string	否	匹配目的ip地址
_start	int32	否	起始位置, 从 0 开始 最小值:0 最大值:2000000

dstPort	uint16	否	匹配目的端口 最小值:0 最大值:65535
srcZone	string	否	过滤或匹配源区域 最小长度:1 最大长度:95
_length	int32	否	获取最大长度，限制最大值，如果数据不够可能返回小于 length 最小值:1 最大值:200 默认值:true
protoNum	uint8	否	匹配协议 最小值:0 最大值:255
srcPort	uint16	否	匹配源端口 最小值:0 最大值:65535
enable	bool	否	过滤启禁用
invalidStatus	bool	否	过滤失效状态策略
_select	string	否	选择字段, 字段用逗号分隔 最大长度:1000
_order	string	否	指定排序方式（正序/倒序） 默认值:true 枚举值:[asc:按升序排列 desc:按降序排列]
_search	string	否	模糊搜索关键字 最大长度:95
name	string	否	资源名 最大长度:1440
dstZone	string	否	过滤或匹配目的区域 最小长度:1 最大长度:95

请求样例:

获取多条应用控制策略配置信息，查询条件：起始位置为0，最多展示100条

GET https://192.168.1.1/api/v1/namespaces/public/appcontrols/policys?_start=0&_length=100

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	分页数据
data.totalItems	int32	总共多少项目

data.totalPages	int32	总共多少页
data.pageNumber	int32	当前页码, 从 1 开始
data.pageSize	int32	每页多大
data.itemsOffset	int32	当前条目偏移, 从 0 开始
data.itemLength	int32	数据列表长度
data.items	object[]	有效数据列表
data.items.uuid	string	唯一标识id
data.items.name	string	应用控制策略名称 最小长度:1 最大长度:95
data.items.showname	string	策略显示名称 最小长度:1 最大长度:95
data.items.description	string	应用控制策略描述 最小长度:0 最大长度:256
data.items.position	int32	应用控制策略位置
data.items.enable	bool	应用控制策略启禁用
data.items.updateTime	string	应用控制策略更新时间
data.items.durationTime	string	应用控制策略最近一次状态更新时间
data.items.group	string	应用控制策略组
data.items.labels	string[]	应用控制策略标签列表
data.items.src	object	应用控制策略源地址
data.items.src.srcZones	string[]	源区域列表
data.items.src.srcAddrs	object	源地址信息
data.items.src.srcAddrs.srcAddrType	string	地址类型 枚举值:[NETOBJECT:ip组 USERGROUP:用户及用户组 MAC:mac地址]
data.items.src.srcAddrs.srcMacAddrs	string[]	mac地址列表
data.items.src.srcAddrs.users	string[]	用户列表
data.items.src.srcAddrs.userGroups	string[]	用户组列表
data.items.src.srcAddrs.srcIpGroups	string[]	ip组列表
data.items.dst	object	应用控制策略目的地址
data.items.dst.dstZones	string[]	引用目的区域列表

data.items.dst.dstAddrs	object	目的地址
data.items.dst.dstAddrs. dstAddrType	string	地址类型 枚举值:[NETOBJECT:ip组 MAC:mac地址 DOMAIN:域名]
data.items.dst.dstAddrs. dstMacAddrs	string[]	mac地址列表
data.items.dst.dstAddrs. domains	string[]	域名列表
data.items.dst.dstAddrs. dstIpGroups	string[]	ip组列表
data.items.dst.services	string[]	服务引用列表
data.items.dst.applications	string[]	应用引用列表
data.items.action	uint32	应用控制策略行为，允许或拒绝 枚举值:[1:允许 0:拒绝]
data.items.schedule	string	应用控制策略起作用时间
data.items.advanceOption	object	应用控制策略高级选项
data.items.advanceOption.logEnable	bool	是否开启应用控制命中日志记录
data.items.advanceOption.autoSynDNS	bool	是否开启主动查询域名
data.items.advanceOption.keepAlive	uint8	设置长连接
data.items.reason	string	应用控制策略变更原因 最小长度:0 最大长度:256
data.items.isdefault	bool	是否是默认策略
data.items.lastHitTime	string	策略最近一次匹配命中时间
data.items.hits	uint64	应用控制策略命中匹配次数

请求成功样例：

```
{
  "data": {
    "itemsOffset": 0,
    "items": [
      {
        "enable": true,
        "updateTime": "2020-07-09 12:06:57",
        "uuid": "D05035431A434AC895D7B8396A8CB4BC",
        "group": "默认策略组",
        "lastHitTime": "2020-07-09 12:46:03",
        "action": 1,
        "dst": {
          "dstZones": [],
          "dstAddrs": {
```

```

        "dstAddrType": "NETOBJECT",
        "dstIpGroups": [
            "全部"
        ]
    },
    "applications": [
        "全部"
    ],
    "services": [
        "any"
    ]
},
"position": 0,
"src": {
    "srcZones": [],
    "srcAddrs": {
        "srcAddrType": "NETOBJECT",
        "srcIpGroups": [
            "全部"
        ]
    }
},
"name": "test",
"description": "",
"hits": 45,
"labels": [],
"isdefault": false,
"durationTime": "2020-07-09 12:06:57",
"schedule": "全天",
"advanceOption": {
    "autoSynDNS": false,
    "logEnable": false,
    "keepAlive": 0
}
},
{
    "enable": true,
    "updateTime": "2020-07-09 13:41:10",
    "uuid": "00000000010000000000000000000000",
    "showname": "默认策略",
    "group": "默认策略组",
    "lastHitTime": "1970-01-01 08:00:00",
    "action": 0,
    "dst": {
        "dstZones": [],
        "dstAddrs": {
            "dstAddrType": "NETOBJECT",
            "dstIpGroups": [
                "全部"
            ]
        }
    },
    "applications": [
        "全部"
    ],
    "services": [
        "any"
    ]
},
"position": 1,
"src": {
    "srcZones": [],
    "srcAddrs": {
        "srcAddrType": "NETOBJECT",
        "srcIpGroups": [
            "全部"
        ]
    }
},
"name": "Default Policy",
"description": "",
"hits": 0,
"labels": [
    "默认"
],
"isdefault": true,
"durationTime": "2020-06-29 11:29:29",
"schedule": "全天",

```

```
        "advanceOption": {
            "autoSynDNS": false,
            "logEnable": false,
            "keepAlive": 0
        }
    },
    "totalItems": 2,
    "pageNumber": 1,
    "itemLength": 2,
    "totalPages": 1,
    "pageSize": 100
},
"code": 0,
"message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.2.1.1.13 获取指定应用控制策略

功能说明: 获取指定应用控制策略

1) API请求数据

GET https://{host}/api/v1/namespaces/@namespace/appcontrols/policys/@name

2) 输入参数

• URL参数

参数	类型	是否必选	描述
_select	string	否	选择字段, 字段用逗号分隔 最大长度:1000

请求样例:

获取单条应用控制策略, 获取名称为test的应用控制策略信息

GET https://192.168.1.1/api/v1/namespaces/public/appcontrols/policys/test

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	应用控制策略
data.uuid	string	唯一标识id
data.name	string	应用控制策略名称 最小长度:1 最大长度:95
data.showname	string	策略显示名称 最小长度:1 最大长度:95

data.description	string	应用控制策略描述 最小长度:0 最大长度:256
data.position	int32	应用控制策略位置
data.enable	bool	应用控制策略启禁用
data.updateTime	string	应用控制策略更新时间
data.durationTime	string	应用控制策略最近一次状态更新时间
data.group	string	应用控制策略组
data.labels	string[]	应用控制策略标签列表
data.src	object	应用控制策略源地址
data.src.srcZones	string[]	源区域列表
data.src.srcAddrs	object	源地址信息
data.src.srcAddrs.srcAddrType	string	地址类型 枚举值:[NETOBJECT:ip组 USERGROUP:用户及用户组 MAC:mac地址]
data.src.srcAddrs.srcMacAddrs	string[]	mac地址列表
data.src.srcAddrs.users	string[]	用户列表
data.src.srcAddrs.userGroups	string[]	用户组列表
data.src.srcAddrs.srcIpGroups	string[]	ip组列表
data.dst	object	应用控制策略目的地址
data.dst.dstZones	string[]	引用目的区域列表
data.dst.dstAddrs	object	目的地址
data.dst.dstAddrs.dstAddrType	string	地址类型 枚举值:[NETOBJECT:ip组 MAC:mac地址 DOMAIN:域名]
data.dst.dstAddrs.dstMacAddrs	string[]	mac地址列表
data.dst.dstAddrs.domains	string[]	域名列表
data.dst.dstAddrs.dstIpGroups	string[]	ip组列表
data.dst.services	string[]	服务引用列表
data.dst.applications	string[]	应用引用列表

data.action	uint32	应用控制策略行为，允许或拒绝 枚举值:[1:允许 0:拒绝]
data.schedule	string	应用控制策略起作用时间
data.advanceOption	object	应用控制策略高级选项
data.advanceOption.log Enable	bool	是否开启应用控制命中日志记录
data.advanceOption.aut oSynDNS	bool	是否开启主动查询域名
data.advanceOption.ke epAlive	uint8	设置长连接
data.reason	string	应用控制策略变更原因 最小长度:0 最大长度:256
data.isdefault	bool	是否是默认策略
data.lastHitTime	string	策略最近一次匹配命中时间
data.hits	uint64	应用控制策略命中匹配次数

请求成功样例：

```
{
  "data": {
    "enable": true,
    "updateTime": "2020-07-09 12:06:57",
    "uuid": "D05035431A434AC895D7B8396A8CB4BC",
    "group": "默认策略组",
    "lastHitTime": "2020-07-09 12:46:03",
    "action": 1,
    "dst": {
      "dstZones": [],
      "dstAddrs": {
        "dstAddrType": "NETOBJECT",
        "dstIpGroups": [
          "全部"
        ]
      }
    },
    "applications": [
      "全部"
    ],
    "services": [
      "any"
    ]
  },
  "position": 0,
  "src": {
    "srcZones": [],
    "srcAddrs": {
      "srcAddrType": "NETOBJECT",
      "srcIpGroups": [
        "全部"
      ]
    }
  },
  "name": "test",
  "description": "",
  "hits": 45,
  "labels": [],
  "isdefault": false,
  "durationTime": "2020-07-09 12:06:57",
  "schedule": "全天",
  "advanceOption": {
    "autoSynDNS": false,

```

```

        "logEnable": false,
        "keepAlive": 0
    },
    "code": 0,
    "message": "成功"
}

```

请求失败样例:

```

{
  "data": "",
  "code": 22,
  "message": "无效参数"
}

```

4.2.1.1.14 增加单项应用控制策略

功能说明: 增加单项应用控制策略

1) API请求数据

POST <https://{host}/api/v1/namespaces/@namespace/appcontrols/policys>

2) 输入参数

• URL参数

参数	类型	是否必选	描述
_where	string	否	指定位置, 用于移动/插入 枚举值:[top:顶部 bottom:底部 before:在目标位置之前 after:在目标位置之后]
_dest	string	否	目标位置关键字对应的值 最大长度:100
_key	string	否	目标位置关键字 枚举值:[position:指定目标位置关键字为 position name:指定目标位置关键字为 name]

• Body参数

参数	类型	是否必选	描述
obj	object	是	应用控制策略
obj.name	string	是	应用控制策略名称 最小长度:1 最大长度:95
obj.showname	string	否	策略显示名称 最小长度:1 最大长度:95
obj.description	string	否	应用控制策略描述 最小长度:0 最大长度:256
obj.position	int32	否	应用控制策略位置

obj.enable	bool	否	应用控制策略启禁用
obj.updateTime	string	否	应用控制策略更新时间
obj.durationTime	string	否	应用控制策略最近一次状态更新时间
obj.group	string	是	应用控制策略组
obj.labels	string[]	否	应用控制策略标签列表
obj.src	object	否	应用控制策略源地址
obj.src.srcZones	string[]	否	源区域列表
obj.src.srcAddrs	object	否	源地址信息
obj.src.srcAddrs.srcAddrType	string	否	地址类型 默认值:true 枚举值:[NETOBJECT:ip组 USERGROUP:用户及用户组 MAC:mac地址]
obj.src.srcAddrs.srcMacAddrs	string[]	否	mac地址列表
obj.src.srcAddrs.users	string[]	否	用户列表
obj.src.srcAddrs.userGroups	string[]	否	用户组列表
obj.src.srcAddrs.srcIpGroups	string[]	否	ip组列表
obj.dst	object	否	应用控制策略目的地址
obj.dst.dstZones	string[]	否	引用目的区域列表
obj.dst.dstAddrs	object	否	目的地址
obj.dst.dstAddrs.dstAddrType	string	否	地址类型 默认值:true 枚举值:[NETOBJECT:ip组 MAC:mac地址 DOMAIN:域名]
obj.dst.dstAddrs.dstMacAddrs	string[]	否	mac地址列表
obj.dst.dstAddrs.domains	string[]	否	域名列表
obj.dst.dstAddrs.dstIpGroups	string[]	否	ip组列表
obj.dst.services	string[]	否	服务引用列表
obj.dst.applications	string[]	否	应用引用列表

obj.action	uint32	否	应用控制策略行为，允许或拒绝 默认值:true 枚举值:[1:允许 0:拒绝]
obj.schedule	string	否	应用控制策略起作用时间
obj.advanceOption	object	否	应用控制策略高级选项
obj.advanceOption.logEnable	bool	否	是否开启应用控制命中日志记录
obj.advanceOption.autoSynDNS	bool	否	是否开启主动查询域名
obj.advanceOption.keepAlive	uint8	否	设置长连接
obj.reason	string	否	应用控制策略变更原因 最小长度:0 最大长度:256
obj.isdefault	bool	否	是否是默认策略
obj.lastHitTime	string	否	策略最近一次匹配命中时间

请求样例:

新增单条应用控制策略，新增策略名为test的应用控制策略

POST https://192.168.1.1/api/v1/namespaces/public/appcontrols/policys

```
{
  "enable": true,
  "src": {
    "srcZones": [],
    "srcAdrs": {
      "srcAddrType": "NETOBJECT",
      "srcIpGroups": [
        "全部"
      ]
    }
  },
  "schedule": "全天",
  "description": "",
  "labels": [],
  "group": "默认策略组",
  "action": 1,
  "name": "test",
  "dst": {
    "dstZones": [],
    "dstAdrs": {
      "dstAddrType": "NETOBJECT",
      "dstIpGroups": [
        "全部"
      ]
    }
  },
  "applications": [
    "全部"
  ],
  "services": [
    "any"
  ]
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	应用控制策略
data.uuid	string	唯一标识id
data.name	string	应用控制策略名称 最小长度:1 最大长度:95
data.showname	string	策略显示名称 最小长度:1 最大长度:95
data.description	string	应用控制策略描述 最小长度:0 最大长度:256
data.position	int32	应用控制策略位置
data.enable	bool	应用控制策略启禁用
data.updateTime	string	应用控制策略更新时间
data.durationTime	string	应用控制策略最近一次状态更新时间
data.group	string	应用控制策略组
data.labels	string[]	应用控制策略标签列表
data.src	object	应用控制策略源地址
data.src.srcZones	string[]	源区域列表
data.src.srcAddrs	object	源地址信息
data.src.srcAddrs.srcAddrType	string	地址类型 枚举值:[NETOBJECT:ip组 USERGROUP:用户及用户组 MAC:mac地址]
data.src.srcAddrs.srcMacAddrs	string[]	mac地址列表
data.src.srcAddrs.users	string[]	用户列表
data.src.srcAddrs.userGroups	string[]	用户组列表
data.src.srcAddrs.srcIpGroups	string[]	ip组列表
data.dst	object	应用控制策略目的地址
data.dst.dstZones	string[]	引用目的区域列表
data.dst.dstAddrs	object	目的地址

data.dst.dstAddrs.dstAddrType	string	地址类型 枚举值:[NETOBJECT:ip组 MAC:mac地址 DOMAIN:域名]
data.dst.dstAddrs.dstMacAddrs	string[]	mac地址列表
data.dst.dstAddrs.domains	string[]	域名列表
data.dst.dstAddrs.dstIpGroups	string[]	ip组列表
data.dst.services	string[]	服务引用列表
data.dst.applications	string[]	应用引用列表
data.action	uint32	应用控制策略行为, 允许或拒绝 枚举值:[1:允许 0:拒绝]
data.schedule	string	应用控制策略起作用时间
data.advanceOption	object	应用控制策略高级选项
data.advanceOption.logEnable	bool	是否开启应用控制命中日志记录
data.advanceOption.autoSynDNS	bool	是否开启主动查询域名
data.advanceOption.keepAlive	uint8	设置长连接
data.reason	string	应用控制策略变更原因 最小长度:0 最大长度:256
data.isdefault	bool	是否是默认策略
data.lastHitTime	string	策略最近一次匹配命中时间
data.hits	uint64	应用控制策略命中匹配次数

请求成功样例:

```
{
  "data": {
    "enable": true,
    "updateTime": "2020-07-09 13:55:29",
    "dst": {
      "dstZones": [],
      "dstAddrs": {
        "dstAddrType": "NETOBJECT",
        "dstIpGroups": [
          "全部"
        ]
      }
    },
    "applications": [
      "全部"
    ],
    "services": [
      "any"
    ]
  }
}
```

```
    },
    "durationTime": "2020-07-09 13:55:29",
    "uuid": "CB26CA52C5404C0D91ABF038D563EDBE",
    "src": {
      "srcZones": [],
      "srcAddrs": {
        "srcAddrType": "NETOBJECT",
        "srcIpGroups": [
          "全部"
        ]
      }
    },
    "schedule": "全天",
    "group": "Default Policy Group",
    "hits": 0,
    "labels": [],
    "isdefault": false,
    "action": 1,
    "name": "test",
    "description": ""
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.2.1.1.15 删除单项应用控制策略

功能说明：删除单项应用控制策略

1) API请求数据

DELETE

https://{host}/api/v1/namespaces/@namespace/appcontrols/policys/@name

2) 输入参数

请求样例：

删除单条应用控制策略，删除策略名称为test的应用控制策略

DELETE https://192.168.1.1/api/v1/namespaces/public/appcontrols/policys/test

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	应用控制策略
data.uuid	string	唯一标识id
data.name	string	应用控制策略名称 最小长度:1 最大长度:95
data.showname	string	策略显示名称 最小长度:1 最大长度:95

data.description	string	应用控制策略描述 最小长度:0 最大长度:256
data.position	int32	应用控制策略位置
data.enable	bool	应用控制策略启禁用
data.updateTime	string	应用控制策略更新时间
data.durationTime	string	应用控制策略最近一次状态更新时间
data.group	string	应用控制策略组
data.labels	string[]	应用控制策略标签列表
data.src	object	应用控制策略源地址
data.src.srcZones	string[]	源区域列表
data.src.srcAddrs	object	源地址信息
data.src.srcAddrs.srcAddrType	string	地址类型 枚举值:[NETOBJECT:ip组 USERGROUP:用户及用户组 MAC:mac地址]
data.src.srcAddrs.srcMacAddrs	string[]	mac地址列表
data.src.srcAddrs.users	string[]	用户列表
data.src.srcAddrs.userGroups	string[]	用户组列表
data.src.srcAddrs.srcIpGroups	string[]	ip组列表
data.dst	object	应用控制策略目的地址
data.dst.dstZones	string[]	引用目的区域列表
data.dst.dstAddrs	object	目的地址
data.dst.dstAddrs.dstAddrType	string	地址类型 枚举值:[NETOBJECT:ip组 MAC:mac地址 DOMAIN:域名]
data.dst.dstAddrs.dstMacAddrs	string[]	mac地址列表
data.dst.dstAddrs.domains	string[]	域名列表
data.dst.dstAddrs.dstIpGroups	string[]	ip组列表
data.dst.services	string[]	服务引用列表
data.dst.applications	string[]	应用引用列表

data.action	uint32	应用控制策略行为，允许或拒绝 枚举值: 1:允许 0:拒绝]
data.schedule	string	应用控制策略起作用时间
data.advanceOption	object	应用控制策略高级选项
data.advanceOption.log Enable	bool	是否开启应用控制命中日志记录
data.advanceOption.autoSynDNS	bool	是否开启主动查询域名
data.advanceOption.keepAlive	uint8	设置长连接
data.reason	string	应用控制策略变更原因 最小长度:0 最大长度:256
data.isdefault	bool	是否是默认策略
data.lastHitTime	string	策略最近一次匹配命中时间
data.hits	uint64	应用控制策略命中匹配次数

请求成功样例：

```
{
  "data": {
    "enable": true,
    "updateTime": "2020-07-09 12:06:57",
    "advanceOption": {
      "autoSynDNS": false,
      "logEnable": false,
      "keepAlive": 0
    },
    "dst": {
      "dstZones": [],
      "dstAddrs": {
        "dstAddrType": "NETOBJECT",
        "dstIpGroups": [
          "全部"
        ]
      },
      "applications": [
        "全部"
      ],
      "services": [
        "any"
      ]
    },
    "durationTime": "2020-07-09 12:06:57",
    "uuid": "D05035431A434AC895D7B8396A8CB4BC",
    "src": {
      "srcZones": [],
      "srcAddrs": {
        "srcAddrType": "NETOBJECT",
        "srcIpGroups": [
          "全部"
        ]
      }
    },
    "schedule": "全天",
    "group": "Default Policy Group",
    "hits": 0,
    "labels": [],
    "isdefault": false,
    "action": 1,
    "name": "test",
  }
}
```

```

    "description": ""
  },
  "code": 0,
  "message": "成功"
}

```

请求失败样例:

```

{
  "data": "",
  "code": 22,
  "message": "无效参数"
}

```

4.2.1.1.16 批量删除应用控制策略

功能说明: 批量删除应用控制策略

1) API请求数据

POST

https://{host}/api/batch/v1/namespaces/@namespace/appcontrols/policys?
_method=delete

2) 输入参数

• Body参数

参数	类型	是否必选	描述
objs	object[]	是	应用控制策略列表
objs.name	string	是	应用控制策略名称 最小长度:1 最大长度:95
objs.showname	string	否	策略显示名称 最小长度:1 最大长度:95
objs.description	string	否	应用控制策略描述 最小长度:0 最大长度:256
objs.position	int32	否	应用控制策略位置
objs.enable	bool	否	应用控制策略启禁用
objs.updateTime	string	否	应用控制策略更新时间
objs.durationTime	string	否	应用控制策略最近一次状态更新时间
objs.group	string	是	应用控制策略组
objs.labels	string[]	否	应用控制策略标签列表
objs.src	object	否	应用控制策略源地址
objs.src.srcZones	string[]	否	源区域列表
objs.src.srcAddrs	object	否	源地址信息
objs.src.srcAddrs.srcAddrType	string	否	地址类型 默认值:true 枚举值:[NETOBJECT:ip组 USERGROUP:用户及用户组 MAC:mac地址]

objs.src.srcAddrs.srcMacAddrs	string[]	否	mac地址列表
objs.src.srcAddrs.users	string[]	否	用户列表
objs.src.srcAddrs.userGroups	string[]	否	用户组列表
objs.src.srcAddrs.srcIpGroups	string[]	否	ip组列表
objs.dst	object	否	应用控制策略目的地址
objs.dst.dstZones	string[]	否	引用目的区域列表
objs.dst.dstAddrs	object	否	目的地址
objs.dst.dstAddrs.dstAddrType	string	否	地址类型 默认值:true 枚举值:[NETOBJECT:ip组 MAC:mac地址 DOMAIN:域名]
objs.dst.dstAddrs.dstMacAddrs	string[]	否	mac地址列表
objs.dst.dstAddrs.domains	string[]	否	域名列表
objs.dst.dstAddrs.dstIpGroups	string[]	否	ip组列表
objs.dst.services	string[]	否	服务引用列表
objs.dst.applications	string[]	否	应用引用列表
objs.action	uint32	否	应用控制策略行为，允许或拒绝 默认值:true 枚举值:[1:允许 0:拒绝]
objs.schedule	string	否	应用控制策略起作用时间
objs.advanceOption	object	否	应用控制策略高级选项
objs.advanceOption.logEnable	bool	否	是否开启应用控制命中日志记录
objs.advanceOption.autoSynDNS	bool	否	是否开启主动查询域名
objs.advanceOption.keepAlive	uint8	否	设置长连接
objs.reason	string	否	应用控制策略变更原因 最小长度:0 最大长度:256
objs.isdefault	bool	否	是否是默认策略
objs.lastHitTime	string	否	策略最近一次匹配命中时间

请求样例：

删除多条应用控制策略，删除策略名称为test的应用控制策略

```
POST https://192.168.1.1/api/batch/v1/namespaces/public/appcontrols/policys?_method=DELETE
[
  {
    "name": "test"
  }
]
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	应用控制策略列表
data.uuid	string	唯一标识id
data.name	string	应用控制策略名称 最小长度:1 最大长度:95
data.showname	string	策略显示名称 最小长度:1 最大长度:95
data.description	string	应用控制策略描述 最小长度:0 最大长度:256
data.position	int32	应用控制策略位置
data.enable	bool	应用控制策略启禁用
data.updateTime	string	应用控制策略更新时间
data.durationTime	string	应用控制策略最近一次状态更新时间
data.group	string	应用控制策略组
data.labels	string[]	应用控制策略标签列表
data.src	object	应用控制策略源地址
data.src.srcZones	string[]	源区域列表
data.src.srcAddrs	object	源地址信息
data.src.srcAddrs.srcAddrType	string	地址类型 枚举值: NETOBJECT:ip组 USERGROUP:用户及用户组 MAC:mac地址]
data.src.srcAddrs.srcMacAddrs	string[]	mac地址列表
data.src.srcAddrs.users	string[]	用户列表
data.src.srcAddrs.userGroups	string[]	用户组列表
data.src.srcAddrs.srcIpGroups	string[]	ip组列表

data.dst	object	应用控制策略目的地址
data.dst.dstZones	string[]	引用目的区域列表
data.dst.dstAddrs	object	目的地址
data.dst.dstAddrs.dstAddrType	string	地址类型 枚举值:[NETOBJECT:ip组 MAC:mac地址 DOMAIN:域名]
data.dst.dstAddrs.dstMacAddrs	string[]	mac地址列表
data.dst.dstAddrs.domains	string[]	域名列表
data.dst.dstAddrs.dstIpGroups	string[]	ip组列表
data.dst.services	string[]	服务引用列表
data.dst.applications	string[]	应用引用列表
data.action	uint32	应用控制策略行为，允许或拒绝 枚举值:[1:允许 0:拒绝]
data.schedule	string	应用控制策略起作用时间
data.advanceOption	object	应用控制策略高级选项
data.advanceOption.logEnable	bool	是否开启应用控制命中日志记录
data.advanceOption.autoSynDNS	bool	是否开启主动查询域名
data.advanceOption.keepAlive	uint8	设置长连接
data.reason	string	应用控制策略变更原因 最小长度:0 最大长度:256
data.isdefault	bool	是否是默认策略
data.lastHitTime	string	策略最近一次匹配命中时间
data.hits	uint64	应用控制策略命中匹配次数

请求成功样例：

```
{
  "data": [
    {
      "enable": true,
      "updateTime": "2020-07-09 13:56:11",
      "dst": {
        "dstZones": [],
        "dstAddrs": {
          "dstAddrType": "NETOBJECT",
          "dstIpGroups": [
            "全部"
          ]
        }
      }
    }
  ]
}
```

```
    },
    "applications": [
      "全部"
    ],
    "services": [
      "any"
    ]
  },
  "durationTime": "2020-07-09 13:55:29",
  "uuid": "CB26CA52C5404C0D91ABF038D563EDBE",
  "src": {
    "srcZones": [],
    "srcAddrs": {
      "srcAddrType": "NETOBJECT",
      "srcIpGroups": [
        "全部"
      ]
    }
  },
  "schedule": "全天",
  "group": "Default Policy Group",
  "hits": 0,
  "labels": [],
  "isdefault": false,
  "action": 1,
  "name": "test",
  "description": ""
},
{
  "code": 0,
  "message": "成功"
}
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.2.1.1.17 修改单项应用控制策略

功能说明: 修改单项应用控制策略

1) API请求数据

PATCH

https://{host}/api/v1/namespaces/@namespace/appcontrols/policys/@name

2) 输入参数

- URL 参数

参数	类型	是否必选	描述
_arrayop	string	否	对象属性是数组的操作控制 枚举值:[add:添加对应的数组元素 remove:删除对应的数组元素]
_key	string	否	目标位置关键字 枚举值:[position:指定目标位置关键字为 position name:指定目标位置关键字为 name]
_dest	string	否	目标位置关键字对应的值 最大长度:100

_where	string	否	指定位置, 用于移动/插入 枚举值:[top:顶部 bottom:底部 before:在目标位置之前 after:在目标位置之后]
--------	--------	---	---

- Body参数

参数	类型	是否必选	描述
obj	object	是	应用控制策略
obj.name	string	是	应用控制策略名称 最小长度:1 最大长度:95
obj.showname	string	否	策略显示名称 最小长度:1 最大长度:95
obj.description	string	否	应用控制策略描述 最小长度:0 最大长度:256
obj.position	int32	否	应用控制策略位置
obj.enable	bool	否	应用控制策略启禁用
obj.updateTime	string	否	应用控制策略更新时间
obj.durationTime	string	否	应用控制策略最近一次状态更新时间
obj.group	string	是	应用控制策略组
obj.labels	string[]	否	应用控制策略标签列表
obj.src	object	否	应用控制策略源地址
obj.src.srcZones	string[]	否	源区域列表
obj.src.srcAddrs	object	否	源地址信息
obj.src.srcAddrs.srcAddrType	string	否	地址类型 默认值:true 枚举值:[NETOBJECT:ip组 USERGROUP:用户及用户组 MAC:mac地址]
obj.src.srcAddrs.srcMacAddrs	string[]	否	mac地址列表
obj.src.srcAddrs.users	string[]	否	用户列表
obj.src.srcAddrs.userGroups	string[]	否	用户组列表
obj.src.srcAddrs.srcIpGroups	string[]	否	ip组列表
obj.dst	object	否	应用控制策略目的地址
obj.dst.dstZones	string[]	否	引用目的区域列表

obj.dst.dstAddr	object	否	目的地址
obj.dst.dstAddr.dstAddrType	string	否	地址类型 默认值:true 枚举值:[NETOBJECT:ip组 MAC:mac地址 DOMAIN:域名]
obj.dst.dstAddr.dstMacAddr	string[]	否	mac地址列表
obj.dst.dstAddr.domains	string[]	否	域名列表
obj.dst.dstAddr.dstIpGroups	string[]	否	ip组列表
obj.dst.services	string[]	否	服务引用列表
obj.dst.applications	string[]	否	应用引用列表
obj.action	uint32	否	应用控制策略行为，允许或拒绝 默认值:true 枚举值:[1:允许 0:拒绝]
obj.schedule	string	否	应用控制策略起作用时间
obj.advanceOption	object	否	应用控制策略高级选项
obj.advanceOption.logEnable	bool	否	是否开启应用控制命中日志记录
obj.advanceOption.autoSynDNS	bool	否	是否开启主动查询域名
obj.advanceOption.keepAlive	uint8	否	设置长连接
obj.reason	string	否	应用控制策略变更原因 最小长度:0 最大长度:256
obj.isdefault	bool	否	是否是默认策略
obj.lastHitTime	string	否	策略最近一次匹配命中时间

请求样例：

修改单条应用控制策略，修改名为test的应用控制策略名字为test_1，其余不改

```
PATCH https://192.168.1.1/api/v1/namespaces/public/appcontrols/policys/test
{
  "name": "test_1"
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码

message	string	错误信息
data	object	应用控制策略
data.uuid	string	唯一标识id
data.name	string	应用控制策略名称 最小长度:1 最大长度:95
data.showname	string	策略显示名称 最小长度:1 最大长度:95
data.description	string	应用控制策略描述 最小长度:0 最大长度:256
data.position	int32	应用控制策略位置
data.enable	bool	应用控制策略启禁用
data.updateTime	string	应用控制策略更新时间
data.durationTime	string	应用控制策略最近一次状态更新时间
data.group	string	应用控制策略组
data.labels	string[]	应用控制策略标签列表
data.src	object	应用控制策略源地址
data.src.srcZones	string[]	源区域列表
data.src.srcAddrs	object	源地址信息
data.src.srcAddrs.srcAddrType	string	地址类型 枚举值:[NETOBJECT:ip组 USERGROUP:用户及用户组 MAC:mac地址]
data.src.srcAddrs.srcMacAddrs	string[]	mac地址列表
data.src.srcAddrs.users	string[]	用户列表
data.src.srcAddrs.userGroups	string[]	用户组列表
data.src.srcAddrs.srcIpGroups	string[]	ip组列表
data.dst	object	应用控制策略目的地址
data.dst.dstZones	string[]	引用目的区域列表
data.dst.dstAddrs	object	目的地址
data.dst.dstAddrs.dstAddrType	string	地址类型 枚举值:[NETOBJECT:ip组 MAC:mac地址 DOMAIN:域名]

data.dst.dstAddrs.dstMacAddrs	string[]	mac地址列表
data.dst.dstAddrs.domains	string[]	域名列表
data.dst.dstAddrs.dstIpGroups	string[]	ip组列表
data.dst.services	string[]	服务引用列表
data.dst.applications	string[]	应用引用列表
data.action	uint32	应用控制策略行为，允许或拒绝 枚举值:[1:允许 0:拒绝]
data.schedule	string	应用控制策略起作用时间
data.advanceOption	object	应用控制策略高级选项
data.advanceOption.logEnable	bool	是否开启应用控制命中日志记录
data.advanceOption.autoSynDNS	bool	是否开启主动查询域名
data.advanceOption.keepAlive	uint8	设置长连接
data.reason	string	应用控制策略变更原因 最小长度:0 最大长度:256
data.isdefault	bool	是否是默认策略
data.lastHitTime	string	策略最近一次匹配命中时间
data.hits	uint64	应用控制策略命中匹配次数

请求成功样例：

```
{
  "data": {
    "enable": true,
    "updateTime": "2020-07-09 13:56:11",
    "labels": [],
    "description": "",
    "schedule": "全天",
    "uuid": "CB26CA52C5404C0D91ABF038D563EDBE",
    "src": {
      "srcZones": [],
      "srcAddrs": {
        "srcAddrType": "NETOBJECT",
        "srcIpGroups": [
          "全部"
        ]
      }
    },
    "durationTime": "2020-07-09 13:55:29",
    "dst": {
      "dstZones": [],
      "dstAddrs": {
        "dstAddrType": "NETOBJECT",
        "dstIpGroups": [
          "全部"
        ]
      }
    },
    "applications": [

```



```

    "全部"
  ],
  "services": [
    "any"
  ]
},
"hits": 0,
"group": "Default Policy Group",
"isdefault": false,
"action": 1,
"name": "test_1",
"__creator": "admin"
},
"code": 0,
"message": "成功"
}

```

请求失败样例:

```

{
  "data": "",
  "code": 22,
  "message": "无效参数"
}

```

4.2.1.1.18 批量修改应用控制策略

功能说明: 批量修改应用控制策略

1) API请求数据

PATCH <https://{host}/api/batch/v1/namespaces/@namespace/appcontrols/policys>

2) 输入参数

• URL参数

参数	类型	是否必选	描述
_where	string	否	指定位置, 用于移动/插入 枚举值:[top:顶部 bottom:底部 before:在目标位置之前 after:在目标位置之后]
_dest	string	否	目标位置关键字对应的值 最大长度:100
_key	string	否	目标位置关键字 枚举值:[position:指定目标位置关键字为 position name:指定目标位置关键字为 name]

• Body参数

参数	类型	是否必选	描述
objs	object[]	是	应用控制策略列表
objs.name	string	是	应用控制策略名称 最小长度:1 最大长度:95

objs.showname	string	否	策略显示名称 最小长度:1 最大长度:95
objs.description	string	否	应用控制策略描述 最小长度:0 最大长度:256
objs.position	int32	否	应用控制策略位置
objs.enable	bool	否	应用控制策略启禁用
objs.updateTime	string	否	应用控制策略更新时间
objs.durationTime	string	否	应用控制策略最近一次状态更新时间
objs.group	string	是	应用控制策略组
objs.labels	string[]	否	应用控制策略标签列表
objs.src	object	否	应用控制策略源地址
objs.src.srcZones	string[]	否	源区域列表
objs.src.srcAddrs	object	否	源地址信息
objs.src.srcAddrs.srcAddrType	string	否	地址类型 默认值:true 枚举值:[NETOBJECT:ip组 USERGROUP:用户及用户组 MAC:mac地址]
objs.src.srcAddrs.srcMacAddrs	string[]	否	mac地址列表
objs.src.srcAddrs.users	string[]	否	用户列表
objs.src.srcAddrs.userGroups	string[]	否	用户组列表
objs.src.srcAddrs.srcIpGroups	string[]	否	ip组列表
objs.dst	object	否	应用控制策略目的地址
objs.dst.dstZones	string[]	否	引用目的区域列表
objs.dst.dstAddrs	object	否	目的地址
objs.dst.dstAddrs.dstAddrType	string	否	地址类型 默认值:true 枚举值:[NETOBJECT:ip组 MAC:mac地址 DOMAIN:域名]
objs.dst.dstAddrs.dstMacAddrs	string[]	否	mac地址列表
objs.dst.dstAddrs.domains	string[]	否	域名列表
objs.dst.dstAddrs.dstIpGroups	string[]	否	ip组列表

objs.dst.services	string[]	否	服务引用列表
objs.dst.applications	string[]	否	应用引用列表
objs.action	uint32	否	应用控制策略行为，允许或拒绝 默认值:true 枚举值:[1:允许 0:拒绝]
objs.schedule	string	否	应用控制策略起作用时间
objs.advanceOption	object	否	应用控制策略高级选项
objs.advanceOption.log Enable	bool	否	是否开启应用控制命中日志记录
objs.advanceOption.autoSynDNS	bool	否	是否开启主动查询域名
objs.advanceOption.keepAlive	uint8	否	设置长连接
objs.reason	string	否	应用控制策略变更原因 最小长度:0 最大长度:256
objs.isdefault	bool	否	是否是默认策略
objs.lastHitTime	string	否	策略最近一次匹配命中时间

请求样例:

修改多条应用控制策略，修改名为test的应用控制策略动作为拒绝（0），其余不改

```
GET https://192.168.1.1/api/v1/namespaces/public/appcontrols/policys
[
  {
    "name": "test",
    "action": 0
  }
]
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	应用控制策略列表
data.uuid	string	唯一标识id
data.name	string	应用控制策略名称 最小长度:1 最大长度:95
data.showname	string	策略显示名称 最小长度:1 最大长度:95

data.description	string	应用控制策略描述 最小长度:0 最大长度:256
data.position	int32	应用控制策略位置
data.enable	bool	应用控制策略启禁用
data.updateTime	string	应用控制策略更新时间
data.durationTime	string	应用控制策略最近一次状态更新时间
data.group	string	应用控制策略组
data.labels	string[]	应用控制策略标签列表
data.src	object	应用控制策略源地址
data.src.srcZones	string[]	源区域列表
data.src.srcAddrs	object	源地址信息
data.src.srcAddrs.srcAddrType	string	地址类型 枚举值:[NETOBJECT:ip组 USERGROUP:用户及用户组 MAC:mac地址]
data.src.srcAddrs.srcMacAddrs	string[]	mac地址列表
data.src.srcAddrs.users	string[]	用户列表
data.src.srcAddrs.userGroups	string[]	用户组列表
data.src.srcAddrs.srcIpGroups	string[]	ip组列表
data.dst	object	应用控制策略目的地址
data.dst.dstZones	string[]	引用目的区域列表
data.dst.dstAddrs	object	目的地址
data.dst.dstAddrs.dstAddrType	string	地址类型 枚举值:[NETOBJECT:ip组 MAC:mac地址 DOMAIN:域名]
data.dst.dstAddrs.dstMacAddrs	string[]	mac地址列表
data.dst.dstAddrs.domains	string[]	域名列表
data.dst.dstAddrs.dstIpGroups	string[]	ip组列表
data.dst.services	string[]	服务引用列表
data.dst.applications	string[]	应用引用列表

data.action	uint32	应用控制策略行为，允许或拒绝 枚举值:[1:允许 0:拒绝]
data.schedule	string	应用控制策略起作用时间
data.advanceOption	object	应用控制策略高级选项
data.advanceOption.log Enable	bool	是否开启应用控制命中日志记录
data.advanceOption.aut oSynDNS	bool	是否开启主动查询域名
data.advanceOption.ke epAlive	uint8	设置长连接
data.reason	string	应用控制策略变更原因 最小长度:0 最大长度:256
data.isdefault	bool	是否是默认策略
data.lastHitTime	string	策略最近一次匹配命中时间
data.hits	uint64	应用控制策略命中匹配次数

请求成功样例：

```
{
  "data": [
    {
      "enable": true,
      "updateTime": "2020-07-09 14:00:35",
      "advanceOption": {
        "autoSynDNS": false,
        "logEnable": false,
        "keepAlive": 0
      },
      "labels": [],
      "description": "",
      "schedule": "全天",
      "uuid": "7CE2D60669E64F6386B57869EFC634C1",
      "src": {
        "srcZones": [],
        "srcAdrs": {
          "srcAddrType": "NETOBJECT",
          "srcIpGroups": [
            "全部"
          ]
        }
      },
      "durationTime": "2020-07-09 13:59:20",
      "dst": {
        "dstZones": [],
        "dstAdrs": {
          "dstAddrType": "NETOBJECT",
          "dstIpGroups": [
            "全部"
          ]
        }
      },
      "applications": [
        "全部"
      ],
      "services": [
        "any"
      ]
    },
    {
      "hits": 0,
      "group": "Default Policy Group",
      "isdefault": false,
    }
  ]
}
```

```
        "action": 0,
        "name": "test",
        "__creator": "admin"
    },
    "code": 0,
    "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.2.1.1.19 获取应用控制策略全局属性

功能说明：获取应用控制策略全局属性

1) API请求数据

GET

https://{host}/api/v1/namespaces/@namespace/appcontrols/globalattrs

2) 输入参数

- URL 参数

参数	类型	是否必选	描述
_select	string	否	选择字段, 字段用逗号分隔 最大长度:1000

请求样例：

获取应用控制策略全局设置

GET https://192.168.1.1/api/v1/namespaces/public/appcontrols/globalattrs

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	应用控制策略全局属性
data.healthCheckEnable	bool	开启或关闭应用控制策略健康状态自动检查
data.modifyLogEnable	bool	开启或关闭应用控制策略修改日志记录
data.conflictCheckEnable	bool	开启或关闭应用控制策略冲突检查

请求成功样例：

```
{
  "data": {
    "conflictCheckEnable": false,
    "healthCheckEnable": true,
    "modifyLogEnable": false
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.2.1.1.20 修改应用控制策略全局属性

功能说明：修改应用控制策略全局属性

1) API请求数据

PATCH

https://{host}/api/v1/namespaces/@namespace/appcontrols/globalattrs

2) 输入参数

• Body参数

参数	类型	是否必选	描述
obj	object	是	应用控制策略全局属性
obj.healthCheckEnable	bool	否	开启或关闭应用控制策略健康状态自动检查
obj.modifyLogEnable	bool	否	开启或关闭应用控制策略修改日志记录 默认值:true
obj.conflictCheckEnable	bool	否	开启或关闭应用控制策略冲突检查

请求样例：

修改应用控制策略全局设置，修改应用控制策略全局设置中的失效检查使能开关为关闭状态（ false ）， 其余不改

```
PATCH https://192.168.1.1/api/v1/namespaces/public/appcontrols/globalattrs
{
  "healthCheckEnable": false
}
```

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	应用控制策略全局属性
data.healthCheckEnable	bool	开启或关闭应用控制策略健康状态自动检查
data.modifyLogEnable	bool	开启或关闭应用控制策略修改日志记录
data.conflictCheckEnable	bool	开启或关闭应用控制策略冲突检查

请求成功样例：

```
{
  "data": {
    "conflictCheckEnable": false,
    "healthCheckEnable": false,
    "modifyLogEnable": false
  },
  "code": 0,
  "message": "成功"
}
```

```
}

请求失败样例:

{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.2.1.2 应用控制策略组

4.2.1.2.1 获取应用控制策略组

功能说明：获取应用控制策略组

1) API请求数据

GET

https://{host}/api/v1/namespaces/@namespace/appcontrols/groups

2) 输入参数

- URL 参数

参数	类型	是否必选	描述
_start	int32	否	起始位置, 从 0 开始 最小值:0 最大值:2000000
_length	int32	否	获取最大长度, 限制最大值, 如果数据不够可能返回小于 length 最小值:1 最大值:200 默认值:true
_search	string	否	模糊搜索关键字 最大长度:95
_select	string	否	选择字段, 字段用逗号分隔 最大长度:1000

请求样例:

获取多条应用控制策略组配置信息，查询条件：起始位置为0，最多展示100条

GET https://192.168.1.1/api/v1/namespaces/public/appcontrols/groups?_start=0&_length=100

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	分页数据
data.totalItems	int32	总共多少项目
data.totalPages	int32	总共多少页
data.pageNumber	int32	当前页码，从 1 开始
data.pageSize	int32	每页多大

data.itemsOffset	int32	当前条目偏移, 从 0 开始
data.itemLength	int32	数据列表长度
data.items	object[]	有效数据列表
data.items.uuid	string	唯一标识id
data.items.name	string	应用控制策略组名称 最小长度:1 最大长度:95
data.items.showname	string	策略组显示名称 最小长度:1 最大长度:95
data.items.position	int32	策略组位置
data.items.policyNum	uint32	应用控制策略组被引用数
data.items.policyStartPos	uint32	策略在策略组中的开始位置
data.items.createTime	string	应用控制策略组创建时间
data.items.isdefault	bool	是否默认策略组

请求成功样例:

```
{
  "data": {
    "itemsOffset": 0,
    "items": [
      {
        "policyNum": 0,
        "position": 0,
        "isdefault": false,
        "createTime": "2020-07-09 12:07:15",
        "uuid": "710688B93E4440D2971C902E05D02FD6",
        "name": "test"
      },
      {
        "showname": "默认策略组",
        "policyNum": 2,
        "position": 1,
        "isdefault": true,
        "uuid": "00000000010000000000000000000001",
        "name": "Default Policy Group"
      }
    ],
    "totalItems": 2,
    "pageNumber": 1,
    "itemLength": 2,
    "totalPages": 1,
    "pageSize": 100
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.2.1.2.2 批量获取应用控制策略组

功能说明：批量获取应用控制策略组

1) API请求数据

POST

https://{host}/api/batch/v1/namespaces/@namespace/appcontrols/groups?
_method=get

2) 输入参数

• URL参数

参数	类型	是否必选	描述
_select	string	否	选择字段, 字段用逗号分隔 最大长度:1000

• Body参数

参数	类型	是否必选	描述
objs	object[]	是	应用控制策略组列表
objs.name	string	是	应用控制策略组名称 最小长度:1 最大长度:95
objs.showname	string	否	策略组显示名称 最小长度:1 最大长度:95
objs.position	int32	否	策略组位置
objs.createTime	string	否	应用控制策略组创建时间
objs.isdefault	bool	否	是否默认策略组

请求样例：

获取指定的多条应用控制策略组，获取名称为test的应用控制策略组信息

```
POST https://192.168.1.1/api/batch/v1/namespaces/public/appcontrols/groups?_method=GET
[
  {
    "name": "test"
  }
]
```

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	应用控制策略组列表
data.uuid	string	唯一标识id
data.name	string	应用控制策略组名称 最小长度:1 最大长度:95

data.showname	string	策略组显示名称 最小长度:1 最大长度:95
data.position	int32	策略组位置
data.policyNum	uint32	应用控制策略组被引用数
data.policyStartPos	uint32	策略在策略组中的开始位置
data.createTime	string	应用控制策略组创建时间
data.isdefault	bool	是否默认策略组

请求成功样例：

```
{
  "data": [
    {
      "createTime": "2020-07-09 12:07:15",
      "isdefault": false,
      "name": "test",
      "uuid": "710688B93E4440D2971C902E05D02FD6",
      "position": 0
    }
  ],
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.2.1.2.3 获取指定应用控制策略组

功能说明：获取指定应用控制策略组

1) API请求数据

GET	https://{host}/api/v1/namespaces/@namespace/appcontrols/groups/@name
-----	--

2) 输入参数

- URL参数

参数	类型	是否必选	描述
_select	string	否	选择字段, 字段用逗号分隔 最大长度:1000

请求样例：

获取指定的单条应用控制策略组，获取名称为test的应用控制策略组信息

GET https://192.168.1.1/api/v1/namespaces/public/appcontrols/groups/test

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码

message	string	错误信息
data	object	应用控制策略组
data.uuid	string	唯一标识id
data.name	string	应用控制策略组名称 最小长度:1 最大长度:95
data.showname	string	策略组显示名称 最小长度:1 最大长度:95
data.position	int32	策略组位置
data.policyNum	uint32	应用控制策略组被引用数
data.policyStartPos	uint32	策略在策略组中的开始位置
data.createTime	string	应用控制策略组创建时间
data.isdefault	bool	是否默认策略组

请求成功样例:

```
{
  "data": {
    "policyNum": 0,
    "position": 0,
    "isdefault": false,
    "createTime": "2020-07-09 12:07:15",
    "uuid": "710688B93E4440D2971C902E05D02FD6",
    "name": "test"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.2.1.2.4 增加单项应用控制策略组

功能说明: 增加单项应用控制策略组

1) API请求数据

POST https://{host}/api/v1/namespaces/@namespace/appcontrols/groups

2) 输入参数

- URL 参数

参数	类型	是否必选	描述
_where	string	否	指定位置, 用于移动/插入 枚举值:[top:顶部 bottom:底部 before:在目标位置之前 after:在目标位置之后]

_dest	string	否	目标位置关键字对应的值 最大长度:100
_key	string	否	目标位置关键字 枚举值: position:指定目标位置关键字为 position name:指定目标位置关键字为 name]

- Body参数

参数	类型	是否必选	描述
obj	object	是	应用控制策略组
obj.name	string	是	应用控制策略组名称 最小长度:1 最大长度:95
obj.showname	string	否	策略组显示名称 最小长度:1 最大长度:95
obj.position	int32	否	策略组位置
obj.createTime	string	否	应用控制策略组创建时间
obj.isdefault	bool	否	是否默认策略组

请求样例:

新增单条应用控制策略组, 新增名称为test的应用控制策略组

```
POST https://192.168.1.1/api/v1/namespaces/public/appcontrols/groups
{
  "policyNum": 0,
  "position": 0,
  "isdefault": false,
  "createTime": "2020-07-09 12:07:15",
  "uuid": "710688B93E4440D2971C902E05D02FD6",
  "name": "test"
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	应用控制策略组
data.uuid	string	唯一标识id
data.name	string	应用控制策略组名称 最小长度:1 最大长度:95
data.showname	string	策略组显示名称 最小长度:1 最大长度:95
data.position	int32	策略组位置
data.policyNum	uint32	应用控制策略组被引用数

data.policyStartPos	uint32	策略在策略组中的开始位置
data.createTime	string	应用控制策略组创建时间
data.isdefault	bool	是否默认策略组

请求成功样例：

```
{
  "data": {
    "isdefault": false,
    "createTime": "2020-07-09 13:38:46",
    "uuid": "C02BD980E7524E6093CC3C3D732E8451",
    "name": "test"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.2.1.2.5 删除单项应用控制策略组

功能说明：删除单项应用控制策略组

1) API请求数据

DELETE

https://{host}/api/v1/namespaces/@namespace/appcontrols/groups/@name

2) 输入参数

- URL 参数

参数	类型	是否必选	描述
isForceDel	bool	否	是否删除策略组同时强制删除组内的所有策略

请求样例：

删除单条应用控制策略组，删除名称为test的应用控制策略组

DELETE https://192.168.1.1/api/v1/namespaces/public/appcontrols/groups/test

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	应用控制策略组
data.uuid	string	唯一标识id
data.name	string	应用控制策略组名称 最小长度:1 最大长度:95

data.showname	string	策略组显示名称 最小长度:1 最大长度:95
data.position	int32	策略组位置
data.policyNum	uint32	应用控制策略组被引用数
data.policyStartPos	uint32	策略在策略组中的开始位置
data.createTime	string	应用控制策略组创建时间
data.isdefault	bool	是否默认策略组

请求成功样例:

```
{
  "data": {
    "isdefault": false,
    "createTime": "2020-07-09 12:07:15",
    "uuid": "710688B93E4440D2971C902E05D02FD6",
    "name": "test"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.2.1.2.6 修改单项应用控制策略组

功能说明: 修改单项应用控制策略组

1) API请求数据

PATCH

https://{host}/api/v1/namespaces/@namespace/appcontrols/groups/@name

2) 输入参数

• URL 参数

参数	类型	是否必选	描述
_where	string	否	指定位置, 用于移动/插入 枚举值:[top:顶部 bottom:底部 before:在目标位置之前 after:在目标位置之后]
_dest	string	否	目标位置关键字对应的值 最大长度:100
_key	string	否	目标位置关键字 枚举值:[position:指定目标位置关键字为 position name:指定目标位置关键字为 name]

- Body参数

参数	类型	是否必选	描述
obj	object	是	应用控制策略组
obj.name	string	是	应用控制策略组名称 最小长度:1 最大长度:95
obj.showname	string	否	策略组显示名称 最小长度:1 最大长度:95
obj.position	int32	否	策略组位置
obj.createTime	string	否	应用控制策略组创建时间
obj.isdefault	bool	否	是否默认策略组

请求样例:

修改单条应用控制策略组, 修改名为test的应用控制策略组名字为test_1, 其余不改

```
PATCH https://192.168.1.1/api/v1/namespaces/public/appcontrols/groups/test
{
  "name": "test_1"
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	应用控制策略组
data.uuid	string	唯一标识id
data.name	string	应用控制策略组名称 最小长度:1 最大长度:95
data.showname	string	策略组显示名称 最小长度:1 最大长度:95
data.position	int32	策略组位置
data.policyNum	uint32	应用控制策略组被引用数
data.policyStartPos	uint32	策略在策略组中的开始位置
data.createTime	string	应用控制策略组创建时间
data.isdefault	bool	是否默认策略组

请求成功样例:

```
{
  "data": {
    "isdefault": false,
    "createTime": "2020-07-09 12:07:15",
    "uuid": "C02BD980E7524E6093CC3C3D732E8451",
    "name": "test_1"
  },
  "code": 0,
```



```
    "message": "成功"
  }
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.2.1.3 应用控制策略标签

4.2.1.3.1 获取应用控制策略标签列表

功能说明：获取应用控制策略标签列表

1) API请求数据

GET <https://{host}/api/v1/namespaces/@namespace/appcontrols/labels>

2) 输入参数

- URL参数

参数	类型	是否必选	描述
_start	int32	否	起始位置, 从 0 开始 最小值:0 最大值:2000000
_length	int32	否	获取最大长度, 限制最大值, 如果数据不够可能返回小于 length 最小值:1 最大值:200 默认值:true
_search	string	否	模糊搜索关键字 最大长度:95
_select	string	否	选择字段, 字段用逗号分隔 最大长度:1000

请求样例：

获取多条应用控制策略标签配置信息，查询条件：起始位置为0，最多展示100条

GET https://192.168.1.1/api/v1/namespaces/public/appcontrols/labels?_start=0&_length=100

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	分页数据
data.totalItems	int32	总共多少项目
data.totalPages	int32	总共多少页
data.pageNumber	int32	当前页码, 从 1 开始
data.pageSize	int32	每页多大

data.itemsOffset	int32	当前条目偏移, 从 0 开始
data.itemLength	int32	数据列表长度
data.items	object[]	有效数据列表
data.items.uuid	string	应用控制策略标签id
data.items.name	string	应用控制策略标签名称 最小长度:1 最大长度:95
data.items.showname	string	策略标签显示名称 最小长度:1 最大长度:95
data.items.policyNum	uint32	应用控制策略标签被引用数
data.items.isdefault	bool	是否是默认标签

请求成功样例:

```
{
  "data": {
    "itemsOffset": 0,
    "items": [
      {
        "isdefault": false,
        "policyNum": 0,
        "uuid": "39D1D7AA03CB410AA82A08752F63A058",
        "name": "test"
      },
      {
        "showname": "默认",
        "policyNum": 1,
        "isdefault": true,
        "uuid": "00000000010000000000000000000002",
        "name": "Default"
      }
    ],
    "totalItems": 2,
    "pageNumber": 1,
    "itemLength": 2,
    "totalPages": 1,
    "pageSize": 100
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.2.1.3.2 获取指定应用控制策略标签

功能说明: 获取指定应用控制策略标签

1) API请求数据

GET

https://{host}/api/v1/namespaces/@namespace/appcontrols/labels/@name

2) 输入参数

- URL 参数

参数	类型	是否必选	描述
----	----	------	----

_select	string	否	选择字段, 字段用逗号分隔 最大长度:1000
---------	--------	---	----------------------------

请求样例:

获取单条应用控制策略标签, 获取名称为test的应用控制策略标签信息

GET https://192.168.1.1/api/v1/namespaces/public/appcontrols/labels/test

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	应用控制策略标签
data.uuid	string	应用控制策略标签id
data.name	string	应用控制策略标签名称 最小长度:1 最大长度:95
data.showname	string	策略标签显示名称 最小长度:1 最大长度:95
data.policyNum	uint32	应用控制策略标签被引用数
data.isdefault	bool	是否是默认标签

请求成功样例:

```
{
  "data": {
    "isdefault": false,
    "policyNum": 0,
    "uuid": "39D1D7AA03CB410AA82A08752F63A058",
    "name": "test"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.2.1.3.3 增加单项应用控制策略标签

功能说明: 增加单项应用控制策略标签

1) API请求数据

POST	https://{host}/api/v1/namespaces/@namespace/appcontrols/labels
------	--

2) 输入参数

• URL参数

参数	类型	是否必选	描述
_where	string	否	指定位置, 用于移动/插入 枚举值:[top:顶部 bottom:底部 before:在目标位置之前 after:在目标位置之后]
_dest	string	否	目标位置关键字对应的值 最大长度:100
_key	string	否	目标位置关键字 枚举值:[position:指定目标位置关键字为 position name:指定目标位置关键字为 name]

• Body参数

参数	类型	是否必选	描述
obj	object	是	应用控制策略标签
obj.name	string	是	应用控制策略标签名称 最小长度:1 最大长度:95
obj.showname	string	否	策略标签显示名称 最小长度:1 最大长度:95
obj.isdefault	bool	否	是否是默认标签

请求样例:

新增单条应用控制策略标签, 新增名称为test的应用控制策略标签

```
POST https://192.168.1.1/api/v1/namespaces/public/appcontrols/labels
{
  "name": "test"
}
```

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	应用控制策略标签
data.uuid	string	应用控制策略标签id
data.name	string	应用控制策略标签名称 最小长度:1 最大长度:95

data.showname	string	策略标签显示名称 最小长度:1 最大长度:95
data.policyNum	uint32	应用控制策略标签被引用数
data.isdefault	bool	是否是默认标签

请求成功样例:

```
{
  "data": {
    "isdefault": false,
    "uuid": "39D1D7AA03CB410AA82A08752F63A058",
    "name": "test"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.2.1.3.4 删除单项应用控制策略标签

功能说明: 删除单项应用控制策略标签

1) API请求数据

DELETE https://{host}/api/v1/namespaces/@namespace/appcontrols/labels/@name

2) 输入参数

请求样例:

删除单条应用控制策略标签，删除名称为test的应用控制策略标签
DELETE https://192.168.1.1/api/v1/namespaces/public/appcontrols/labels/test

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	应用控制策略标签
data.uuid	string	应用控制策略标签id
data.name	string	应用控制策略标签名称 最小长度:1 最大长度:95
data.showname	string	策略标签显示名称 最小长度:1 最大长度:95
data.policyNum	uint32	应用控制策略标签被引用数
data.isdefault	bool	是否是默认标签

请求成功样例:

```
{
  "data": {
    "isdefault": false,
    "uuid": "39D1D7AA03CB410AA82A08752F63A058",
    "name": "test"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.2.1.3.5 批量添加应用控制策略标签

功能说明: 批量添加应用控制策略标签

1) API请求数据

POST <https://{host}/api/batch/v1/namespaces/@namespace/appcontrols/labels>

2) 输入参数

• URL 参数

参数	类型	是否必选	描述
_where	string	否	指定位置, 用于移动/插入 枚举值:[top:顶部 bottom:底部 before:在目标位置之前 after:在目标位置之后]
_dest	string	否	目标位置关键字对应的值 最大长度:100
_key	string	否	目标位置关键字 枚举值:[position:指定目标位置关键字为 position name:指定目标位置关键字为 name]

• Body参数

参数	类型	是否必选	描述
objs	object[]	是	应用控制策略标签列表
objs.name	string	是	应用控制策略标签名称 最小长度:1 最大长度:95
objs.showname	string	否	策略标签显示名称 最小长度:1 最大长度:95
objs.isdefault	bool	否	是否是默认标签

请求样例:

```
新增多条应用控制策略标签，新增名称为test的应用控制策略标签

POST https://192.168.1.1/api/batch/v1/namespaces/public/appcontrols/labels
[
  {
    "name": "test"
  }
]
```

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	应用控制策略标签列表
data.uuid	string	应用控制策略标签id
data.name	string	应用控制策略标签名称 最小长度:1 最大长度:95
data.showname	string	策略标签显示名称 最小长度:1 最大长度:95
data.policyNum	uint32	应用控制策略标签被引用数
data.isdefault	bool	是否是默认标签

请求成功样例:

```
{
  "data": [
    {
      "isdefault": false,
      "uuid": "46F348DCB2534413BA14C2ED2B773ADE",
      "name": "test"
    }
  ],
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.2.1.3.6 批量删除应用控制策略标签

功能说明: 批量删除应用控制策略标签

1) API请求数据

POST	https://{host}/api/batch/v1/namespaces/@namespace/appcontrols/labels?_method=delete
------	---

2) 输入参数

• Body参数

参数	类型	是否必填	描述
objs	object[]	是	应用控制策略标签列表
objs.name	string	是	应用控制策略标签名称 最小长度:1 最大长度:95
objs.showname	string	否	策略标签显示名称 最小长度:1 最大长度:95
objs.isdefault	bool	否	是否是默认标签

请求样例:

删除多条应用控制策略标签, 删除名称为test的应用控制策略标签

```
POST https://192.168.1.1/api/batch/v1/namespaces/public/appcontrols/labels?_method=DELETE
[
  {
    "name": "test"
  }
]
```

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	应用控制策略标签列表
data.uuid	string	应用控制策略标签id
data.name	string	应用控制策略标签名称 最小长度:1 最大长度:95
data.showname	string	策略标签显示名称 最小长度:1 最大长度:95
data.policyNum	uint32	应用控制策略标签被引用数
data.isdefault	bool	是否是默认标签

请求成功样例:

```
{
  "data": [
    {
      "isdefault": false,
      "uuid": "46F348DCB2534413BA14C2ED2B773ADE",
      "name": "test"
    }
  ],
  "code": 0,
  "message": "成功"
}
```

请求失败样例:


```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.2.1.3.7 修改单项应用控制策略标签

功能说明：修改单项应用控制策略标签

1) API请求数据

PATCH https://{host}/api/v1/namespaces/@namespace/appcontrols/labels/@name

2) 输入参数

• URL参数

参数	类型	是否必选	描述
_where	string	否	指定位置, 用于移动/插入 枚举值:[top:顶部 bottom:底部 before:在目标位置之前 after:在目标位置之后]
_dest	string	否	目标位置关键字对应的值 最大长度:100
_key	string	否	目标位置关键字 枚举值:[position:指定目标位置关键字为 position name:指定目标位置关键字为 name]

• Body参数

参数	类型	是否必选	描述
obj	object	是	应用控制策略标签
obj.name	string	是	应用控制策略标签名称 最小长度:1 最大长度:95
obj.showname	string	否	策略标签显示名称 最小长度:1 最大长度:95
obj.isdefault	bool	否	是否是默认标签

请求样例：

修改单条应用控制策略标签，修改名为test的应用控制策略标签名字为test_1，其余不改

```
PATCH https://192.168.1.1/api/v1/namespaces/public/appcontrols/labels/test
{
  "name": "test_1"
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	应用控制策略标签
data.uuid	string	应用控制策略标签id
data.name	string	应用控制策略标签名称 最小长度:1 最大长度:95
data.showname	string	策略标签显示名称 最小长度:1 最大长度:95
data.policyNum	uint32	应用控制策略标签被引用数
data.isdefault	bool	是否是默认标签

请求成功样例:

```
{
  "data": {
    "isdefault": false,
    "uuid": "39D1D7AA03CB410AA82A08752F63A058",
    "name": "test_1"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.2.1.4 策略生命周期管理

4.2.1.4.1 获取生命周期日志列表

功能说明: 获取生命周期日志列表

1) API请求数据

GET

https://{host}/api/v1/namespaces/@namespace/appcontrols/lifecycles/list

2) 输入参数

- URL参数

参数	类型	是否必选	描述
_start	int32	否	起始位置, 从 0 开始 最小值:0 最大值:2000000

modifyType	string	否	策略修改类型，包括新增，删除及修改 枚举值:[add:新增策略的修改类型 del:删除策略的修改类型 modify:修改策略的修改类型 all:所有修改类型]
endDate	string	否	终止日期
policyName	string	否	策略名称 最小长度:1 最大长度:128
_search	string	否	模糊搜索关键字 最大长度:95
showPolicyDetail	bool	否	显示策略详细信息
_length	int32	否	获取最大长度，限制最大值，如果数据不够 可能返回小于 length 最小值:1 最大值:200 默认值:true
policyUuid	string	否	策略名称或all 最小长度:1 最大长度:128
startDate	string	否	开始日期
modifyUserName	string	否	修改策略的用户名称 最小长度:1 最大长度:60

请求样例：

获取多条应用控制策略生命周期日志信息，查询条件：起始位置为0，最多展示100条

GET https://192.168.1.1/api/v1/namespaces/public/appcontrols/lifecycles/list?_start=0&_length=100

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	分页数据
data.totalItems	int32	总共多少项目
data.totalPages	int32	总共多少页
data.pageNumber	int32	当前页码，从 1 开始
data.pageSize	int32	每页多大
data.itemsOffset	int32	当前条目偏移, 从 0 开始
data.itemLength	int32	数据列表长度
data.items	object[]	有效数据列表

data.items.id	string	记录id
data.items.time	string	修改策略时间
data.items.policyName	string	应用控制策略名称 最小长度:1 最大长度:95
data.items.modifyType	string	修改类型 枚举值:[add:新增策略的修改类型 del:删除策略的修改类型 modify:修改策略的修改类型 none:无修改类型]
data.items.UserName	string	修改策略的用户名称 最大长度:95
data.items.modifyIp	string	修改策略的用户ip 最小长度:0 最大长度:48
data.items.modifyReasons	string	修改策略的原因 最大长度:256
data.items.modifySummary	string	修改策略的内容概要
data.items.policyDetail	object	修改策略的详细信息
data.items.policyDetail.name	string	应用控制策略名称 最小长度:1 最大长度:95
data.items.policyDetail.group	string	策略组名 最小长度:1 最大长度:95
data.items.policyDetail.description	string	描述信息 最小长度:0 最大长度:95
data.items.policyDetail.priority	string	策略优先级id 最小长度:1 最大长度:95
data.items.policyDetail.label	string	策略标签
data.items.policyDetail.srcZoneInfo	string	源区域信息
data.items.policyDetail.srcAddrInfo	string	源地址信息
data.items.policyDetail.dstZoneInfo	string	目的区域信息
data.items.policyDetail.dstAddrInfo	string	目的地址信息
data.items.policyDetail.servInfo	string	服务信息

data.items.policyDetail.applInfo	string	应用信息
data.items.policyDetail.s tatus	string	状态信息
data.items.policyDetail. action	string	动作信息
data.items.policyDetail.s chedule	string	生效时间信息
data.items.policyDetail.k eepAlive	string	长连接信息
data.items.policyDetail.l ogEnable	string	日志记录开关
data.items.policyDetail. autoSynDNS	string	自动查询dns

请求成功样例：

```
{
  "data": {
    "itemsOffset": 0,
    "items": [
      {
        "policyName": "test",
        "modifySummary": "新增策略",
        "modifyType": "add",
        "id": "202007091541",
        "modifyReasons": "",
        "time": "2020-07-09 15:30:40",
        "modifyIp": "200.200.152.213",
        "UserName": "admin"
      }
    ],
    "totalItems": 1,
    "pageNumber": 1,
    "itemLength": 1,
    "totalPages": 1,
    "pageSize": 100
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.2.1.4.2 获取生命周期管理日志对比详情

功能说明：获取生命周期管理日志对比详情

1) API请求数据

GET	https://{host}/api/v1/namespaces/@namespace/appcontrols/lifecycles/detailist
-----	--

2) 输入参数

- URL参数

参数	类型	是否必选	描述
recordId	string	否	记录id 最小长度:1 最大长度:64

请求样例:

获取单条应用控制策略生命周期日志对比信息，获取应用控制策略生命周期日志记录id为20200709|15|4|1与前一次修改的对比信息

GET https://192.168.1.1/api/v1/namespaces/public/appcontrols/lifecycles/detailist?recordId=20200709|15|4|1

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	策略修改详情列表
data.name	string	策略中修改的字段名称 最大长度:95
data.state	string	策略修改类型包括增删改及无变化 枚举值: add:新增策略的修改类型 del:删除策略的修改类型 modify:修改策略的修改类型 none:无修改类型]
data.isExpired	bool	日志是否过期
data.before	object[]	修改前内容
data.before.text	string	字段内容
data.before.state	string	修改类型 枚举值: add:新增策略的修改类型 del:删除策略的修改类型 modify:修改策略的修改类型 none:无修改类型]
data.after	object[]	修改后内容
data.after.text	string	字段内容
data.after.state	string	修改类型 枚举值: add:新增策略的修改类型 del:删除策略的修改类型 modify:修改策略的修改类型 none:无修改类型]

请求成功样例:

```
{
  "data": [
    {
      "after": [
        {
          "state": "none",
          "text": "test"
        }
      ],
      "before": [],
      "state": "none",
      "name": "name",
      "isExpired": false
    },
    {
      "after": [
        {
          "state": "none",
          "text": ""
        }
      ],
      "before": [],
      "state": "none",
      "name": "description",
      "isExpired": false
    },
    {
      "after": [
        {
          "state": "none",
          "text": "默认策略组"
        }
      ],
      "before": [],
      "state": "none",
      "name": "group",
      "isExpired": false
    },
    {
      "after": [
        {
          "state": "none",
          "text": "1"
        }
      ],
      "before": [],
      "state": "none",
      "name": "defaultId",
      "isExpired": false
    },
    {
      "after": [],
      "before": [],
      "state": "none",
      "name": "label",
      "isExpired": false
    },
    {
      "after": [
        {
          "state": "none",
          "text": "any"
        }
      ],
      "before": [],
      "state": "none",
      "name": "sourceZoneS",
      "isExpired": false
    },
    {
      "after": [
        {
          "state": "none",
          "text": "全部"
        }
      ]
    }
  ]
}
```

```

    ],
    "before": [],
    "state": "none",
    "name": "ipUsrGroups",
    "isExpired": false
  },
  {
    "after": [
      {
        "state": "none",
        "text": "any"
      }
    ],
    "before": [],
    "state": "none",
    "name": "destZoneS",
    "isExpired": false
  },
  {
    "after": [
      {
        "state": "none",
        "text": "全部"
      }
    ],
    "before": [],
    "state": "none",
    "name": "ipGrps",
    "isExpired": false
  },
  {
    "after": [
      {
        "state": "none",
        "text": "any(TCP:srcPorts[0-65535],dstPorts[0-65535];UDP:srcPorts[0-65535],dstPorts[0-65535];ICMP:type 0-255, code 0-255;ICMPv6:type 0-255, code 0-255;协议号:2-5,7-16,18-57,59-255;)"
      }
    ],
    "before": [],
    "state": "none",
    "name": "services",
    "isExpired": false
  },
  {
    "after": [
      {
        "state": "none",
        "text": "全部/全部"
      }
    ],
    "before": [],
    "state": "none",
    "name": "applications",
    "isExpired": false
  },
  {
    "after": [
      {
        "state": "none",
        "text": "全天"
      }
    ],
    "before": [],
    "state": "none",
    "name": "time",
    "isExpired": false
  },
  {
    "after": [
      {
        "state": "none",
        "text": "允许"
      }
    ],
    "before": [],
    "state": "none",

```



```
    "name": "action",
    "isExpired": false
  },
  {
    "after": [
      {
        "state": "none",
        "text": "不记录日志"
      }
    ],
    "before": [],
    "state": "none",
    "name": "log",
    "isExpired": false
  },
  {
    "after": [
      {
        "state": "none",
        "text": "关闭"
      }
    ],
    "before": [],
    "state": "none",
    "name": "longHref",
    "isExpired": false
  },
  {
    "after": [
      {
        "state": "none",
        "text": "启用"
      }
    ],
    "before": [],
    "state": "none",
    "name": "status",
    "isExpired": false
  },
  {
    "after": [
      {
        "state": "none",
        "text": "被动"
      }
    ],
    "before": [],
    "state": "none",
    "name": "autoSyn",
    "isExpired": false
  }
],
"code": 0,
"message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.2.1.4.3 获取策略修改日志策略名或用户名列表

功能说明: 获取策略修改日志策略名或用户名列表

1) API请求数据

GET

https://{host}/api/v1/namespaces/@namespace/appcontrols/lifecycles/name
list

2) 输入参数

• URL参数

参数	类型	是否必选	描述
_start	int32	否	起始位置, 从 0 开始 最小值:0 最大值:2000000
_length	int32	否	获取最大长度, 限制最大值, 如果数据不够可能返回小于 length 最小值:1 最大值:200 默认值:true
colName	string	否	用户名或策略名 枚举值:[POLICY_NAME:获取策略名 USER_NAME:获取修改用户]

请求样例:

获取多条应用控制策略生命周期日志策略名称信息, 查询条件: 起始位置为0, 最多展示100条

GET https://192.168.1.1/api/v1/namespaces/public/appcontrols/lifecycles/namelist?_start=0&_length=100

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	分页数据
data.totalItems	int32	总共多少项目
data.totalPages	int32	总共多少页
data.pageNumber	int32	当前页码, 从 1 开始
data.pageSize	int32	每页多大
data.itemsOffset	int32	当前条目偏移, 从 0 开始
data.itemLength	int32	数据列表长度
data.items	object[]	有效数据列表
data.items.name	string	用户名或策略名 最小长度:1 最大长度:95
data.items.value	string	用户名或策略uuid 最大长度:256

请求成功样例:

```
{
  "data": {
    "itemsOffset": 0,
    "items": [
```

```
{
  {
    "name": "全部",
    "value": "all"
  },
  {
    "name": "test",
    "value": "D6A7C95D58ED40BD91A8EA60227A8B6A"
  }
],
"totalItems": 2,
"pageNumber": 1,
"itemLength": 2,
"totalPages": 1,
"pageSize": 100
},
"code": 0,
"message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.2.1.4.4 导出策略修改日志

功能说明: 导出策略修改日志

1) API请求数据

GET

https://{host}/api/v1/namespaces/@namespace/appcontrols/lifecycles/export

2) 输入参数

- URL参数

参数	类型	是否必选	描述
policyName	string	否	策略名称 最小长度:1 最大长度:128
modifyType	string	否	策略修改类型, 包括新增, 删除及修改 枚举值:[add:新增策略的修改类型 del:删除策略的修改类型 modify:修改策略的修改类型 all:所有修改类型]
modifyUserName	string	否	修改策略的用户名称 最小长度:1 最大长度:60
policyUuid	string	否	策略名称或all 最小长度:1 最大长度:128
startDate	string	否	开始日期
endDate	string	否	结止日期

请求样例:

导出应用控制策略生命周期日志

GET https://192.168.1.1/api/v1/namespaces/public/appcontrols/lifecycles/export

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	string	

请求成功样例：

```
{
  "data": "/export/appconLifeCycle.csv",
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.2.1.4.5 修改策略修改日志导出列表

功能说明：修改策略修改日志导出列表

1) API请求数据

POST	https://{host}/api/v1/namespaces/@namespace/appcontrols/lifecycles/exportfield
------	--

2) 输入参数

- Body参数

参数	类型	是否必选	描述
obj	string[]	是	策略导出字段列表

请求样例：

更新应用控制策略生命周期日志导出内容，设置应用控制策略生命周期日志导出内容为名称和策略组

```
POST https://192.168.1.1/api/v1/namespaces/public/appcontrols/lifecycles/exportfield
[
  "NAME",
  "GROUP"
]
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	bool	

请求成功样例：

```
{
  "data": true,
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.2.1.4.6 获取策略修改日志导出列表

功能说明：获取策略修改日志导出列表

1) API请求数据

GET	https://{host}/api/v1/namespaces/@namespace/appcontrols/lifecycles/exportfield
-----	--

2) 输入参数

请求样例：

获取应用控制策略生命周期日志导出内容，设置应用控制策略生命周期日志导出内容为名称和策略组

GET https://192.168.1.1/api/v1/namespaces/public/appcontrols/lifecycles/exportfield

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	string[]	策略导出字段列表

请求成功样例：

```
{
  "data": [
    "NAME",
    "GROUP"
  ],
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.2.2 地域访问控制

4.2.2.1 地域访问控制策略

4.2.2.1.1 获取地域访问控制策略

功能说明：通过此接口，获得符合条件的全部地域访问控制策略

1) API请求数据

GET

https://{host}/api/v1/namespaces/@namespace/areactlpolicys

2) 输入参数

- URL 参数

参数	类型	是否必选	描述
_start	int32	否	起始位置, 从 0 开始 最小值:0 最大值:2000000
_search	string	否	模糊搜索关键字 最大长度:95
_length	int32	否	获取最大长度, 限制最大值, 如果数据不够可能返回小于 length 最小值:1 最大值:200 默认值:true

请求样例：

获取地域访问控制策略

GET https://192.168.1.1/api/v1/namespaces/public/areactlpolicys?_search=ctl_2

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	分页数据
data.totalItems	int32	总共多少项目
data.totalPages	int32	总共多少页
data.pageNumber	int32	当前页码, 从 1 开始
data.pageSize	int32	每页多大
data.itemsOffset	int32	当前条目偏移, 从 0 开始
data.itemLength	int32	数据列表长度
data.items	object[]	有效数据列表
data.items.uuid	string	唯一标识id
data.items.name	string	名称 最小长度:1 最大长度:95

data.items.description	string	描述 最小长度:0 最大长度:95
data.items.enable	bool	是否启用
data.items.action	string	控制方式 枚举值:[ALLOW:只允许以下国家/地区访问 DENY:拒绝以下国家/地区访问]
data.items.srcZones	string[]	源外网区域
data.items.dstIPGroups	string[]	目的网络对象
data.items.boardAndCountry	string[]	国家/地区
data.items.position	uint32	策略位置

请求成功样例:

```
{
  "data": {
    "itemsOffset": 0,
    "items": [
      {
        "enable": true,
        "position": 2,
        "dstIPGroups": [
          "ipv4_1",
          "ipv4_2",
          "ipv4_3"
        ],
        "srcZones": [
          "zoneL2_1",
          "zoneL3_1",
          "zoneVirtual_1"
        ],
        "boardAndCountry": [
          "非洲地区/贝宁"
        ],
        "uuid": "BD9C3643B36E4EAFB354848260142C1A",
        "action": "ALLOW",
        "name": "areactl_2",
        "description": ""
      }
    ],
    "totalItems": 1,
    "pageNumber": 1,
    "itemLength": 1,
    "totalPages": 1,
    "pageSize": 100
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.2.2.1.2 获取单条地域访问控制策略

功能说明：通过此接口，根据名称获取单条地域访问控制策略

1) API请求数据

GET

https://{host}/api/v1/namespaces/@namespace/areactlpolicys/@name

2) 输入参数

请求样例：

获取单条地域访问控制策略

GET https://192.168.1.1/api/v1/namespaces/public/areactlpolicys/areactl_2

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	地域访问控制策略
data.uuid	string	唯一标识id
data.name	string	名称 最小长度:1 最大长度:95
data.description	string	描述 最小长度:0 最大长度:95
data.enable	bool	是否启用
data.action	string	控制方式 枚举值: ALLOW:只允许以下国家/地区访问 DENY:拒绝以下国家/地区访问]
data.srcZones	string[]	源外网区域
data.dstIPGroups	string[]	目的网络对象
data.boardAndCountry	string[]	国家/地区
data.position	uint32	策略位置

请求成功样例：

```
{
  "data": {
    "enable": true,
    "position": 2,
    "dstIPGroups": [
      "ipv4_1",
      "ipv4_2",
      "ipv4_3"
    ],
    "srcZones": [
      "zoneL2_1",
      "zoneL3_1",
      "zoneVirtual_1"
    ]
  }
}
```



```
"boardAndCountry": [
  "非洲地区/贝宁"
],
"uuid": "BD9C3643B36E4EAFB354848260142C1A",
"action": "ALLOW",
"name": "areactl_2",
"description": "",
},
"code": 0,
"message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.2.2.1.3 新增地域访问控制策略

功能说明: 通过此接口, 添加一条地域访问控制策略

1) API请求数据

POST https://{host}/api/v1/namespaces/@namespace/areactlpolicys

2) 输入参数

• Body参数

参数	类型	是否必填	描述
policy	object	是	地域访问控制策略
policy.name	string	是	名称 最小长度:1 最大长度:95
policy.description	string	否	描述 最小长度:0 最大长度:95
policy.enable	bool	否	是否启用 默认值:true
policy.action	string	否	控制方式 默认值:true 枚举值: ALLOW:只允许以下国家/地区访问 DENY:拒绝以下国家/地区访问]
policy.srcZones	string[]	否	源外网区域
policy.dstIPGroups	string[]	否	目的网络对象
policy.boardAndCountry	string[]	否	国家/地区
policy.position	uint32	否	策略位置

请求样例:

新增地域访问控制策略

```
POST https://192.168.1.1/api/v1/namespaces/public/areactlpolicys
{
  "enable": true,
  "boardAndCountry": [
```

```

    "非洲地区/贝宁"
  ],
  "dstIPGroups": [
    "ipv4_1",
    "ipv4_2",
    "ipv4_3"
  ],
  "name": "areactl_2",
  "action": "ALLOW",
  "srcZones": [
    "zoneL2_1",
    "zoneL3_1",
    "zoneVirtual_1"
  ],
  "description": ""
}

```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	地域访问控制策略
data.uuid	string	唯一标识id
data.name	string	名称 最小长度:1 最大长度:95
data.description	string	描述 最小长度:0 最大长度:95
data.enable	bool	是否启用
data.action	string	控制方式 枚举值: ALLOW:只允许以下国家/地区访问 DENY:拒绝以下国家/地区访问]
data.srcZones	string[]	源外网区域
data.dstIPGroups	string[]	目的网络对象
data.boardAndCountry	string[]	国家/地区
data.position	uint32	策略位置

请求成功样例:

```

{
  "data": {
    "enable": true,
    "boardAndCountry": [
      "非洲地区/贝宁"
    ],
    "srcZones": [
      "zoneL2_1",
      "zoneL3_1",
      "zoneVirtual_1"
    ],
    "dstIPGroups": [
      "ipv4_1",
      "ipv4_2",
      "ipv4_3"
    ]
  }
}

```

```
    },
    "uuid": "BD9C3643B36E4EAFB354848260142C1A",
    "action": "ALLOW",
    "name": "areactl_2",
    "description": ""
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.2.2.1.4 修改地域访问控制策略

功能说明: 通过此接口, 根据名称修改对应的地域访问控制策略

1) API请求数据

PATCH <https://{host}/api/v1/namespaces/@namespace/areactlpolicys/@name>

2) 输入参数

• URL参数

参数	类型	是否必选	描述
_arrayop	string	否	对象属性是数组的操作控制 枚举值:[add:添加对应的数组元素 remove:删除对应的数组元素]
_key	string	否	目标位置关键字 枚举值:[position:指定目标位置关键字为 position name:指定目标位置关键字为 name]
_dest	string	否	目标位置关键字对应的值 最大长度:100
_where	string	否	指定位置, 用于移动/插入 枚举值:[top:顶部 bottom:底部 before:在目标位置之前 after:在目标位置之后]

• Body参数

参数	类型	是否必选	描述
policy	object	是	地域访问控制策略
policy.name	string	是	名称 最小长度:1 最大长度:95

policy.description	string	否	描述 最小长度:0 最大长度:95
policy.enable	bool	否	是否启用 默认值:true
policy.action	string	否	控制方式 默认值:true 枚举值:[ALLOW:只允许以下国家/地区访问 DENY:拒绝以下国家/地区访问]
policy.srcZones	string[]	否	源外网区域
policy.dstIPGroups	string[]	否	目的网络对象
policy.boardAndCountry	string[]	否	国家/地区
policy.position	uint32	否	策略位置

请求样例:

修改地域访问控制策略

```
PATCH https://192.168.1.1/api/v1/namespaces/public/areactlpolicys/areactl_2
{
  "enable": false,
  "description": "a test"
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	地域访问控制策略
data.uuid	string	唯一标识id
data.name	string	名称 最小长度:1 最大长度:95
data.description	string	描述 最小长度:0 最大长度:95
data.enable	bool	是否启用
data.action	string	控制方式 枚举值:[ALLOW:只允许以下国家/地区访问 DENY:拒绝以下国家/地区访问]
data.srcZones	string[]	源外网区域
data.dstIPGroups	string[]	目的网络对象
data.boardAndCountry	string[]	国家/地区

data.position	uint32	策略位置
---------------	--------	------

请求成功样例：

```
{
  "data": {
    "enable": false,
    "boardAndCountry": [
      "非洲地区/贝宁"
    ],
    "srcZones": [
      "zoneL2_1",
      "zoneL3_1",
      "zoneVirtual_1"
    ],
    "dstIPGroups": [
      "ipv4_1",
      "ipv4_2",
      "ipv4_3"
    ],
    "uuid": "BD9C3643B36E4EAFB354848260142C1A",
    "action": "ALLOW",
    "name": "areactl_2",
    "description": "a test"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.2.2.1.5 全量修改地域访问控制策略

功能说明：通过此接口，根据名称修改对应的地域访问控制策略

1) API请求数据

PUT

https://{host}/api/v1/namespaces/@namespace/areactlpolicys/@name

2) 输入参数

- Body参数

参数	类型	是否必选	描述
policy	object	是	地域访问控制策略
policy.name	string	是	名称 最小长度:1 最大长度:95
policy.description	string	否	描述 最小长度:0 最大长度:95
policy.enable	bool	否	是否启用 默认值:true

policy.action	string	否	控制方式 默认值:true 枚举值:[ALLOW:只允许以下国家/地区访问 DENY:拒绝以下国家/地区访问]
policy.srcZones	string[]	否	源外网区域
policy.dstIPGroups	string[]	否	目的网络对象
policy.boardAndCountry	string[]	否	国家/地区
policy.position	uint32	否	策略位置

请求样例:

全量修改地域访问控制策略

```
PUT https://192.168.1.1/api/v1/namespaces/public/areactlpolicys/areactl_2
{
  "enable": true,
  "boardAndCountry": [
    "非洲地区/安哥拉"
  ],
  "dstIPGroups": [
    "ipv4_1"
  ],
  "name": "areactl_2",
  "action": "DENY",
  "srcZones": [
    "zoneL2_1"
  ],
  "description": ""
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	地域访问控制策略
data.uuid	string	唯一标识id
data.name	string	名称 最小长度:1 最大长度:95
data.description	string	描述 最小长度:0 最大长度:95
data.enable	bool	是否启用
data.action	string	控制方式 枚举值:[ALLOW:只允许以下国家/地区访问 DENY:拒绝以下国家/地区访问]
data.srcZones	string[]	源外网区域
data.dstIPGroups	string[]	目的网络对象

data.boardAndCountry	string[]	国家/地区
data.position	uint32	策略位置

请求成功样例：

```
{
  "data": {
    "enable": true,
    "boardAndCountry": [
      "非洲地区/安哥拉"
    ],
    "srcZones": [
      "zoneL2_1"
    ],
    "dstIPGroups": [
      "ipv4_1"
    ],
    "uuid": "BD9C3643B36E4EAFB354848260142C1A",
    "action": "DENY",
    "name": "areactl_2",
    "description": ""
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.2.2.1.6 删除地域访问控制策略

功能说明：通过此接口，根据名称删除对应的地域访问控制策略

1) API请求数据

DELETE

https://{host}/api/v1/namespaces/@namespace/areactlpolicys/@name

2) 输入参数

请求样例：

删除地域访问控制策略

DELETE https://192.168.1.1/api/v1/namespaces/public/areactlpolicys/areactl_2

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	地域访问控制策略
data.uuid	string	唯一标识id
data.name	string	名称 最小长度:1 最大长度:95

data.description	string	描述 最小长度:0 最大长度:95
data.enable	bool	是否启用
data.action	string	控制方式 枚举值:[ALLOW:只允许以下国家/地区访问 DENY:拒绝以下国家/地区访问]
data.srcZones	string[]	源外网区域
data.dstIPGroups	string[]	目的网络对象
data.boardAndCountry	string[]	国家/地区
data.position	uint32	策略位置

请求成功样例:

```
{
  "data": {
    "enable": true,
    "boardAndCountry": [
      "非洲地区/安哥拉"
    ],
    "srcZones": [
      "zoneL2_1"
    ],
    "dstIPGroups": [
      "ipv4_1"
    ],
    "uuid": "BD9C3643B36E4EAFB354848260142C1A",
    "action": "DENY",
    "name": "areactl_2",
    "description": ""
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.2.2.1.7 批量删除地域访问控制策略

功能说明: 通过此接口, 根据名称批量删除对应的地域访问控制策略

1) API请求数据

POST

https://{host}/api/batch/v1/namespaces/@namespace/areactlpolicys?_method=delete

2) 输入参数

- Body参数

参数	类型	是否必选	描述
policys	object[]	是	地域访问控制策略

polycys.name	string	是	名称 最小长度:1 最大长度:95
polycys.description	string	否	描述 最小长度:0 最大长度:95
polycys.enable	bool	否	是否启用 默认值:true
polycys.action	string	否	控制方式 默认值:true 枚举值:[ALLOW:只允许以下国家/地区访问 DENY:拒绝以下国家/地区访问]
polycys.srcZones	string[]	否	源外网区域
polycys.dstIPGroups	string[]	否	目的网络对象
polycys.boardAndCountry	string[]	否	国家/地区
polycys.position	uint32	否	策略位置

请求样例:

批量删除地域访问控制策略

POST https://192.168.1.1/api/v1/namespaces/public/areactlpolycys?_method=DELETE

```
[
  {
    "name": "areactl_1"
  },
  {
    "name": "areactl_2"
  },
  {
    "name": "areactl_3"
  }
]
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	地域访问控制策略
data.uuid	string	唯一标识id
data.name	string	名称 最小长度:1 最大长度:95
data.description	string	描述 最小长度:0 最大长度:95
data.enable	bool	是否启用

data.action	string	控制方式 枚举值:[ALLOW:只允许以下国家/地区访问 DENY:拒绝以下国家/地区访问]
data.srcZones	string[]	源外网区域
data.dstIPGroups	string[]	目的网络对象
data.boardAndCountry	string[]	国家/地区
data.position	uint32	策略位置

请求成功样例:

```
{
  "data": [
    {
      "enable": true,
      "boardAndCountry": [
        "非洲地区/贝宁"
      ],
      "srcZones": [
        "zoneL2_1",
        "zoneL3_1",
        "zoneVirtual_1"
      ],
      "dstIPGroups": [
        "ipv4_1",
        "ipv4_2",
        "ipv4_3"
      ],
      "uuid": "002F928CDDBC4EBDBBB8EE9E260EB9BA",
      "action": "ALLOW",
      "name": "areactl_1",
      "description": ""
    },
    {
      "enable": true,
      "boardAndCountry": [
        "非洲地区/贝宁"
      ],
      "srcZones": [
        "zoneL2_1",
        "zoneL3_1",
        "zoneVirtual_1"
      ],
      "dstIPGroups": [
        "ipv4_1",
        "ipv4_2",
        "ipv4_3"
      ],
      "uuid": "943365D54B434E5692AB892F10B52EBB",
      "action": "ALLOW",
      "name": "areactl_2",
      "description": ""
    },
    {
      "enable": true,
      "boardAndCountry": [
        "非洲地区/贝宁"
      ],
      "srcZones": [
        "zoneL2_1",
        "zoneL3_1",
        "zoneVirtual_1"
      ],
      "dstIPGroups": [
        "ipv4_1",
        "ipv4_2",
        "ipv4_3"
      ],
      "uuid": "37D2C6E86597489897D657CB37DA6F97",
      "action": "ALLOW",
    }
  ]
}
```

```
    "name": "areactl_3",  
    "description": ""  
  }  
],  
"code": 0,  
"message": "成功"  
}
```

请求失败样例:

```
{  
  "data": "",  
  "code": 22,  
  "message": "无效参数"  
}
```

4.2.2.1.8 批量修改地域访问控制策略

功能说明: 通过此接口, 根据名称批量修改对应的地域访问控制策略

1) API请求数据

PATCH <https://{host}/api/batch/v1/namespaces/@namespace/areactlpolicys>

2) 输入参数

• URL参数

参数	类型	是否必选	描述
_where	string	否	指定位置, 用于移动/插入 枚举值:[top:顶部 bottom:底部 before:在目标位置之前 after:在目标位置之后]
_dest	string	否	目标位置关键字对应的值 最大长度:100
_key	string	否	目标位置关键字 枚举值:[position:指定目标位置关键字为 position name:指定目标位置关键字为 name]

• Body参数

参数	类型	是否必选	描述
policys	object[]	是	地域访问控制策略
policys.name	string	是	名称 最小长度:1 最大长度:95
policys.description	string	否	描述 最小长度:0 最大长度:95
policys.enable	bool	否	是否启用 默认值:true

polycys.action	string	否	控制方式 默认值:true 枚举值:[ALLOW:只允许以下国家/地区访问 DENY:拒绝以下国家/地区访问]
polycys.srcZones	string[]	否	源外网区域
polycys.dstIPGroups	string[]	否	目的网络对象
polycys.boardAndCountry	string[]	否	国家/地区
polycys.position	uint32	否	策略位置

请求样例:

批量修改地域访问控制策略

PATCH https://192.168.1.1/api/v1/namespaces/public/areactlpolycys

```
[
  {
    "name": "areactl_1",
    "enable": false
  },
  {
    "name": "areactl_2",
    "enable": false
  },
  {
    "name": "areactl_3",
    "enable": false
  }
]
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	地域访问控制策略
data.uuid	string	唯一标识id
data.name	string	名称 最小长度:1 最大长度:95
data.description	string	描述 最小长度:0 最大长度:95
data.enable	bool	是否启用
data.action	string	控制方式 枚举值:[ALLOW:只允许以下国家/地区访问 DENY:拒绝以下国家/地区访问]
data.srcZones	string[]	源外网区域
data.dstIPGroups	string[]	目的网络对象

data.boardAndCountry	string[]	国家/地区
data.position	uint32	策略位置

请求成功样例:

```
{
  "data": [
    {
      "enable": false,
      "boardAndCountry": [
        "非洲地区/贝宁"
      ],
      "srcZones": [
        "zoneL2_1",
        "zoneL3_1",
        "zoneVirtual_1"
      ],
      "dstIPGroups": [
        "ipv4_1",
        "ipv4_2",
        "ipv4_3"
      ],
      "uuid": "CC63C544946244458E9C37FAD7E6D28D",
      "action": "ALLOW",
      "name": "areactl_1",
      "description": ""
    },
    {
      "enable": false,
      "boardAndCountry": [
        "非洲地区/贝宁"
      ],
      "srcZones": [
        "zoneL2_1",
        "zoneL3_1",
        "zoneVirtual_1"
      ],
      "dstIPGroups": [
        "ipv4_1",
        "ipv4_2",
        "ipv4_3"
      ],
      "uuid": "836A8D7E906C433DAE166193FB10AB2E",
      "action": "ALLOW",
      "name": "areactl_2",
      "description": ""
    },
    {
      "enable": false,
      "boardAndCountry": [
        "非洲地区/贝宁"
      ],
      "srcZones": [
        "zoneL2_1",
        "zoneL3_1",
        "zoneVirtual_1"
      ],
      "dstIPGroups": [
        "ipv4_1",
        "ipv4_2",
        "ipv4_3"
      ],
      "uuid": "59B39375517844D38F56EB38913A4346",
      "action": "ALLOW",
      "name": "areactl_3",
      "description": ""
    }
  ],
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.2.2.2 排除列表

4.2.2.2.1 查看地域访问控制排除列表

功能说明：通过此接口，获取地域访问控制排除列表

1) API请求数据

GET <https://{host}/api/v1/namespaces/@namespace/areactlwhitelist>

2) 输入参数

请求样例：

查看地域访问控制排除列表

GET <https://192.168.1.1/api/v1/namespaces/public/areactlwhitelist>

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	排除列表，排除列表的IP地址不受“地域访问控制”的约束。
data.ipList	string[]	IPv4地址

请求成功样例：

```
{
  "data": {
    "ipList": [
      "200.200.155.1",
      "200.200.155.2"
    ]
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.2.2.2.2 修改地域访问控制排除列表

功能说明：通过此接口，重置地域访问控制排除列表

1) API请求数据

PATCH <https://{host}/api/v1/namespaces/@namespace/areactlwhitelist>

2) 输入参数

• URL参数

参数	类型	是否必选	描述
_arrayop	string	否	对象属性是数组的操作控制 枚举值:[add:添加对应的数组元素 remove:删除对应的数组元素]

• Body参数

参数	类型	是否必选	描述
obj	object	是	排除列表，排除列表的IP地址不受“地域访问控制”的约束。
obj.ipList	string[]	是	IPv4地址

请求样例：

修改地域访问控制排除列表

```
PATCH https://192.168.1.1/api/v1/namespaces/public/areactlwhitelist?_arrayop=add
{
  "ipList": [
    "200.200.155.1",
    "200.200.155.2"
  ]
}
```

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	排除列表，排除列表的IP地址不受“地域访问控制”的约束。
data.ipList	string[]	IPv4地址

请求成功样例：

```
{
  "data": {
    "ipList": [
      "200.200.155.1",
      "200.200.155.2"
    ]
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.2.2.3 已拒绝的IP

4.2.2.3.1 查看已拒绝的IP

功能说明：查询所有的已拒绝IP信息(共保留7天)的数据

1) API请求数据

GET

https://{host}/api/v1/namespaces/@namespace/rejip/records

2) 输入参数

- URL 参数

参数	类型	是否必选	描述
_start	int32	否	起始位置, 从 0 开始 最小值:0 最大值:2000000
_search	string	否	模糊搜索关键字 最大长度:95
_length	int32	否	获取最大长度, 限制最大值, 如果数据不够可能返回小于 length 最小值:1 最大值:200 默认值:true

请求样例:

查看已拒绝的IP

GET https://192.168.1.1/api/v1/namespaces/public/rejip/records

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	分页数据
data.totalItems	int32	总共多少项目
data.totalPages	int32	总共多少页
data.pageNumber	int32	当前页码, 从 1 开始
data.pageSize	int32	每页多大
data.itemsOffset	int32	当前条目偏移, 从 0 开始
data.itemLength	int32	数据列表长度
data.items	object[]	有效数据列表
data.items.ip	string	IPV4 address
data.items.location	string	地址
data.items.rejectDate	string	日期
data.items.rejectCount	int32	拒绝次数

请求成功样例:

```
{
  "data": {
    "itemsOffset": 0,
    "items": [],
    "totalItems": 0,
    "pageNumber": 1,
    "itemLength": 0,
    "totalPages": 1,
    "pageSize": 100
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.2.2.3.2 清空已拒绝IP列表

功能说明: 清空所有的已拒绝IP信息(共保留7天)的数据

1) API请求数据

POST https://{host}/api/v1/namespaces/@namespace/rejip/clearall

2) 输入参数

• Body参数

参数	类型	是否必选	描述
obj	string	是	

请求样例:

清空所有的已拒绝IP信息(共保留7天)的数据

POST https://192.168.1.1/api/v1/namespaces/public/rejip/clearall

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	string	

请求成功样例:

```
{
  "data": "success",
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
}
```

```
    "message": "无效参数"
  }
```

4.2.2.3.3 清零一个已拒绝IP的阻断次数

功能说明：将已拒绝IP列表中的某个IP的阻断次数清零

1) API请求数据

PATCH https://{host}/api/v1/namespaces/@namespace/rejip/records

2) 输入参数

• Body参数

参数	类型	是否必选	描述
obj	object	是	复位已拒绝IP的拒绝次数
obj.ip	string	是	IPV4 address
obj.rejectDate	string	是	日期
obj.rejectCount	int32	是	拒绝次数

请求样例：

将已拒绝IP列表中的某个IP的阻断次数清零

```
PATCH https://192.168.1.1/api/v1/namespaces/public/rejip/records
{
  "rejectDate": "今天",
  "ip": "198.168.1.104",
  "rejectCount": 0
}
```

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	string	

请求成功样例：

```
{
  "data": "success",
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.2.2.4 内网访问控制

4.2.2.4.1 查看内网访问控制策略

功能说明：通过此接口，查看内网访问控制策略

1) API请求数据

GET https://{host}/api/v1/namespaces/@namespace/lanctlpolicy

2) 输入参数

请求样例：

查看内网访问控制策略

GET https://192.168.1.1/api/v1/namespaces/public/lanctlpolicy

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	内网访问控制
data.enable	bool	是否启用
data.internalZones	string[]	内网配置/内网区域
data.IPGroups	string[]	内网配置/网络对象
data.allowPrivateIP	bool	是否勾选'放行私有IP网段(10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16)'

请求成功样例：

```
{
  "data": {
    "enable": true,
    "allowPrivateIP": false,
    "internalZones": [
      "zoneL3_1",
      "zoneL2_1",
      "zoneVirtual_1"
    ],
    "IPGroups": [
      "ipv4_1",
      "ipv4_2",
      "ipv4_3"
    ]
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.2.2.4.2 修改内网访问控制策略

功能说明：通过此接口，修改内网访问控制策略

1) API请求数据

PATCH https://{host}/api/v1/namespaces/@namespace/lanctlpolicy

2) 输入参数

• URL参数

参数	类型	是否必选	描述
_arrayop	string	否	对象属性是数组的操作控制 枚举值: add:添加对应的数组元素 remove:删除对应的数组元素]

• Body参数

参数	类型	是否必选	描述
lan	object	是	内网访问控制
lan.enable	bool	否	是否启用
lan.internalZones	string[]	否	内网配置/内网区域
lan.IPGroups	string[]	否	内网配置/网络对象
lan.allowPrivateIP	bool	否	是否勾选'放行私有IP网段(10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16)'

请求样例:

修改内网访问控制策略

```
PATCH https://192.168.1.1/api/v1/namespaces/public/lanctlpolicy
{
  "enable": true,
  "allowPrivateIP": false,
  "internalZones": [
    "zoneL3_1",
    "zoneL2_1",
    "zoneVirtual_1"
  ],
  "IPGroups": [
    "ipv4_1",
    "ipv4_2",
    "ipv4_3"
  ]
}
```

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	内网访问控制
data.enable	bool	是否启用
data.internalZones	string[]	内网配置/内网区域
data.IPGroups	string[]	内网配置/网络对象

data.allowPrivateIP	bool	是否勾选'放行私有IP网段(10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16)'
---------------------	------	---

请求成功样例:

```
{
  "data": {
    "enable": true,
    "allowPrivateIP": false,
    "internalZones": [
      "zoneL3_1",
      "zoneL2_1",
      "zoneVirtual_1"
    ],
    "IPGroups": [
      "ipv4_1",
      "ipv4_2",
      "ipv4_3"
    ]
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.2.3 连接数控制

4.2.3.1 获取连接数控制策略

功能说明: 通过此接口, 可以获取所有连接数控制策略

1) API请求数据

GET <https://{host}/api/v1/namespaces/@namespace/connctlpolicys>

2) 输入参数

- URL参数

参数	类型	是否必选	描述
_start	int32	否	起始位置, 从 0 开始 最小值:0 最大值:2000000
_length	int32	否	获取最大长度, 限制最大值, 如果数据不够可能返回小于 length 最小值:1 最大值:200 默认值:true

请求样例:

获取连接数控制策略, 查询条件: 起始位置为0, 最多展示100条

GET https://192.168.1.1/api/v1/namespaces/public/connctlpolicys?_start=0&_length=100

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	分页数据
data.totalItems	int32	总共多少项目
data.totalPages	int32	总共多少页
data.pageNumber	int32	当前页码, 从 1 开始
data.pageSize	int32	每页多大
data.itemsOffset	int32	当前条目偏移, 从 0 开始
data.itemLength	int32	数据列表长度
data.items	object[]	有效数据列表
data.items.uuid	string	唯一标识id
data.items.name	string	名称 最小长度:1 最大长度:95
data.items.description	string	描述 最小长度:0 最大长度:95
data.items.enable	bool	是否启用
data.items.policyType	string	类型 枚举值:[SRC:源IP连接数控制 DST:目的IP连接数控制 BIE:双向IP连接数控制]
data.items.srcZones	string[]	源区域
data.items.dstZones	string[]	目的区域
data.items.srcIPGroups	string[]	源网络对象
data.items.dstIPGroups	string[]	目的网络对象
data.items.maxConnNum	int32	最大并发连接数, 0为不限制 最小值:0 最大值:32768
data.items.position	uint32	策略位置

请求成功样例:

```
{
  "data": {
    "itemsOffset": 0,
    "items": [
      {
        "enable": false,
        "position": 0,
        "srcZones": [
          "z2",
          "z3",

```

```

        "z1"
      ],
      "srcIPGroups": [
        "ip1",
        "ip2",
        "ip3"
      ],
      "policyType": "SRC",
      "maxConnNum": 32768,
      "uuid": "786021EE1CCC4C29957AB80E1FC4C57C",
      "name": "src",
      "description": "srcccc"
    },
    {
      "enable": true,
      "position": 1,
      "dstIPGroups": [
        "ip1",
        "ip2",
        "ip3"
      ],
      "uuid": "38380AD650D74AACB6BD22A298E8D10D",
      "dstZones": [
        "z1",
        "z2",
        "z3"
      ],
      "maxConnNum": 0,
      "policyType": "DST",
      "name": "dst",
      "description": "dsttttt"
    },
    {
      "enable": false,
      "position": 2,
      "dstIPGroups": [
        "ip1",
        "ip2",
        "ip3"
      ],
      "srcZones": [
        "z1",
        "z2",
        "z3"
      ],
      "maxConnNum": 0,
      "srcIPGroups": [
        "ip1",
        "ip2",
        "ip3"
      ],
      "dstZones": [
        "z2",
        "z3",
        "z1"
      ],
      "policyType": "BIE",
      "uuid": "07D42F35CFDE4C60A9DFD869866DDF35",
      "name": "bie",
      "description": "bieeeee"
    }
  ],
  "totalItems": 3,
  "pageNumber": 1,
  "itemLength": 3,
  "totalPages": 1,
  "pageSize": 100
},
"code": 0,
"message": "成功"
}

```

请求失败样例：

```

{
  "data": "",
  "code": 22,

```

```
    "message": "无效参数"
}
```

4.2.3.2 获取一条连接数控制策略

功能说明：通过此接口，可以通过策略名查询到单条连接数控制策略

1) API请求数据

GET

https://{host}/api/v1/namespaces/@namespace/connctlpolicys/@name

2) 输入参数

请求样例：

通过策略名，获取一条连接数控制策略

GET https://192.168.1.1/api/v1/namespaces/public/connctlpolicys/src

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	连接数控制
data.uuid	string	唯一标识id
data.name	string	名称 最小长度:1 最大长度:95
data.description	string	描述 最小长度:0 最大长度:95
data.enable	bool	是否启用
data.policyType	string	类型 枚举值:[SRC:源IP连接数控制 DST:目的IP连接数控制 BIE:双向IP连接数控制]
data.srcZones	string[]	源区域
data.dstZones	string[]	目的区域
data.srcIPGroups	string[]	源网络对象
data.dstIPGroups	string[]	目的网络对象
data.maxConnNum	int32	最大并发连接数，0为不限制 最小值:0 最大值:32768
data.position	uint32	策略位置

请求成功样例：

```
{
  "data": {
    "enable": false,
```



```

    "position": 0,
    "srcZones": [
      "z2",
      "z3",
      "z1"
    ],
    "srcIPGroups": [
      "ip1",
      "ip2",
      "ip3"
    ],
    "policyType": "SRC",
    "maxConnNum": 32768,
    "uuid": "786021EE1CCC4C29957AB80E1FC4C57C",
    "name": "src",
    "description": "srcccc"
  },
  "code": 0,
  "message": "成功"
}

```

请求失败样例:

```

{
  "data": "",
  "code": 22,
  "message": "无效参数"
}

```

4.2.3.3 添加连接数控制策略

功能说明: 通过此接口, 可以添加一条连接数控制策略

1) API请求数据

POST <https://{host}/api/v1/namespaces/@namespace/connctlpolicys>

2) 输入参数

• Body参数

参数	类型	是否必选	描述
policy	object	是	连接数控制
policy.name	string	是	名称 最小长度:1 最大长度:95
policy.description	string	否	描述 最小长度:0 最大长度:95
policy.enable	bool	否	是否启用 默认值:true
policy.policyType	string	否	类型 枚举值:[SRC:源IP连接数控制 DST:目的IP连接数控制 BIE:双向IP连接数控制]
policy.srcZones	string[]	否	源区域
policy.dstZones	string[]	否	目的区域
policy.srcIPGroups	string[]	否	源网络对象
policy.dstIPGroups	string[]	否	目的网络对象

policy.maxConnNum	int32	否	最大并发连接数，0为不限制 最小值:0 最大值:32768 默认值:true
policy.position	uint32	否	策略位置

请求样例:

添加连接数控制策略

POST https://192.168.1.1/api/v1/namespaces/public/connctlpolicys

```
{
  "enable": true,
  "maxConnNum": 32768,
  "srcIPGroups": [
    "ipv4_1"
  ],
  "policyType": "SRC",
  "srcZones": [
    "zoneL2_1",
    "zoneL3_1",
    "zoneVirtual_1"
  ],
  "name": "conn_src_1",
  "description": ""
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	连接数控制
data.uuid	string	唯一标识id
data.name	string	名称 最小长度:1 最大长度:95
data.description	string	描述 最小长度:0 最大长度:95
data.enable	bool	是否启用
data.policyType	string	类型 枚举值:[SRC:源IP连接数控制 DST:目的IP连接数控制 BIE:双向IP连接数控制]
data.srcZones	string[]	源区域
data.dstZones	string[]	目的区域
data.srcIPGroups	string[]	源网络对象
data.dstIPGroups	string[]	目的网络对象

data.maxConnNum	int32	最大并发连接数，0为不限制 最小值:0 最大值:32768
data.position	uint32	策略位置

请求成功样例：

```
{
  "data": {
    "enable": true,
    "maxConnNum": 32768,
    "srcIPGroups": [
      "ipv4_1"
    ],
    "srcZones": [
      "zoneL2_1",
      "zoneL3_1",
      "zoneVirtual_1"
    ],
    "policyType": "SRC",
    "uuid": "BE6CF3C6D27944C2A958668183BCC47E",
    "name": "conn_src_1",
    "description": ""
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.2.3.4 修改连接数控制策略

功能说明：通过此接口，可以通过策略名，修改对应的连接数控制策略

1) API请求数据

PATCH

https://{host}/api/v1/namespaces/@namespace/connctlpolicys/@name

2) 输入参数

• URL 参数

参数	类型	是否必选	描述
_arrayop	string	否	对象属性是数组的操作控制 枚举值:[add:添加对应的数组元素 remove:删除对应的数组元素]
_key	string	否	目标位置关键字 枚举值:[position:指定目标位置关键字为 position name:指定目标位置关键字为 name]
_dest	string	否	目标位置关键字对应的值 最大长度:100

_where	string	否	指定位置, 用于移动/插入 枚举值:[top:顶部 bottom:底部 before:在目标位置之前 after:在目标位置之后]
--------	--------	---	---

- Body参数

参数	类型	是否必选	描述
policy	object	是	连接数控制
policy.name	string	是	名称 最小长度:1 最大长度:95
policy.description	string	否	描述 最小长度:0 最大长度:95
policy.enable	bool	否	是否启用 默认值:true
policy.policyType	string	否	类型 枚举值:[SRC:源IP连接数控制 DST:目的IP连接数控制 BIE:双向IP连接数控制]
policy.srcZones	string[]	否	源区域
policy.dstZones	string[]	否	目的区域
policy.srcIPGroups	string[]	否	源网络对象
policy.dstIPGroups	string[]	否	目的网络对象
policy.maxConnNum	int32	否	最大并发连接数, 0为不限制 最小值:0 最大值:32768 默认值:true
policy.position	uint32	否	策略位置

请求样例:

修改连接数控制策略

```
PATCH https://192.168.1.1/api/batch/v1/namespaces/public/conn_src_1
{
  "maxConnNum": 550
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息

data	object	连接数控制
data.uuid	string	唯一标识id
data.name	string	名称 最小长度:1 最大长度:95
data.description	string	描述 最小长度:0 最大长度:95
data.enable	bool	是否启用
data.policyType	string	类型 枚举值:[SRC:源IP连接数控制 DST:目的IP连接数控制 BIE:双向IP连接数控制]
data.srcZones	string[]	源区域
data.dstZones	string[]	目的区域
data.srcIPGroups	string[]	源网络对象
data.dstIPGroups	string[]	目的网络对象
data.maxConnNum	int32	最大并发连接数, 0为不限制 最小值:0 最大值:32768
data.position	uint32	策略位置

请求成功样例:

```
{
  "data": {
    "enable": true,
    "maxConnNum": 550,
    "srcIPGroups": [
      "ipv4_1"
    ],
    "srcZones": [
      "zoneL2_1",
      "zoneL3_1",
      "zoneVirtual_1"
    ],
    "policyType": "SRC",
    "uuid": "BE6CF3C6D27944C2A958668183BCC47E",
    "name": "conn_src_1",
    "description": ""
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.2.3.5 全量修改连接数控制策略

功能说明：通过此接口，可以通过策略名，修改对应的连接数控制策略

1) API请求数据

PUT

https://{host}/api/v1/namespaces/@namespace/connctlpolicys/@name

2) 输入参数

- Body参数

参数	类型	是否必选	描述
policy	object	是	连接数控制
policy.name	string	是	名称 最小长度:1 最大长度:95
policy.description	string	否	描述 最小长度:0 最大长度:95
policy.enable	bool	否	是否启用 默认值:true
policy.policyType	string	否	类型 枚举值:[SRC:源IP连接数控制 DST:目的IP连接数控制 BIE:双向IP连接数控制]
policy.srcZones	string[]	否	源区域
policy.dstZones	string[]	否	目的区域
policy.srcIPGroups	string[]	否	源网络对象
policy.dstIPGroups	string[]	否	目的网络对象
policy.maxConnNum	int32	否	最大并发连接数，0为不限制 最小值:0 最大值:32768 默认值:true
policy.position	uint32	否	策略位置

请求样例：

全量修改连接数控制策略

```
PUT https://192.168.1.1/api/batch/v1/namespaces/public/conn_src_1
{
  "enable": true,
  "maxConnNum": 32768,
  "srcIPGroups": [
    "ipv4_1"
  ],
  "policyType": "SRC",
  "srcZones": [
    "zoneL2_1",
    "zoneL3_1",
    "zoneVirtual_1"
  ],
  "name": "conn_src_1",
  "description": ""
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	连接数控制
data.uuid	string	唯一标识id
data.name	string	名称 最小长度:1 最大长度:95
data.description	string	描述 最小长度:0 最大长度:95
data.enable	bool	是否启用
data.policyType	string	类型 枚举值:[SRC:源IP连接数控制 DST:目的IP连接数控制 BIE:双向IP连接数控制]
data.srcZones	string[]	源区域
data.dstZones	string[]	目的区域
data.srcIPGroups	string[]	源网络对象
data.dstIPGroups	string[]	目的网络对象
data.maxConnNum	int32	最大并发连接数，0为不限制 最小值:0 最大值:32768
data.position	uint32	策略位置

请求成功样例：

```
{
  "data": {
    "enable": true,
    "maxConnNum": 32768,
    "srcIPGroups": [
      "ipv4_1"
    ],
    "srcZones": [
      "zoneL2_1",
      "zoneL3_1",
      "zoneVirtual_1"
    ],
    "policyType": "SRC",
    "uuid": "BE6CF3C6D27944C2A958668183BCC47E",
    "name": "conn_src_1",
    "description": ""
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.2.3.6 删除一条连接数控制策略

功能说明：通过此接口，可以通过策略名，删除对应的连接数控制策略

1) API请求数据

DELETE <https://{host}/api/v1/namespaces/@namespace/connctlpolicys/@name>

2) 输入参数

请求样例：

删除一条连接数控制策略

DELETE https://192.168.1.1/api/v1/namespaces/public/connctlpolicys/conn_src_1

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	连接数控制
data.uuid	string	唯一标识id
data.name	string	名称 最小长度:1 最大长度:95
data.description	string	描述 最小长度:0 最大长度:95
data.enable	bool	是否启用
data.policyType	string	类型 枚举值:[SRC:源IP连接数控制 DST:目的IP连接数控制 BIE:双向IP连接数控制]
data.srcZones	string[]	源区域
data.dstZones	string[]	目的区域
data.srcIPGroups	string[]	源网络对象
data.dstIPGroups	string[]	目的网络对象
data.maxConnNum	int32	最大并发连接数，0为不限制 最小值:0 最大值:32768
data.position	uint32	策略位置

请求成功样例：


```
{
  "data": {
    "enable": true,
    "maxConnNum": 32768,
    "srcIPGroups": [
      "ipv4_1"
    ],
    "srcZones": [
      "zoneL2_1",
      "zoneL3_1",
      "zoneVirtual_1"
    ],
    "policyType": "SRC",
    "uuid": "BF372CB677E444D499E9EBB11BD76265",
    "name": "conn_src_2",
    "description": ""
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.2.3.7 批量删除连接数控制策略

功能说明：通过此接口，可以通过策略名，批量删除对应的连接数控制策略

1) API请求数据

POST

https://{host}/api/batch/v1/namespaces/@namespace/connctlpolicys?_method=delete

2) 输入参数

• Body参数

参数	类型	是否必选	描述
policys	object[]	是	连接数控制
policys.name	string	是	名称 最小长度:1 最大长度:95
policys.description	string	否	描述 最小长度:0 最大长度:95
policys.enable	bool	否	是否启用 默认值:true
policys.policyType	string	否	类型 枚举值:[SRC:源IP连接数控制 DST:目的IP连接数控制 BIE:双向IP连接数控制]
policys.srcZones	string[]	否	源区域
policys.dstZones	string[]	否	目的区域
policys.srcIPGroups	string[]	否	源网络对象

polycys.dstIPGroups	string[]	否	目的网络对象
polycys.maxConnNum	int32	否	最大并发连接数，0为不限制 最小值:0 最大值:32768 默认值:true
polycys.position	uint32	否	策略位置

请求样例:

批量删除连接数控制策略

```
POST https://192.168.1.1/api/batch/v1/namespaces/public/connctlpolycys?_method=DELETE
[
  {
    "name": "conn_src_1"
  },
  {
    "name": "conn_src_2"
  },
  {
    "name": "conn_src_3"
  }
]
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	连接数控制
data.uuid	string	唯一标识id
data.name	string	名称 最小长度:1 最大长度:95
data.description	string	描述 最小长度:0 最大长度:95
data.enable	bool	是否启用
data.policyType	string	类型 枚举值:[SRC:源IP连接数控制 DST:目的IP连接数控制 BIE:双向IP连接数控制]
data.srcZones	string[]	源区域
data.dstZones	string[]	目的区域
data.srcIPGroups	string[]	源网络对象
data.dstIPGroups	string[]	目的网络对象
data.maxConnNum	int32	最大并发连接数，0为不限制 最小值:0 最大值:32768

data.position	uint32	策略位置
---------------	--------	------

请求成功样例：

```
{
  "data": [
    {
      "enable": true,
      "maxConnNum": 32768,
      "srcIPGroups": [
        "ipv4_1"
      ],
      "srcZones": [
        "zoneL2_1",
        "zoneL3_1",
        "zoneVirtual_1"
      ],
      "policyType": "SRC",
      "uuid": "39F15DD061724F2897985578473BFE8E",
      "name": "conn_src_1",
      "description": ""
    },
    {
      "enable": true,
      "maxConnNum": 32768,
      "srcIPGroups": [
        "ipv4_1"
      ],
      "srcZones": [
        "zoneL2_1",
        "zoneL3_1",
        "zoneVirtual_1"
      ],
      "policyType": "SRC",
      "uuid": "40254F7272FA417CA986FCCBD719293E",
      "name": "conn_src_2",
      "description": ""
    },
    {
      "enable": true,
      "maxConnNum": 32768,
      "srcIPGroups": [
        "ipv4_1"
      ],
      "srcZones": [
        "zoneL2_1",
        "zoneL3_1",
        "zoneVirtual_1"
      ],
      "policyType": "SRC",
      "uuid": "86B4C23924174220917106C0327889AC",
      "name": "conn_src_3",
      "description": ""
    }
  ],
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.2.3.8 批量修改连接数控制策略

功能说明：通过此接口，可以通过策略名，批量修改对应的连接数控制策略

1) API请求数据

PATCH https://{host}/api/batch/v1/namespaces/@namespace/connctlpolicys

2) 输入参数

• URL参数

参数	类型	是否必选	描述
_where	string	否	指定位置, 用于移动/插入 枚举值:[top:顶部 bottom:底部 before:在目标位置之前 after:在目标位置之后]
_dest	string	否	目标位置关键字对应的值 最大长度:100
_key	string	否	目标位置关键字 枚举值:[position:指定目标位置关键字为 position name:指定目标位置关键字为 name]

• Body参数

参数	类型	是否必选	描述
policys	object[]	是	连接数控制
policys.name	string	是	名称 最小长度:1 最大长度:95
policys.description	string	否	描述 最小长度:0 最大长度:95
policys.enable	bool	否	是否启用 默认值:true
policys.policyType	string	否	类型 枚举值:[SRC:源IP连接数控制 DST:目的IP连接数控制 BIE:双向IP连接数控制]
policys.srcZones	string[]	否	源区域
policys.dstZones	string[]	否	目的区域
policys.srcIPGroups	string[]	否	源网络对象
policys.dstIPGroups	string[]	否	目的网络对象
policys.maxConnNum	int32	否	最大并发连接数, 0为不限制 最小值:0 最大值:32768 默认值:true
policys.position	uint32	否	策略位置

请求样例:

```
批量修改连接数控制策略
PATCH https://192.168.1.1/api/v1/namespaces/public/connctlpolicys
[
  {
    "name": "conn_src_1",
    "enable": false
  },
  {
    "name": "conn_src_2",
    "enable": false
  },
  {
    "name": "conn_src_3",
    "enable": false
  }
]
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	连接数控制
data.uuid	string	唯一标识id
data.name	string	名称 最小长度:1 最大长度:95
data.description	string	描述 最小长度:0 最大长度:95
data.enable	bool	是否启用
data.policyType	string	类型 枚举值:[SRC:源IP连接数控制 DST:目的IP连接数控制 BIE:双向IP连接数控制]
data.srcZones	string[]	源区域
data.dstZones	string[]	目的区域
data.srcIPGroups	string[]	源网络对象
data.dstIPGroups	string[]	目的网络对象
data.maxConnNum	int32	最大并发连接数，0为不限制 最小值:0 最大值:32768
data.position	uint32	策略位置

请求成功样例:

```
{
  "data": [
    {
```

```
    "enable": false,
    "maxConnNum": 32768,
    "srcIPGroups": [
      "ipv4_1"
    ],
    "srcZones": [
      "zoneL2_1",
      "zoneL3_1",
      "zoneVirtual_1"
    ],
    "policyType": "SRC",
    "uuid": "3DEF5349788E4C33B75449B77B7D9D28",
    "name": "conn_src_1",
    "description": ""
  },
  {
    "enable": false,
    "maxConnNum": 32768,
    "srcIPGroups": [
      "ipv4_1"
    ],
    "srcZones": [
      "zoneL2_1",
      "zoneL3_1",
      "zoneVirtual_1"
    ],
    "policyType": "SRC",
    "uuid": "4F14E8BB878649D596E7F1EB7DA23E57",
    "name": "conn_src_2",
    "description": ""
  },
  {
    "enable": false,
    "maxConnNum": 32768,
    "srcIPGroups": [
      "ipv4_1"
    ],
    "srcZones": [
      "zoneL2_1",
      "zoneL3_1",
      "zoneVirtual_1"
    ],
    "policyType": "SRC",
    "uuid": "483D2A8F9B2C489AB7D0505D4227791B",
    "name": "conn_src_3",
    "description": ""
  }
],
"code": 0,
"message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.3 认证

4.3.1 用户组管理

4.3.1.1 认证用户

4.3.1.1.1 查看某个组下的认证用户列表

功能说明：获取某个组的认证用户列表

1) API请求数据

POST https://{host}/api/v1/namespaces/@namespace/authusers?_method=get

2) 输入参数

• URL参数

参数	类型	是否必选	描述
_start	int32	否	起始位置, 从 0 开始 最小值:0 最大值:2000000
_length	int32	否	获取最大长度, 限制最大值, 如果数据不够可能返回小于 length 最小值:1 最大值:200 默认值:true

• Body参数

参数	类型	是否必选	描述
obj	object	是	搜索条件设置
obj.path	string	是	设置搜索哪一路径下的用户,例如/test/ 最小长度:1 最大长度:1440
obj.baseFilter	object	否	基本搜索条件
obj.baseFilter.search	string	否	基本搜索条件选择 默认值:true 枚举值:[NAME:ip、mac、name三选一, 按照name过滤 IP:ip、mac、name三选一, 按照ip过滤 MAC:ip、mac、name三选一, 按照mac过滤]
obj.baseFilter.ip	object	是	ip、mac、name三选一, 按照ip过滤 格式:ip-range
obj.baseFilter.ip.start	string	是	ip范围的起始位置 格式:ip-address
obj.baseFilter.ip.end	string	否	ip范围的终止位置 格式:ip-address
obj.baseFilter.ip.bits	uint8	否	ip掩码简写
obj.baseFilter.ip.mask	string	否	ip掩码 格式:ip-address
obj.baseFilter.mac	string	是	ip、mac、name三选一, 按照mac过滤
obj.baseFilter.name	string	是	ip、mac、name三选一, 按照name过滤 最小长度:1 最大长度:95
obj.expireTime	object	是	按照账号过期时间过滤
obj.expireTime.enable	bool	否	是否开启过期时间开关, 默认关闭

obj.expireTime.start	string	是	过期时间区间起始值
obj.expireTime.end	string	是	过期时间区间结束值
obj.userStatus	string	否	按照用户启禁用状态过滤，默认是任意状态 默认值:true 枚举值:[ALL:任意状态 ENABLE:启用状态 DISABLE:禁用状态]
obj.multipleLogin	bool	否	按照是否可以多人同时登录过滤，默认关闭

请求样例:

查看认证用户列表, 查询条件: 查询用户组test下, 用户状态为任意状态, 用户名或显示名为user1的用户, 结果展示前100条

```
{
  "path": "/test/",
  "userStatus": "ALL",
  "baseFilter": {
    "name": "user1",
    "search": "NAME"
  }
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	分页数据
data.totalItems	int32	总共多少项目
data.totalPages	int32	总共多少页
data.pageNumber	int32	当前页码, 从 1 开始
data.pageSize	int32	每页多大
data.itemsOffset	int32	当前条目偏移, 从 0 开始
data.itemLength	int32	数据列表长度
data.items	object[]	有效数据列表
data.items.name	string	用户名称 最小长度:1 最大长度:95
data.items.uuid	string	用户uuid
data.items.displayName	string	用户的显示名称 最小长度:0 最大长度:95
data.items.path	string	用户路径 最小长度:1 最大长度:1440

data.items.description	string	用户描述 最小长度:0 最大长度:95
data.items.userEnable	bool	用户启禁用状态
data.items.localPasswd	object	可选，本地密码设置
data.items.localPasswd.enable	bool	是否启用本地密码设置
data.items.localPasswd.password	string	本地密码 最小长度:1 最大长度:40
data.items.ipmacBind	object	可选，地址绑定设置
data.items.ipmacBind.enable	bool	是否启用地址绑定
data.items.ipmacBind.bindType	string	绑定方式,分为单向和双向 枚举值:[DUPLEX:地址和用户双向绑定 SIMPLEX:地址和用户单向绑定]
data.items.ipmacBind.bindAddr	string	绑定的地址选择 枚举值:[IP:绑定IP地址，MAC地址不绑定 MAC:绑定MAC地址，IP地址不绑定 IPMAC:绑定IP和MAC地址]
data.items.ipmacBind.bindIp	object[]	绑定的ip地址
data.items.ipmacBind.bindIp.start	string	ip范围的起始位置 格式:ip-address
data.items.ipmacBind.bindIp.end	string	ip范围的终止位置 格式:ip-address
data.items.ipmacBind.bindIp.bits	uint8	ip掩码简写
data.items.ipmacBind.bindIp.mask	string	ip掩码 格式:ip-address
data.items.ipmacBind.bindMac	string[]	绑定的mac地址
data.items.ipmacBind.bindIpMac	object[]	绑定的ip和mac地址列表
data.items.ipmacBind.bindIpMac.ip	string	IP地址信息
data.items.ipmacBind.bindIpMac.mac	string	MAC地址信息
data.items.multipleLogin	object	可选，多人同时登录设置
data.items.multipleLogin.enable	bool	是否开启多人同时登录

data.items.multipleLogin.loginNumber	uint8	可以同时登录的人数 最小值:1 最大值:5
data.items.logoutDisply	bool	可选, 登录成功后是否弹出注销窗口
data.items.autoLogout	object	可选, 设置无流量自动注销时间
data.items.autoLogout.enable	bool	是否开启注销时间限制
data.items.autoLogout.logoutTime	uint32	多长时间无流量自动注销, 分钟为单位 最小值:1 最大值:65535
data.items.expireTime	object	可选, 设置账号过期时间
data.items.expireTime.status	string	账号过期的类型选择 枚举值: NEVER:永不过期 EXPIRE:需要设置过期的日期]
data.items.expireTime.expireDate	string	过期的日期设置

请求成功样例:

```
{
  "data": {
    "itemsOffset": 0,
    "items": [
      {
        "expireTime": {
          "status": "NEVER"
        },
        "logoutDisply": false,
        "localPasswd": {
          "enable": false
        },
        "autoLogout": {
          "enable": true,
          "logoutTime": 120
        },
        "uuid": "F656CAED17D34FB19B615D0E9B4554EB",
        "path": "/test/",
        "multipleLogin": {
          "enable": false,
          "loginNumber": 5
        },
        "displayName": "",
        "name": "user1",
        "ipmacBind": {
          "enable": true,
          "bindAddr": "IP",
          "bindIp": [
            {
              "start": "192.168.1.1",
              "end": "192.168.1.1"
            }
          ]
        },
        "bindType": "SIMPLEX"
      },
      "userEnable": true,
      "description": ""
    ]
  },
  "totalItems": 1,
  "pageNumber": 1,
  "itemLength": 1,
  "totalPages": 1,
  "pageSize": 100
}
```

```
    },
    "code": 0,
    "message": "成功"
  }
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.3.1.1.2 获取一个认证用户

功能说明：获取一个认证用户

1) API请求数据

GET

https://{host}/api/v1/namespaces/@namespace/authusers/@name

2) 输入参数

请求样例：

获取一个认证用户test的信息

GET https://192.168.1.1/api/v1/namespaces/public/authusers/test

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	认证用户信息
data.name	string	用户名称 最小长度:1 最大长度:95
data.uuid	string	用户uuid
data.displayName	string	用户的显示名称 最小长度:0 最大长度:95
data.path	string	用户路径 最小长度:1 最大长度:1440
data.description	string	用户描述 最小长度:0 最大长度:95
data.userEnable	bool	用户启禁用状态
data.localPasswd	object	可选，本地密码设置
data.localPasswd.enable	bool	是否启用本地密码设置
data.localPasswd.password	string	本地密码 最小长度:1 最大长度:40

data.ipmacBind	object	可选，地址绑定设置
data.ipmacBind.enable	bool	是否启用地址绑定
data.ipmacBind.bindType	string	绑定方式,分为单向和双向 枚举值:[DUPLEX:地址和用户双向绑定 SIMPLEX:地址和用户单向绑定]
data.ipmacBind.bindAddr	string	绑定的地址选择 枚举值:[IP:绑定IP地址，MAC地址不绑定 MAC:绑定MAC地址，IP地址不绑定 IPMAC:绑定IP和MAC地址]
data.ipmacBind.bindIp	object[]	绑定的ip地址
data.ipmacBind.bindIp.start	string	ip范围的起始位置 格式:ip-address
data.ipmacBind.bindIp.end	string	ip范围的终止位置 格式:ip-address
data.ipmacBind.bindIp.bits	uint8	ip掩码简写
data.ipmacBind.bindIp.mask	string	ip掩码 格式:ip-address
data.ipmacBind.bindMac	string[]	绑定的mac地址
data.ipmacBind.bindIpMac	object[]	绑定的ip和mac地址列表
data.ipmacBind.bindIpMac.ip	string	IP地址信息
data.ipmacBind.bindIpMac.mac	string	MAC地址信息
data.multipleLogin	object	可选，多人同时登录设置
data.multipleLogin.enable	bool	是否开启多人同时登录
data.multipleLogin.loginNumber	uint8	可以同时登录的人数 最小值:1 最大值:5
data.logoutDisply	bool	可选，登录成功后是否弹出注销窗口
data.autoLogout	object	可选，设置无流量自动注销时间
data.autoLogout.enable	bool	是否开启注销时间限制
data.autoLogout.logoutTime	uint32	多长时间无流量自动注销，分钟为单位 最小值:1 最大值:65535
data.expireTime	object	可选，设置账号过期时间

data.expireTime.status	string	账号过期的类型选择 枚举值:[NEVER:永不过期 EXPIRE:需要设置过期的日期]
data.expireTime.expireDate	string	过期的日期设置

请求成功样例：

```
{
  "data": {
    "expireTime": {
      "status": "NEVER"
    },
    "logoutDisply": true,
    "localPasswd": {
      "enable": false
    },
    "autoLogout": {
      "enable": true,
      "logoutTime": 120
    },
    "path": "/",
    "userEnable": true,
    "name": "t1",
    "displayName": "",
    "ipmacBind": {
      "enable": false,
      "bindAddr": "IP",
      "bindType": "DUPLEX"
    },
    "multipleLogin": {
      "enable": false,
      "loginNumber": 1
    },
    "description": ""
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.3.1.1.3 添加一个认证用户

功能说明：添加一个认证用户

1) API请求数据

POST

https://{host}/api/v1/namespaces/@namespace/authusers

2) 输入参数

• Body参数

参数	类型	是否必选	描述
obj	object	是	认证用户信息
obj.name	string	是	用户名称 最小长度:1 最大长度:95

obj.displayName	string	否	用户的显示名称 最小长度:0 最大长度:95
obj.path	string	否	用户路径 最小长度:1 最大长度:1440
obj.description	string	否	用户描述 最小长度:0 最大长度:95
obj.userEnable	bool	否	用户启禁用状态 默认值:true
obj.localPasswd	object	否	可选, 本地密码设置
obj.localPasswd.enable	bool	否	是否启用本地密码设置
obj.localPasswd.password	string	是	本地密码 最小长度:1 最大长度:40
obj.ipmacBind	object	否	可选, 地址绑定设置
obj.ipmacBind.enable	bool	否	是否启用地址绑定
obj.ipmacBind.bindType	string	否	绑定方式,分为单向和双向 默认值:true 枚举值:[DUPLEX:地址和用户双向绑定 SIMPLEX:地址和用户单向绑定]
obj.ipmacBind.bindAddr	string	否	绑定的地址选择 默认值:true 枚举值:[IP:绑定IP地址, MAC地址不绑定 MAC:绑定MAC地址, IP地址不绑定 IPMAC:绑定IP和MAC地址]
obj.ipmacBind.bindIp	object[]	是	绑定的ip地址
obj.ipmacBind.bindIp.start	string	是	ip范围的起始位置 格式:ip-address
obj.ipmacBind.bindIp.end	string	否	ip范围的终止位置 格式:ip-address
obj.ipmacBind.bindIp.bits	uint8	否	ip掩码简写
obj.ipmacBind.bindIp.mask	string	否	ip掩码 格式:ip-address
obj.ipmacBind.bindMac	string[]	是	绑定的mac地址
obj.ipmacBind.bindIpMac	object[]	是	绑定的ip和mac地址列表
obj.ipmacBind.bindIpMac.ip	string	是	IP地址信息
obj.ipmacBind.bindIpMac.mac	string	是	MAC地址信息

obj.multipleLogin	object	否	可选，多人同时登录设置
obj.multipleLogin.enable	bool	否	是否开启多人同时登录
obj.multipleLogin.loginNumber	uint8	否	可以同时登录的人数 最小值:1 最大值:5 默认值:true
obj.logoutDisply	bool	否	可选，登录成功后是否弹出注销窗口
obj.autoLogout	object	否	可选，设置无流量自动注销时间
obj.autoLogout.enable	bool	否	是否开启注销时间限制 默认值:true
obj.autoLogout.logoutTime	uint32	否	多长时间无流量自动注销，分钟为单位 最小值:1 最大值:65535 默认值:true
obj.expireTime	object	否	可选，设置账号过期时间
obj.expireTime.status	string	否	账号过期的类型选择 默认值:true 枚举值:[NEVER:永不过期 EXPIRE:需要设置过期的日期]
obj.expireTime.expireDate	string	是	过期的日期设置

请求样例：

添加一个认证用户test

POST https://192.168.1.1/api/v1/namespaces/public/authusers

```
{
  "displayName": "test",
  "ipmacBind": {
    "enable": true,
    "bindAddr": "IP",
    "bindIp": [
      {
        "start": "192.168.1.1"
      }
    ],
    "bindType": "DUPLEX"
  },
  "name": "test",
  "path": "/"
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	认证用户信息

data.name	string	用户名称 最小长度:1 最大长度:95
data.uuid	string	用户uuid
data.displayName	string	用户的显示名称 最小长度:0 最大长度:95
data.path	string	用户路径 最小长度:1 最大长度:1440
data.description	string	用户描述 最小长度:0 最大长度:95
data.userEnable	bool	用户启禁用状态
data.localPasswd	object	可选，本地密码设置
data.localPasswd.enable	bool	是否启用本地密码设置
data.localPasswd.password	string	本地密码 最小长度:1 最大长度:40
data.ipmacBind	object	可选，地址绑定设置
data.ipmacBind.enable	bool	是否启用地址绑定
data.ipmacBind.bindType	string	绑定方式,分为单向和双向 枚举值:[DUPLEX:地址和用户双向绑定 SIMPLEX:地址和用户单向绑定]
data.ipmacBind.bindAddr	string	绑定的地址选择 枚举值:[IP:绑定IP地址，MAC地址不绑定 MAC:绑定MAC地址，IP地址不绑定 IPMAC:绑定IP和MAC地址]
data.ipmacBind.bindIp	object[]	绑定的ip地址
data.ipmacBind.bindIp.start	string	ip范围的起始位置 格式:ip-address
data.ipmacBind.bindIp.end	string	ip范围的终止位置 格式:ip-address
data.ipmacBind.bindIp.bits	uint8	ip掩码简写
data.ipmacBind.bindIp.mask	string	ip掩码 格式:ip-address
data.ipmacBind.bindMac	string[]	绑定的mac地址
data.ipmacBind.bindIpMac	object[]	绑定的ip和mac地址列表

data.ipmacBind.bindIpMac.ip	string	IP地址信息
data.ipmacBind.bindIpMac.mac	string	MAC地址信息
data.multipleLogin	object	可选，多人同时登录设置
data.multipleLogin.enable	bool	是否开启多人同时登录
data.multipleLogin.loginNumber	uint8	可以同时登录的人数 最小值:1 最大值:5
data.logoutDisply	bool	可选，登录成功后是否弹出注销窗口
data.autoLogout	object	可选，设置无流量自动注销时间
data.autoLogout.enable	bool	是否开启注销时间限制
data.autoLogout.logoutTime	uint32	多长时间无流量自动注销，分钟为单位 最小值:1 最大值:65535
data.expireTime	object	可选，设置账号过期时间
data.expireTime.status	string	账号过期的类型选择 枚举值: NEVER:永不过期 EXPIRE:需要设置过期的日期]
data.expireTime.expireDate	string	过期的日期设置

请求成功样例：

```
{
  "data": {
    "displayName": "test",
    "ipmacBind": {
      "enable": true,
      "bindAddr": "IP",
      "bindIp": [
        {
          "start": "192.168.1.1"
        }
      ],
      "bindType": "DUPLEX"
    },
    "name": "test",
    "path": "/"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.3.1.1.4 删除一个认证用户

功能说明：删除一个认证用户

1) API请求数据

DELETE https://{host}/api/v1/namespaces/@namespace/authusers/@name

2) 输入参数

请求样例：

删除一个认证用户test

DELETE https://192.168.1.1/api/v1/namespaces/public/authusers/test

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	认证用户信息
data.name	string	用户名称 最小长度:1 最大长度:95
data.uuid	string	用户uuid
data.displayName	string	用户的显示名称 最小长度:0 最大长度:95
data.path	string	用户路径 最小长度:1 最大长度:1440
data.description	string	用户描述 最小长度:0 最大长度:95
data.userEnable	bool	用户启禁用状态
data.localPasswd	object	可选，本地密码设置
data.localPasswd.enable	bool	是否启用本地密码设置
data.localPasswd.password	string	本地密码 最小长度:1 最大长度:40
data.ipmacBind	object	可选，地址绑定设置
data.ipmacBind.enable	bool	是否启用地址绑定
data.ipmacBind.bindType	string	绑定方式,分为单向和双向 枚举值:[DUPLEX:地址和用户双向绑定 SIMPLEX:地址和用户单向绑定]

data.ipmacBind.bindAddr	string	绑定的地址选择 枚举值:[IP:绑定IP地址, MAC地址不绑定 MAC:绑定MAC地址, IP地址不绑定 IPMAC:绑定IP和MAC地址]
data.ipmacBind.bindIp	object[]	绑定的ip地址
data.ipmacBind.bindIp.start	string	ip范围的起始位置 格式:ip-address
data.ipmacBind.bindIp.end	string	ip范围的终止位置 格式:ip-address
data.ipmacBind.bindIp.bits	uint8	ip掩码简写
data.ipmacBind.bindIp.mask	string	ip掩码 格式:ip-address
data.ipmacBind.bindMac	string[]	绑定的mac地址
data.ipmacBind.bindIpMac	object[]	绑定的ip和mac地址列表
data.ipmacBind.bindIpMac.ip	string	IP地址信息
data.ipmacBind.bindIpMac.mac	string	MAC地址信息
data.multipleLogin	object	可选, 多人同时登录设置
data.multipleLogin.enable	bool	是否开启多人同时登录
data.multipleLogin.loginNumber	uint8	可以同时登录的人数 最小值:1 最大值:5
data.logoutDisply	bool	可选, 登录成功后是否弹出注销窗口
data.autoLogout	object	可选, 设置无流量自动注销时间
data.autoLogout.enable	bool	是否开启注销时间限制
data.autoLogout.logoutTime	uint32	多长时间无流量自动注销, 分钟为单位 最小值:1 最大值:65535
data.expireTime	object	可选, 设置账号过期时间
data.expireTime.status	string	账号过期的类型选择 枚举值:[NEVER:永不过期 EXPIRE:需要设置过期的日期]
data.expireTime.expireDate	string	过期的日期设置

请求成功样例:

```
{
  "data": {
```

```
{
  "name": "test"
},
{
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.3.1.1.5 增量更新一个认证用户

功能说明: 增量更新一个认证用户

1) API请求数据

PATCH <https://{host}/api/v1/namespaces/@namespace/authusers/@name>

2) 输入参数

• URL参数

参数	类型	是否必选	描述
_arrayop	string	否	对象属性是数组的操作控制 枚举值: add:添加对应的数组元素 remove:删除对应的数组元素]

• Body参数

参数	类型	是否必选	描述
obj	object	是	认证用户信息
obj.name	string	是	用户名称 最小长度:1 最大长度:95
obj.displayName	string	否	用户的显示名称 最小长度:0 最大长度:95
obj.path	string	否	用户路径 最小长度:1 最大长度:1440
obj.description	string	否	用户描述 最小长度:0 最大长度:95
obj.userEnable	bool	否	用户启禁用状态 默认值:true
obj.localPasswd	object	否	可选, 本地密码设置
obj.localPasswd.enable	bool	否	是否启用本地密码设置
obj.localPasswd.password	string	是	本地密码 最小长度:1 最大长度:40

obj.ipmacBind	object	否	可选，地址绑定设置
obj.ipmacBind.enable	bool	否	是否启用地址绑定
obj.ipmacBind.bindType	string	否	绑定方式,分为单向和双向 默认值:true 枚举值:[DUPLEX:地址和用户双向绑定 SIMPLEX:地址和用户单向绑定]
obj.ipmacBind.bindAddr	string	否	绑定的地址选择 默认值:true 枚举值:[IP:绑定IP地址，MAC地址不绑定 MAC:绑定MAC地址，IP地址不绑定 IPMAC:绑定IP和MAC地址]
obj.ipmacBind.bindIp	object[]	是	绑定的ip地址
obj.ipmacBind.bindIp.start	string	是	ip范围的起始位置 格式:ip-address
obj.ipmacBind.bindIp.end	string	否	ip范围的终止位置 格式:ip-address
obj.ipmacBind.bindIp.bits	uint8	否	ip掩码简写
obj.ipmacBind.bindIp.mask	string	否	ip掩码 格式:ip-address
obj.ipmacBind.bindMac	string[]	是	绑定的mac地址
obj.ipmacBind.bindIpMac	object[]	是	绑定的ip和mac地址列表
obj.ipmacBind.bindIpMac.ip	string	是	IP地址信息
obj.ipmacBind.bindIpMac.mac	string	是	MAC地址信息
obj.multipleLogin	object	否	可选，多人同时登录设置
obj.multipleLogin.enable	bool	否	是否开启多人同时登录
obj.multipleLogin.loginNumber	uint8	否	可以同时登录的人数 最小值:1 最大值:5 默认值:true
obj.logoutDisply	bool	否	可选，登录成功后是否弹出注销窗口
obj.autoLogout	object	否	可选，设置无流量自动注销时间
obj.autoLogout.enable	bool	否	是否开启注销时间限制 默认值:true
obj.autoLogout.logoutTime	uint32	否	多长时间无流量自动注销，分钟为单位 最小值:1 最大值:65535 默认值:true
obj.expireTime	object	否	可选，设置账号过期时间

obj.expireTime.status	string	否	账号过期的类型选择 默认值:true 枚举值:[NEVER:永不过期 EXPIRE:需要设置过期的日期]
obj.expireTime.expireDate	string	是	过期的日期设置

请求样例:

增量更新一个认证用户test,修改内容: 绑定地址改为双向绑定, 绑定IP地址为192.168.11.11, 多人登录选项置为打开, 允许人数为5, 其他字段不修改

```
PATCH https://192.168.1.1/api/v1/namespaces/public/authusers/test
{
  "ipmacBind": {
    "enable": true,
    "bindAddr": "IP",
    "bindIp": [
      {
        "start": "192.168.11.11",
        "end": "192.168.11.11"
      }
    ],
    "bindType": "DUPLEX"
  },
  "name": "test",
  "multipleLogin": {
    "enable": true,
    "loginNumber": 5
  }
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	认证用户信息
data.name	string	用户名称 最小长度:1 最大长度:95
data.uuid	string	用户uuid
data.displayName	string	用户的显示名称 最小长度:0 最大长度:95
data.path	string	用户路径 最小长度:1 最大长度:1440
data.description	string	用户描述 最小长度:0 最大长度:95
data.userEnable	bool	用户启禁用状态
data.localPasswd	object	可选, 本地密码设置

data.localPasswd.enable	bool	是否启用本地密码设置
data.localPasswd.password	string	本地密码 最小长度:1 最大长度:40
data.ipmacBind	object	可选，地址绑定设置
data.ipmacBind.enable	bool	是否启用地址绑定
data.ipmacBind.bindType	string	绑定方式,分为单向和双向 枚举值:[DUPLEX:地址和用户双向绑定 SIMPLEX:地址和用户单向绑定]
data.ipmacBind.bindAddr	string	绑定的地址选择 枚举值:[IP:绑定IP地址，MAC地址不绑定 MAC:绑定MAC地址，IP地址不绑定 IPMAC:绑定IP和MAC地址]
data.ipmacBind.bindIp	object[]	绑定的ip地址
data.ipmacBind.bindIp.start	string	ip范围的起始位置 格式:ip-address
data.ipmacBind.bindIp.end	string	ip范围的终止位置 格式:ip-address
data.ipmacBind.bindIp.bits	uint8	ip掩码简写
data.ipmacBind.bindIp.mask	string	ip掩码 格式:ip-address
data.ipmacBind.bindMac	string[]	绑定的mac地址
data.ipmacBind.bindIpMac	object[]	绑定的ip和mac地址列表
data.ipmacBind.bindIpMac.ip	string	IP地址信息
data.ipmacBind.bindIpMac.mac	string	MAC地址信息
data.multipleLogin	object	可选，多人同时登录设置
data.multipleLogin.enable	bool	是否开启多人同时登录
data.multipleLogin.loginNumber	uint8	可以同时登录的人数 最小值:1 最大值:5
data.logoutDisply	bool	可选，登录成功后是否弹出注销窗口
data.autoLogout	object	可选，设置无流量自动注销时间
data.autoLogout.enable	bool	是否开启注销时间限制

data.autoLogout.logoutTime	uint32	多长时间无流量自动注销，分钟为单位 最小值:1 最大值:65535
data.expireTime	object	可选，设置账号过期时间
data.expireTime.status	string	账号过期的类型选择 枚举值: NEVER:永不过期 EXPIRE:需要设置过期的日期]
data.expireTime.expireDate	string	过期的日期设置

请求成功样例：

```
{
  "data": {
    "ipmacBind": {
      "enable": true,
      "bindAddr": "IP",
      "bindIp": [
        {
          "start": "192.168.11.11",
          "end": "192.168.11.11"
        }
      ],
      "bindType": "DUPLEX"
    },
    "name": "test",
    "multipleLogin": {
      "enable": true,
      "loginNumber": 5
    }
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.3.1.1.6 全量更新一个认证用户

功能说明：全量更新一个认证用户

1) API请求数据

PUT <https://{host}/api/v1/namespaces/@namespace/authusers/@name>

2) 输入参数

• Body参数

参数	类型	是否必选	描述
obj	object	是	认证用户信息
obj.name	string	是	用户名称 最小长度:1 最大长度:95

obj.displayName	string	否	用户的显示名称 最小长度:0 最大长度:95
obj.path	string	否	用户路径 最小长度:1 最大长度:1440
obj.description	string	否	用户描述 最小长度:0 最大长度:95
obj.userEnable	bool	否	用户启禁用状态 默认值:true
obj.localPasswd	object	否	可选, 本地密码设置
obj.localPasswd.enable	bool	否	是否启用本地密码设置
obj.localPasswd.password	string	是	本地密码 最小长度:1 最大长度:40
obj.ipmacBind	object	否	可选, 地址绑定设置
obj.ipmacBind.enable	bool	否	是否启用地址绑定
obj.ipmacBind.bindType	string	否	绑定方式,分为单向和双向 默认值:true 枚举值:[DUPLEX:地址和用户双向绑定 SIMPLEX:地址和用户单向绑定]
obj.ipmacBind.bindAddr	string	否	绑定的地址选择 默认值:true 枚举值:[IP:绑定IP地址, MAC地址不绑定 MAC:绑定MAC地址, IP地址不绑定 IPMAC:绑定IP和MAC地址]
obj.ipmacBind.bindIp	object[]	是	绑定的ip地址
obj.ipmacBind.bindIp.start	string	是	ip范围的起始位置 格式:ip-address
obj.ipmacBind.bindIp.end	string	否	ip范围的终止位置 格式:ip-address
obj.ipmacBind.bindIp.bits	uint8	否	ip掩码简写
obj.ipmacBind.bindIp.mask	string	否	ip掩码 格式:ip-address
obj.ipmacBind.bindMac	string[]	是	绑定的mac地址
obj.ipmacBind.bindIpMac	object[]	是	绑定的ip和mac地址列表
obj.ipmacBind.bindIpMac.ip	string	是	IP地址信息
obj.ipmacBind.bindIpMac.mac	string	是	MAC地址信息

obj.multipleLogin	object	否	可选，多人同时登录设置
obj.multipleLogin.enable	bool	否	是否开启多人同时登录
obj.multipleLogin.loginNumber	uint8	否	可以同时登录的人数 最小值:1 最大值:5 默认值:true
obj.logoutDisply	bool	否	可选，登录成功后是否弹出注销窗口
obj.autoLogout	object	否	可选，设置无流量自动注销时间
obj.autoLogout.enable	bool	否	是否开启注销时间限制 默认值:true
obj.autoLogout.logoutTime	uint32	否	多长时间无流量自动注销，分钟为单位 最小值:1 最大值:65535 默认值:true
obj.expireTime	object	否	可选，设置账号过期时间
obj.expireTime.status	string	否	账号过期的类型选择 默认值:true 枚举值:[NEVER:永不过期 EXPIRE:需要设置过期的日期]
obj.expireTime.expireDate	string	是	过期的日期设置

请求样例:

全量更新一个认证用户test,修改内容: 绑定地址改为双向绑定, 绑定MAC地址0a-0b-0c-0d-0e-1f, 过期时间设置为2021-03-27 00:00:00, 没有传字段用默认值填充

```
PUT https://192.168.1.1/api/v1/namespaces/public/authusers/test
{
  "expireTime": {
    "status": "EXPIRE",
    "expireDate": "2021-03-27 00:00:00"
  },
  "ipmacBind": {
    "enable": true,
    "bindMac": [
      "0a-0b-0c-0d-0e-1f"
    ],
    "bindAddr": "MAC",
    "bindType": "DUPLEX"
  },
  "name": "t2"
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	认证用户信息

data.name	string	用户名称 最小长度:1 最大长度:95
data.uuid	string	用户uuid
data.displayName	string	用户的显示名称 最小长度:0 最大长度:95
data.path	string	用户路径 最小长度:1 最大长度:1440
data.description	string	用户描述 最小长度:0 最大长度:95
data.userEnable	bool	用户启禁用状态
data.localPasswd	object	可选，本地密码设置
data.localPasswd.enable	bool	是否启用本地密码设置
data.localPasswd.password	string	本地密码 最小长度:1 最大长度:40
data.ipmacBind	object	可选，地址绑定设置
data.ipmacBind.enable	bool	是否启用地址绑定
data.ipmacBind.bindType	string	绑定方式,分为单向和双向 枚举值:[DUPLEX:地址和用户双向绑定 SIMPLEX:地址和用户单向绑定]
data.ipmacBind.bindAddr	string	绑定的地址选择 枚举值:[IP:绑定IP地址，MAC地址不绑定 MAC:绑定MAC地址，IP地址不绑定 IPMAC:绑定IP和MAC地址]
data.ipmacBind.bindIp	object[]	绑定的ip地址
data.ipmacBind.bindIp.start	string	ip范围的起始位置 格式:ip-address
data.ipmacBind.bindIp.end	string	ip范围的终止位置 格式:ip-address
data.ipmacBind.bindIp.bits	uint8	ip掩码简写
data.ipmacBind.bindIp.mask	string	ip掩码 格式:ip-address
data.ipmacBind.bindMac	string[]	绑定的mac地址
data.ipmacBind.bindIpMac	object[]	绑定的ip和mac地址列表

data.ipmacBind.bindIpMac.ip	string	IP地址信息
data.ipmacBind.bindIpMac.mac	string	MAC地址信息
data.multipleLogin	object	可选，多人同时登录设置
data.multipleLogin.enable	bool	是否开启多人同时登录
data.multipleLogin.loginNumber	uint8	可以同时登录的人数 最小值:1 最大值:5
data.logoutDisply	bool	可选，登录成功后是否弹出注销窗口
data.autoLogout	object	可选，设置无流量自动注销时间
data.autoLogout.enable	bool	是否开启注销时间限制
data.autoLogout.logoutTime	uint32	多长时间无流量自动注销，分钟为单位 最小值:1 最大值:65535
data.expireTime	object	可选，设置账号过期时间
data.expireTime.status	string	账号过期的类型选择 枚举值: NEVER:永不过期 EXPIRE:需要设置过期的日期]
data.expireTime.expireDate	string	过期的日期设置

请求成功样例：

```
{
  "data": {
    "expireTime": {
      "status": "EXPIRE",
      "expireDate": "2021-03-27 00:00:00"
    },
    "ipmacBind": {
      "enable": true,
      "bindMac": [
        "0a-0b-0c-0d-0e-1f"
      ],
      "bindAddr": "MAC",
      "bindType": "DUPLEX"
    },
    "name": "t2"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.3.1.1.7 批量删除认证用户

功能说明：批量删除认证用户

1) API请求数据

POST

https://{host}/api/batch/v1/namespaces/@namespace/authusers?_method=delete

2) 输入参数

• Body参数

参数	类型	是否必选	描述
obj	object[]	是	认证用户列表
obj.name	string	是	用户名称 最小长度:1 最大长度:95
obj.displayName	string	否	用户的显示名称 最小长度:0 最大长度:95
obj.path	string	否	用户路径 最小长度:1 最大长度:1440
obj.description	string	否	用户描述 最小长度:0 最大长度:95
obj.userEnable	bool	否	用户启禁用状态 默认值:true
obj.localPasswd	object	否	可选，本地密码设置
obj.localPasswd.enable	bool	否	是否启用本地密码设置
obj.localPasswd.password	string	是	本地密码 最小长度:1 最大长度:40
obj.ipmacBind	object	否	可选，地址绑定设置
obj.ipmacBind.enable	bool	否	是否启用地址绑定
obj.ipmacBind.bindType	string	否	绑定方式,分为单向和双向 默认值:true 枚举值:[DUPLEX:地址和用户双向绑定 SIMPLEX:地址和用户单向绑定]
obj.ipmacBind.bindAddr	string	否	绑定的地址选择 默认值:true 枚举值:[IP:绑定IP地址，MAC地址不绑定 MAC:绑定MAC地址，IP地址不绑定 IPMAC:绑定IP和MAC地址]
obj.ipmacBind.bindIp	object[]	是	绑定的ip地址

obj.ipmacBind.bindIp.start	string	是	ip范围的起始位置 格式:ip-address
obj.ipmacBind.bindIp.end	string	否	ip范围的终止位置 格式:ip-address
obj.ipmacBind.bindIp.bits	uint8	否	ip掩码简写
obj.ipmacBind.bindIp.mask	string	否	ip掩码 格式:ip-address
obj.ipmacBind.bindMac	string[]	是	绑定的mac地址
obj.ipmacBind.bindIpMac	object[]	是	绑定的ip和mac地址列表
obj.ipmacBind.bindIpMac.ip	string	是	IP地址信息
obj.ipmacBind.bindIpMac.mac	string	是	MAC地址信息
obj.multipleLogin	object	否	可选，多人同时登录设置
obj.multipleLogin.enable	bool	否	是否开启多人同时登录
obj.multipleLogin.loginNumber	uint8	否	可以同时登录的人数 最小值:1 最大值:5 默认值:true
obj.logoutDisply	bool	否	可选，登录成功后是否弹出注销窗口
obj.autoLogout	object	否	可选，设置无流量自动注销时间
obj.autoLogout.enable	bool	否	是否开启注销时间限制 默认值:true
obj.autoLogout.logoutTime	uint32	否	多长时间无流量自动注销，分钟为单位 最小值:1 最大值:65535 默认值:true
obj.expireTime	object	否	可选，设置账号过期时间
obj.expireTime.status	string	否	账号过期的类型选择 默认值:true 枚举值:[NEVER:永不过期 EXPIRE:需要设置过期的日期]
obj.expireTime.expireDate	string	是	过期的日期设置

请求样例:

批量删除认证用户

```
POST https://192.168.1.1/api/batch/v1/namespaces/public/authusers?_method=DELETE
[
  {
    "name": "test1"
  },
  {
    "name": "test2"
  },
]
```

```
{
  "name": "test3"
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	认证用户列表
data.name	string	用户名称 最小长度:1 最大长度:95
data.uuid	string	用户uuid
data.displayName	string	用户的显示名称 最小长度:0 最大长度:95
data.path	string	用户路径 最小长度:1 最大长度:1440
data.description	string	用户描述 最小长度:0 最大长度:95
data.userEnable	bool	用户启禁用状态
data.localPasswd	object	可选，本地密码设置
data.localPasswd.enable	bool	是否启用本地密码设置
data.localPasswd.password	string	本地密码 最小长度:1 最大长度:40
data.ipmacBind	object	可选，地址绑定设置
data.ipmacBind.enable	bool	是否启用地址绑定
data.ipmacBind.bindType	string	绑定方式,分为单向和双向 枚举值:[DUPLEX:地址和用户双向绑定 SIMPLEX:地址和用户单向绑定]
data.ipmacBind.bindAddr	string	绑定的地址选择 枚举值:[IP:绑定IP地址，MAC地址不绑定 MAC:绑定MAC地址，IP地址不绑定 IPMAC:绑定IP和MAC地址]
data.ipmacBind.bindIp	object[]	绑定的ip地址
data.ipmacBind.bindIp.start	string	ip范围的起始位置 格式:ip-address

data.ipmacBind.bindIp.end	string	ip范围的终止位置 格式:ip-address
data.ipmacBind.bindIp.bits	uint8	ip掩码简写
data.ipmacBind.bindIp.mask	string	ip掩码 格式:ip-address
data.ipmacBind.bindMac	string[]	绑定的mac地址
data.ipmacBind.bindIpMac	object[]	绑定的ip和mac地址列表
data.ipmacBind.bindIpMac.ip	string	IP地址信息
data.ipmacBind.bindIpMac.mac	string	MAC地址信息
data.multipleLogin	object	可选，多人同时登录设置
data.multipleLogin.enable	bool	是否开启多人同时登录
data.multipleLogin.loginNumber	uint8	可以同时登录的人数 最小值:1 最大值:5
data.logoutDisply	bool	可选，登录成功后是否弹出注销窗口
data.autoLogout	object	可选，设置无流量自动注销时间
data.autoLogout.enable	bool	是否开启注销时间限制
data.autoLogout.logoutTime	uint32	多长时间无流量自动注销，分钟为单位 最小值:1 最大值:65535
data.expireTime	object	可选，设置账号过期时间
data.expireTime.status	string	账号过期的类型选择 枚举值: NEVER:永不过期 EXPIRE:需要设置过期的日期]
data.expireTime.expireDate	string	过期的日期设置

请求成功样例：

```
{
  "data": [
    {
      "name": "test1"
    },
    {
      "name": "test1"
    },
    {
      "name": "test1"
    }
  ],
  "code": 0,
  "message": "成功"
}
```


请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.3.1.1.8 批量新增认证用户

功能说明：批量新增认证用户

1) API请求数据

POST <https://{host}/api/batch/v1/namespaces/@namespace/authusers>

2) 输入参数

• Body参数

参数	类型	是否必选	描述
obj	object[]	是	认证用户列表
obj.name	string	是	用户名称 最小长度:1 最大长度:95
obj.displayName	string	否	用户的显示名称 最小长度:0 最大长度:95
obj.path	string	否	用户路径 最小长度:1 最大长度:1440
obj.description	string	否	用户描述 最小长度:0 最大长度:95
obj.userEnable	bool	否	用户启禁用状态 默认值:true
obj.localPasswd	object	否	可选，本地密码设置
obj.localPasswd.enable	bool	否	是否启用本地密码设置
obj.localPasswd.password	string	是	本地密码 最小长度:1 最大长度:40
obj.ipmacBind	object	否	可选，地址绑定设置
obj.ipmacBind.enable	bool	否	是否启用地址绑定
obj.ipmacBind.bindType	string	否	绑定方式,分为单向和双向 默认值:true 枚举值:[DUPLEX:地址和用户双向绑定 SIMPLEX:地址和用户单向绑定]

obj.ipmacBind.bindAddr	string	否	绑定的地址选择 默认值:true 枚举值:[IP:绑定IP地址, MAC地址不绑定 MAC:绑定MAC地址, IP地址不绑定 IPMAC:绑定IP和MAC地址]
obj.ipmacBind.bindIp	object[]	是	绑定的ip地址
obj.ipmacBind.bindIp.start	string	是	ip范围的起始位置 格式:ip-address
obj.ipmacBind.bindIp.end	string	否	ip范围的终止位置 格式:ip-address
obj.ipmacBind.bindIp.bits	uint8	否	ip掩码简写
obj.ipmacBind.bindIp.mask	string	否	ip掩码 格式:ip-address
obj.ipmacBind.bindMac	string[]	是	绑定的mac地址
obj.ipmacBind.bindIpMac	object[]	是	绑定的ip和mac地址列表
obj.ipmacBind.bindIpMac.ip	string	是	IP地址信息
obj.ipmacBind.bindIpMac.mac	string	是	MAC地址信息
obj.multipleLogin	object	否	可选, 多人同时登录设置
obj.multipleLogin.enable	bool	否	是否开启多人同时登录
obj.multipleLogin.loginNumber	uint8	否	可以同时登录的人数 最小值:1 最大值:5 默认值:true
obj.logoutDisply	bool	否	可选, 登录成功后是否弹出注销窗口
obj.autoLogout	object	否	可选, 设置无流量自动注销时间
obj.autoLogout.enable	bool	否	是否开启注销时间限制 默认值:true
obj.autoLogout.logoutTime	uint32	否	多长时间无流量自动注销, 分钟为单位 最小值:1 最大值:65535 默认值:true
obj.expireTime	object	否	可选, 设置账号过期时间
obj.expireTime.status	string	否	账号过期的类型选择 默认值:true 枚举值:[NEVER:永不过期 EXPIRE:需要设置过期的日期]
obj.expireTime.expireDate	string	是	过期的日期设置

请求样例:

```
批量新增认证用户,都新增在根组下
POST https://192.168.1.1/api/batch/v1/namespaces/public/authusers
[
  {
    "name": "test1",
    "path": "/"
  },
  {
    "name": "test2",
    "path": "/"
  },
  {
    "name": "test3",
    "path": "/"
  }
]
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	认证用户列表
data.name	string	用户名称 最小长度:1 最大长度:95
data.uuid	string	用户uuid
data.displayName	string	用户的显示名称 最小长度:0 最大长度:95
data.path	string	用户路径 最小长度:1 最大长度:1440
data.description	string	用户描述 最小长度:0 最大长度:95
data.userEnable	bool	用户启禁用状态
data.localPasswd	object	可选，本地密码设置
data.localPasswd.enable	bool	是否启用本地密码设置
data.localPasswd.password	string	本地密码 最小长度:1 最大长度:40
data.ipmacBind	object	可选，地址绑定设置
data.ipmacBind.enable	bool	是否启用地址绑定

data.ipmacBind.bindType	string	绑定方式,分为单向和双向 枚举值:[DUPLEX:地址和用户双向绑定 SIMPLEX:地址和用户单向绑定]
data.ipmacBind.bindAddr	string	绑定的地址选择 枚举值:[IP:绑定IP地址, MAC地址不绑定 MAC:绑定MAC地址, IP地址不绑定 IPMAC:绑定IP和MAC地址]
data.ipmacBind.bindIp	object[]	绑定的ip地址
data.ipmacBind.bindIp.start	string	ip范围的起始位置 格式:ip-address
data.ipmacBind.bindIp.end	string	ip范围的终止位置 格式:ip-address
data.ipmacBind.bindIp.bits	uint8	ip掩码简写
data.ipmacBind.bindIp.mask	string	ip掩码 格式:ip-address
data.ipmacBind.bindMac	string[]	绑定的mac地址
data.ipmacBind.bindIpMac	object[]	绑定的ip和mac地址列表
data.ipmacBind.bindIpMac.ip	string	IP地址信息
data.ipmacBind.bindIpMac.mac	string	MAC地址信息
data.multipleLogin	object	可选, 多人同时登录设置
data.multipleLogin.enable	bool	是否开启多人同时登录
data.multipleLogin.loginNumber	uint8	可以同时登录的人数 最小值:1 最大值:5
data.logoutDisply	bool	可选, 登录成功后是否弹出注销窗口
data.autoLogout	object	可选, 设置无流量自动注销时间
data.autoLogout.enable	bool	是否开启注销时间限制
data.autoLogout.logoutTime	uint32	多长时间无流量自动注销, 分钟为单位 最小值:1 最大值:65535
data.expireTime	object	可选, 设置账号过期时间
data.expireTime.status	string	账号过期的类型选择 枚举值:[NEVER:永不过期 EXPIRE:需要设置过期的日期]

data.expireTime.expire Date	string	过期的日期设置
-----------------------------	--------	---------

请求成功样例:

```
{
  "data": [
    {
      "name": "test1",
      "path": "/"
    },
    {
      "name": "test2",
      "path": "/"
    },
    {
      "name": "test3",
      "path": "/"
    }
  ],
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.3.1.1.9 批量更新认证用户

功能说明: 批量更新认证用户

1) API请求数据

PATCH https://{host}/api/batch/v1/namespaces/@namespace/authusers

2) 输入参数

• Body参数

参数	类型	是否必选	描述
obj	object[]	是	认证用户列表
obj.name	string	是	用户名称 最小长度:1 最大长度:95
obj.displayName	string	否	用户的显示名称 最小长度:0 最大长度:95
obj.path	string	否	用户路径 最小长度:1 最大长度:1440
obj.description	string	否	用户描述 最小长度:0 最大长度:95
obj.userEnable	bool	否	用户启禁用状态 默认值:true
obj.localPasswd	object	否	可选, 本地密码设置

obj.localPasswd.enable	bool	否	是否启用本地密码设置
obj.localPasswd.password	string	是	本地密码 最小长度:1 最大长度:40
obj.ipmacBind	object	否	可选，地址绑定设置
obj.ipmacBind.enable	bool	否	是否启用地址绑定
obj.ipmacBind.bindType	string	否	绑定方式,分为单向和双向 默认值:true 枚举值:[DUPLEX:地址和用户双向绑定 SIMPLEX:地址和用户单向绑定]
obj.ipmacBind.bindAddr	string	否	绑定的地址选择 默认值:true 枚举值:[IP:绑定IP地址，MAC地址不绑定 MAC:绑定MAC地址，IP地址不绑定 IPMAC:绑定IP和MAC地址]
obj.ipmacBind.bindIp	object[]	是	绑定的ip地址
obj.ipmacBind.bindIp.start	string	是	ip范围的起始位置 格式:ip-address
obj.ipmacBind.bindIp.end	string	否	ip范围的终止位置 格式:ip-address
obj.ipmacBind.bindIp.bits	uint8	否	ip掩码简写
obj.ipmacBind.bindIp.mask	string	否	ip掩码 格式:ip-address
obj.ipmacBind.bindMac	string[]	是	绑定的mac地址
obj.ipmacBind.bindIpMac	object[]	是	绑定的ip和mac地址列表
obj.ipmacBind.bindIpMac.ip	string	是	IP地址信息
obj.ipmacBind.bindIpMac.mac	string	是	MAC地址信息
obj.multipleLogin	object	否	可选，多人同时登录设置
obj.multipleLogin.enable	bool	否	是否开启多人同时登录
obj.multipleLogin.loginNumber	uint8	否	可以同时登录的人数 最小值:1 最大值:5 默认值:true
obj.logoutDisply	bool	否	可选，登录成功后是否弹出注销窗口
obj.autoLogout	object	否	可选，设置无流量自动注销时间
obj.autoLogout.enable	bool	否	是否开启注销时间限制 默认值:true

obj.autoLogout.logoutTime	uint32	否	多长时间无流量自动注销，分钟为单位 最小值:1 最大值:65535 默认值:true
obj.expireTime	object	否	可选，设置账号过期时间
obj.expireTime.status	string	否	账号过期的类型选择 默认值:true 枚举值:[NEVER:永不过期 EXPIRE:需要设置过期的日期]
obj.expireTime.expireDate	string	是	过期的日期设置

请求样例:

批量更新认证用户,只修改描述字段

PATCH https://192.168.1.1/api/batch/v1/namespaces/public/authusers

```
[
  {
    "name": "test1",
    "description": "111"
  },
  {
    "name": "test2",
    "description": "222"
  },
  {
    "name": "test3",
    "description": "333"
  }
]
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	认证用户列表
data.name	string	用户名称 最小长度:1 最大长度:95
data.uuid	string	用户uuid
data.displayName	string	用户的显示名称 最小长度:0 最大长度:95
data.path	string	用户路径 最小长度:1 最大长度:1440
data.description	string	用户描述 最小长度:0 最大长度:95
data.userEnable	bool	用户启禁用状态

data.localPasswd	object	可选，本地密码设置
data.localPasswd.enable	bool	是否启用本地密码设置
data.localPasswd.password	string	本地密码 最小长度:1 最大长度:40
data.ipmacBind	object	可选，地址绑定设置
data.ipmacBind.enable	bool	是否启用地址绑定
data.ipmacBind.bindType	string	绑定方式,分为单向和双向 枚举值:[DUPLEX:地址和用户双向绑定 SIMPLEX:地址和用户单向绑定]
data.ipmacBind.bindAddr	string	绑定的地址选择 枚举值:[IP:绑定IP地址，MAC地址不绑定 MAC:绑定MAC地址，IP地址不绑定 IPMAC:绑定IP和MAC地址]
data.ipmacBind.bindIp	object[]	绑定的ip地址
data.ipmacBind.bindIp.start	string	ip范围的起始位置 格式:ip-address
data.ipmacBind.bindIp.end	string	ip范围的终止位置 格式:ip-address
data.ipmacBind.bindIp.bits	uint8	ip掩码简写
data.ipmacBind.bindIp.mask	string	ip掩码 格式:ip-address
data.ipmacBind.bindMac	string[]	绑定的mac地址
data.ipmacBind.bindIpMac	object[]	绑定的ip和mac地址列表
data.ipmacBind.bindIpMac.ip	string	IP地址信息
data.ipmacBind.bindIpMac.mac	string	MAC地址信息
data.multipleLogin	object	可选，多人同时登录设置
data.multipleLogin.enable	bool	是否开启多人同时登录
data.multipleLogin.loginNumber	uint8	可以同时登录的人数 最小值:1 最大值:5
data.logoutDisply	bool	可选，登录成功后是否弹出注销窗口
data.autoLogout	object	可选，设置无流量自动注销时间
data.autoLogout.enable	bool	是否开启注销时间限制

data.autoLogout.logoutTime	uint32	多长时间无流量自动注销，分钟为单位 最小值:1 最大值:65535
data.expireTime	object	可选，设置账号过期时间
data.expireTime.status	string	账号过期的类型选择 枚举值: NEVER:永不过期 EXPIRE:需要设置过期的日期]
data.expireTime.expireDate	string	过期的日期设置

请求成功样例:

```
{
  "data": [
    {
      "name": "test1",
      "description": "111"
    },
    {
      "name": "test2",
      "description": "222"
    },
    {
      "name": "test3",
      "description": "333"
    }
  ],
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.3.1.2 认证用户组

4.3.1.2.1 获取指定认证用户组的信息

功能说明: 获取指定组的基本信息,输入时要进行编码,/test写为 '%2ftest'

1) API请求数据

GET https://{host}/api/v1/namespaces/@namespace/authorginfo/@pathname

2) 输入参数

请求样例:

获取认证用户组'/test'的基本信息,编码为'%2ftest'

GET https://192.168.1.1/api/v1/namespaces/public/authorginfo/%2ftest

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码

message	string	错误信息
data	object	返回指定组的详细信息
data.authOrg	object	用户组基本信息
data.authOrg.name	string	用户组名称 最小长度:1 最大长度:95
data.authOrg.uuid	string	用户组的uuid
data.authOrg.path	string	用户组路径 最小长度:1 最大长度:1440
data.authOrg.description	string	用户组描述 最小长度:0 最大长度:95
data.authOrg.leaf	bool	是否是叶子节点，即是否为最里层，不可配置项，仅用于树形结构的展示
data.orgs	string[]	组名列表
data.users	string[]	用户名列表
data.totalUsers	int32	该组的总用户数

请求成功样例：

```
{
  "data": {
    "authOrg": {
      "path": "/test",
      "name": "test",
      "description": ""
    },
    "totalUsers": 0
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.3.1.2.2 获取用户组列表

功能说明：获取用户组列表

1) API请求数据

GET

https://{host}/api/v1/namespaces/@namespace/authorgs

2) 输入参数

- URL 参数

参数	类型	是否必选	描述
_start	int32	否	起始位置, 从 0 开始 最小值:0 最大值:2000000

_length	int32	否	获取最大长度，限制最大值，如果数据不够可能返回小于 length 最小值:1 最大值:200 默认值:true
---------	-------	---	--

请求样例:

```
获取用户组列表,结果展示前100条

GET https://192.168.1.1/api/v1/namespaces/public/authorgs?_start=0&_length=100
[]
```

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	分页数据
data.totalItems	int32	总共多少项目
data.totalPages	int32	总共多少页
data.pageNumber	int32	当前页码，从 1 开始
data.pageSize	int32	每页多大
data.itemsOffset	int32	当前条目偏移, 从 0 开始
data.itemLength	int32	数据列表长度
data.items	object[]	有效数据列表
data.items.name	string	用户组名称 最小长度:1 最大长度:95
data.items.uuid	string	用户组的uuid
data.items.path	string	用户组路径 最小长度:1 最大长度:1440
data.items.description	string	用户组描述 最小长度:0 最大长度:95
data.items.leaf	bool	是否是叶子节点，即是否为最里层，不可配置项，仅用于树形结构的展示

请求成功样例:

```
{
  "data": [
    {
      "path": "/",
      "leaf": false,
      "name": "/",
      "description": ""
    },
    {
      "path": "/t1",
      "leaf": false,
      "name": "t1",

```

```
        "description": ""
      },
      {
        "path": "/t1/",
        "leaf": true,
        "name": "test",
        "description": ""
      },
      {
        "path": "/",
        "leaf": true,
        "name": "默认组",
        "description": ""
      }
    ],
    "code": 0,
    "message": "成功"
  }
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.3.1.2.3 新增一个认证用户组

功能说明：新增一个认证用户组

1) API请求数据

POST

https://{host}/api/v1/namespaces/@namespace/authorgs

2) 输入参数

- Body参数

参数	类型	是否必选	描述
obj	object	是	用户组
obj.name	string	是	用户组名称 最小长度:1 最大长度:95
obj.path	string	是	用户组路径 最小长度:1 最大长度:1440
obj.description	string	是	用户组描述 最小长度:0 最大长度:95

请求样例：

新增一个认证用户组test,路径为'/t1/'

```
POST https://192.168.1.1/api/v1/namespaces/public/authorgs
{
  "name": "test",
  "path": "/t1/"
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	用户组
data.name	string	用户组名称 最小长度:1 最大长度:95
data.uuid	string	用户组的uuid
data.path	string	用户组路径 最小长度:1 最大长度:1440
data.description	string	用户组描述 最小长度:0 最大长度:95
data.leaf	bool	是否是叶子节点，即是否为最里层，不可配置项，仅用于树形结构的展示

请求成功样例：

```
{
  "data": {
    "name": "test",
    "path": "/t1/"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.3.1.2.4 删除一个认证用户组

功能说明：删除一个认证用户组,输入时要进行编码,/test写为 '%2ftest'

1) API请求数据

DELETE

https://{host}/api/v1/namespaces/@namespace/authorgs/@pathname

2) 输入参数

请求样例：

删除一个认证用户组'/test',编码为'%2ftest'

DELETE https://192.168.1.1/api/v1/namespaces/public/authorgs/%2ftest

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	用户组
data.name	string	用户组名称 最小长度:1 最大长度:95
data.uuid	string	用户组的uuid
data.path	string	用户组路径 最小长度:1 最大长度:1440
data.description	string	用户组描述 最小长度:0 最大长度:95
data.leaf	bool	是否是叶子节点, 即是否为最里层, 不可配置项, 仅用于树形结构的展示

请求成功样例:

```
{
  "data": {
    "name": "test",
    "path": "/"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.3.1.2.5 全量更新一个认证用户组

功能说明: 全量更新一个认证用户组,输入时要进行编码,/test'写为 '%2ftest'

1) API请求数据

PUT https://{host}/api/v1/namespaces/@namespace/authorgs/@pathname

2) 输入参数

- Body参数

参数	类型	是否必选	描述
obj	object	是	用户组
obj.name	string	是	用户组名称 最小长度:1 最大长度:95

obj.path	string	是	用户组路径 最小长度:1 最大长度:1440
obj.description	string	是	用户组描述 最小长度:0 最大长度:95

请求样例:

```
    全量修改一个认证用户组'/test',修改组名为t4,其他字段用默认值填充
    PUT https://192.168.1.1/api/v1/namespaces/public/authorgs/%2ftest
    {
      "name": "t4"
    }
```

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	用户组
data.name	string	用户组名称 最小长度:1 最大长度:95
data.uuid	string	用户组的uuid
data.path	string	用户组路径 最小长度:1 最大长度:1440
data.description	string	用户组描述 最小长度:0 最大长度:95
data.leaf	bool	是否是叶子节点，即是否为最里层，不可配置项，仅用于树形结构的展示

请求成功样例:

```
{
  "data": {
    "name": "t4"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.3.1.2.6 增量更新一个认证用户组

功能说明：增量更新一个认证用户组,输入时要进行编码,/test'写为 '%2ftest'

1) API请求数据

PATCH

https://{host}/api/v1/namespaces/@namespace/authorgs/@pathname

2) 输入参数

- Body参数

参数	类型	是否必选	描述
obj	object	是	用户组
obj.name	string	是	用户组名称 最小长度:1 最大长度:95
obj.path	string	是	用户组路径 最小长度:1 最大长度:1440
obj.description	string	是	用户组描述 最小长度:0 最大长度:95

请求样例：

增量修改一个认证用户组'/test',只修改描述，其他字段不修改

```
PATCH https://192.168.1.1/api/v1/namespaces/public/authorgs/%2ftest
{
  "name": "test",
  "description": "111"
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	用户组
data.name	string	用户组名称 最小长度:1 最大长度:95
data.uuid	string	用户组的uuid
data.path	string	用户组路径 最小长度:1 最大长度:1440
data.description	string	用户组描述 最小长度:0 最大长度:95
data.leaf	bool	是否是叶子节点，即是否为最里层，不可配置项，仅用于树形结构的展示

请求成功样例:

```
{
  "data": {
    "name": "test",
    "description": "111"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.3.1.2.7 批量删除认证用户组

功能说明: 批量删除认证用户组

1) API请求数据

POST

https://{host}/api/batch/v1/namespaces/@namespace/authorgs?_method=delete

2) 输入参数

• Body参数

参数	类型	是否必选	描述
obj	object[]	是	用户组列表
obj.name	string	是	用户组名称 最小长度:1 最大长度:95
obj.path	string	是	用户组路径 最小长度:1 最大长度:1440
obj.description	string	是	用户组描述 最小长度:0 最大长度:95

请求样例:

批量删除认证用户组,路径为/

POST https://192.168.1.1/api/batch/v1/namespaces/public/authorgs?_method=DELETE

```
[
  {
    "name": "test1",
    "path": "/"
  },
  {
    "name": "test2",
    "path": "/"
  }
]
```

3) 输出参数

• API返回数据

参数	类型	描述
----	----	----

code	int	错误码
message	string	错误信息
data	object[]	用户组列表
data.name	string	用户组名称 最小长度:1 最大长度:95
data.uuid	string	用户组的uuid
data.path	string	用户组路径 最小长度:1 最大长度:1440
data.description	string	用户组描述 最小长度:0 最大长度:95
data.leaf	bool	是否是叶子节点，即是否为最里层，不可配置项，仅用于树形结构的展示

请求成功样例：

```
{
  "data": [
    {
      "name": "test1",
      "path": "/"
    },
    {
      "name": "test2",
      "path": "/"
    }
  ],
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.3.1.2.8 批量新增认证用户组

功能说明：批量新增认证用户组

1) API请求数据

POST

https://{host}/api/batch/v1/namespaces/@namespace/authorgs

2) 输入参数

• Body参数

参数	类型	是否必选	描述
obj	object[]	是	用户组列表
obj.name	string	是	用户组名称 最小长度:1 最大长度:95

obj.path	string	是	用户组路径 最小长度:1 最大长度:1440
obj.description	string	是	用户组描述 最小长度:0 最大长度:95

请求样例:

```
批量新增认证用户组，路径为/
POST https://192.168.1.1/api/batch/v1/namespaces/public/authorgs
[
  {
    "name": "test1",
    "path": "/"
  },
  {
    "name": "test2",
    "path": "/"
  }
]
```

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	用户组列表
data.name	string	用户组名称 最小长度:1 最大长度:95
data.uuid	string	用户组的uuid
data.path	string	用户组路径 最小长度:1 最大长度:1440
data.description	string	用户组描述 最小长度:0 最大长度:95
data.leaf	bool	是否是叶子节点，即是否为最里层，不可配置项，仅用于树形结构的展示

请求成功样例:

```
{
  "data": [
    {
      "name": "test1",
      "path": "/"
    },
    {
      "name": "test2",
      "path": "/"
    }
  ],
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.3.1.2.9 批量更新用户组

功能说明: 仅支持批量更新认证用户组的描述,不支持批量改名和移动

1) API请求数据

PATCH https://{host}/api/batch/v1/namespaces/@namespace/authorgs

2) 输入参数

- Body参数

参数	类型	是否必选	描述
obj	object[]	是	用户组列表
obj.name	string	是	用户组名称 最小长度:1 最大长度:95
obj.path	string	是	用户组路径 最小长度:1 最大长度:1440
obj.description	string	是	用户组描述 最小长度:0 最大长度:95

请求样例:

批量修改认证用户组,只修改描述字段

```
PATCH https://192.168.1.1/api/batch/v1/namespaces/public/authorgs
[
  {
    "name": "test1",
    "description": "111"
  },
  {
    "name": "test2",
    "description": "222"
  }
]
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	用户组列表
data.name	string	用户组名称 最小长度:1 最大长度:95
data.uuid	string	用户组的uuid

data.path	string	用户组路径 最小长度:1 最大长度:1440
data.description	string	用户组描述 最小长度:0 最大长度:95
data.leaf	bool	是否是叶子节点，即是否为最里层，不可配置项，仅用于树形结构的展示

请求成功样例：

```
{
  "data": [
    {
      "name": "test1",
      "description": "111"
    },
    {
      "name": "test2",
      "description": "222"
    }
  ],
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.3.2 用户认证

4.3.2.1 认证策略

4.3.2.1.1 获取全部认证策略

功能说明：获取全部认证策略

1) API请求数据

GET

https://{host}/api/v1/namespaces/@namespace/authpolicys

2) 输入参数

请求样例：

获取认证策略列表

GET https://192.168.1.1/api/v1/namespaces/public/authpolicys

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	认证策略列表

data.uuid	string	认证策略UUID
data.name	string	策略名称 最小长度:1 最大长度:95
data.description	string	策略描述 最小长度:0 最大长度:95
data.position	int32	策略的位置信息 最小值:0 最大值:64
data.IpRangeList	object	策略适用的ip和mac列表
data.IpRangeList.ip	object[]	适用的IP地址信息
data.IpRangeList.ip.start	string	ip范围的起始位置 格式:ip-address
data.IpRangeList.ip.end	string	ip范围的终止位置 格式:ip-address
data.IpRangeList.ip.bits	uint8	ip掩码简写
data.IpRangeList.ip.mask	string	ip掩码 格式:ip-address
data.IpRangeList.mac	string[]	适用的MAC地址信息
data.authType	string	可选, 认证方式 枚举值:[NOAUTH:不需要认证 LOCALAUTH:本地密码认证 SSOAUTH:SSO登录认证]
data.noAuth	string	认证方式选择不需要认证时, 需要设置用户名 枚举值:[IPTOUSER:ip作为用户名 MACTOUSER:mac作为用户名 PCTOUSER:计算机名作为用户名]
data.exceptUser	string[]	认证方式选择单点登录, 需要添加例外用户
data.newUserOption	string	新用户选项 枚举值:[LOCALUSER:添加的本地组, 作为本地用户 TEMPUSER:临时账号 FORBIDUSER:不允许新用户认证]
data.localUser	object	添加到本地组用户设置
data.localUser.localPathAdd	string	添加到本地组路径, 默认是/ 最小长度:1 最大长度:1440
data.localUser.IdapAutoAdd	bool	是否开启LDAP自动同步到组

data.localUser.multipleLogin	string	多人同时登录限制 枚举值:[PUBLIC:可以多人同时登录 PRIVATE:只可以一人使用]
data.localUser.bindType	string	可选，地址绑定方式 枚举值:[DUPLEX:地址和用户双向绑定 SIMPLEX:地址和用户单向绑定]
data.localUser.bindAddr	string	可选，绑定地址选择 枚举值:[IP:绑定IP地址，MAC地址不绑定 MAC:绑定MAC地址，IP地址不绑定 IPMAC:绑定IP和MAC地址]
data.tempRightPath	string	作为临时账号时，分配的权限组路径,默认是/ 最小长度:1 最大长度:1440

请求成功样例：

```
{
  "data": [
    {
      "localUser": {
        "ldapAutoAdd": false,
        "bindType": "SIMPLEX",
        "localPathAdd": "/",
        "multipleLogin": "PUBLIC",
        "bindAddr": "IP"
      },
      "position": 0,
      "newUserOption": "LOCALUSER",
      "authType": "LOCALAUTH",
      "name": "t1",
      "ipRangeList": {
        "ip": [
          {
            "start": "4.4.4.4",
            "end": "4.4.4.55"
          }
        ],
        "mac": []
      },
      "description": ""
    },
    {
      "localUser": {
        "ldapAutoAdd": false,
        "bindType": "SIMPLEX",
        "localPathAdd": "/",
        "multipleLogin": "PUBLIC",
        "bindAddr": "IP"
      },
      "position": 1,
      "newUserOption": "LOCALUSER",
      "authType": "LOCALAUTH",
      "name": "默认策略",
      "ipRangeList": {
        "ip": [
          {
            "start": "0.0.0.0",
            "end": "255.255.255.255"
          }
        ],
        "mac": []
      }
    }
  ]
}
```

```
        "description": "默认策略"
      },
    ],
    "code": 0,
    "message": "成功"
  }
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.3.2.1.2 获取指定认证策略

功能说明: 获取指定认证策略

1) API请求数据

GET

https://{host}/api/v1/namespaces/@namespace/authpolicys/@name

2) 输入参数

请求样例:

获取指定认证策略test

GET https://192.168.1.1/api/v1/namespaces/public/authpolicys/test

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	单条认证策略信息
data.uuid	string	认证策略UUID
data.name	string	策略名称 最小长度:1 最大长度:95
data.description	string	策略描述 最小长度:0 最大长度:95
data.position	int32	策略的位置信息 最小值:0 最大值:64
data IpRangeList	object	策略适用的ip和mac列表
data IpRangeList.ip	object[]	适用的IP地址信息
data IpRangeList.ip.start	string	ip范围的起始位置 格式:ip-address
data IpRangeList.ip.end	string	ip范围的终止位置 格式:ip-address
data IpRangeList.ip.bits	uint8	ip掩码简写

data.IpRangeList.ip.mask	string	ip掩码 格式:ip-address
data.IpRangeList.mac	string[]	适用的MAC地址信息
data.authType	string	可选，认证方式 枚举值:[NOAUTH:不需要认证 LOCALAUTH:本地密码认证 SSOAUTH:SSO登录认证]
data.noAuth	string	认证方式选择不需要认证时，需要设置用户名 枚举值:[IPTOUSER:ip作为用户名 MACTOUSER:mac作为用户名 PCTOUSER:计算机名作为用户名]
data.exceptUser	string[]	认证方式选择单点登录，需要添加例外用户
data.newUserOption	string	新用户选项 枚举值:[LOCALUSER:添加的本地组，作为本地用户 TEMPUSER:临时账号 FORBIDUSER:不允许新用户认证]
data.localUser	object	添加到本地组用户设置
data.localUser.localPathAdd	string	添加到本地组路径，默认是/ 最小长度:1 最大长度:1440
data.localUser.ldapAutoAdd	bool	是否开启LDAP自动同步到组
data.localUser.multipleLogin	string	多人同时登录限制 枚举值:[PUBLIC:可以多人同时登录 PRIVATE:只可以一人使用]
data.localUser.bindType	string	可选，地址绑定方式 枚举值:[DUPLEX:地址和用户双向绑定 SIMPLEX:地址和用户单向绑定]
data.localUser.bindAddr	string	可选，绑定地址选择 枚举值:[IP:绑定IP地址，MAC地址不绑定 MAC:绑定MAC地址，IP地址不绑定 IPMAC:绑定IP和MAC地址]
data.tempRightPath	string	作为临时账号时，分配的权限组路径,默认是/ 最小长度:1 最大长度:1440

请求成功样例：

```
{
  "data": {
    "position": 0,
    "noAuth": "IPTOUSER",
    "uuid": "FBC78BA00CC24E73B6DFCAF14C7C8EE3",
    "localUser": {
      "ldapAutoAdd": false,
      "bindType": "DUPLEX",
      "localPathAdd": "/",
      "multipleLogin": "PUBLIC",
      "bindAddr": "IP"
    },
    "newUserOption": "LOCALUSER",
    "ipRangeList": {
      "ip": [
        {
          "start": "192.168.1.10",
          "end": "192.168.1.10"
        }
      ],
      "mac": []
    },
    "name": "test",
    "authType": "NOAUTH",
    "description": ""
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.3.2.1.3 新增一条认证策略

功能说明: 新增一条认证策略

1) API请求数据

POST

https://{host}/api/v1/namespaces/@namespace/authpolicys

2) 输入参数

- URL参数

参数	类型	是否必选	描述
_where	string	否	指定位置, 用于移动/插入 枚举值:[top:顶部 bottom:底部 before:在目标位置之前 after:在目标位置之后]
_dest	string	否	目标位置关键字对应的值 最大长度:100
_key	string	否	目标位置关键字 枚举值:[position:指定目标位置关键字为 position name:指定目标位置关键字为 name]

• Body参数

参数	类型	是否必选	描述
obj	object	是	单条认证策略信息
obj.name	string	是	策略名称 最小长度:1 最大长度:95
obj.description	string	否	策略描述 最小长度:0 最大长度:95
obj.position	int32	否	策略的位置信息 最小值:0 最大值:64
obj IpRangeList	object	否	策略适用的ip和mac列表
obj IpRangeList.ip	object[]	是	适用的IP地址信息
obj IpRangeList.ip.start	string	是	ip范围的起始位置 格式:ip-address
obj IpRangeList.ip.end	string	否	ip范围的终止位置 格式:ip-address
obj IpRangeList.ip.bits	uint8	否	ip掩码简写
obj IpRangeList.ip.mask	string	否	ip掩码 格式:ip-address
obj IpRangeList.mac	string[]	是	适用的MAC地址信息
obj.authType	string	否	可选, 认证方式 默认值:true 枚举值:[NOAUTH:不需要认证 LOCALAUTH:本地密码认证 SSOAUTH:SSO登录认证]
obj.noAuth	string	否	认证方式选择不需要认证时, 需要设置用户名 默认值:true 枚举值:[IPTOUSER:ip作为用户名 MACTOUSER:mac作为用户名 PCTOUSER:计算机名作为用户名]
obj.exceptUser	string[]	是	认证方式选择单点登录, 需要添加例外用户
obj.newUserOption	string	否	新用户选项 默认值:true 枚举值:[LOCALUSER:添加的本地组, 作为本地用户 TEMPUSER:临时账号 FORBIDUSER:不允许新用户认证]
obj.localUser	object	是	添加到本地组用户设置

obj.localUser.localPathAdd	string	是	添加到本地组路径，默认是/ 最小长度:1 最大长度:1440
obj.localUser.ldapAutoAdd	bool	否	是否开启LDAP自动同步到组
obj.localUser.multipleLogin	string	否	多人同时登录限制 默认值:true 枚举值:[PUBLIC:可以多人同时登录 PRIVATE:只可以一人使用]
obj.localUser.bindType	string	否	可选，地址绑定方式 默认值:true 枚举值:[DUPLEX:地址和用户双向绑定 SIMPLEX:地址和用户单向绑定]
obj.localUser.bindAddr	string	否	可选，绑定地址选择 默认值:true 枚举值:[IP:绑定IP地址，MAC地址不绑定 MAC:绑定MAC地址，IP地址不绑定 IPMAC:绑定IP和MAC地址]
obj.tempRightPath	string	是	作为临时账号时，分配的权限组路径,默认是/ 最小长度:1 最大长度:1440

请求样例:

```
新增认证策略test,策略适用范围是192.168.1.10~192.168.1.100
POST https://192.168.1.1/api/v1/namespaces/public/authpolicys
{
  "name": "test",
  "ipRangeList": {
    "ip": [
      {
        "start": "192.168.1.10",
        "end": "192.168.1.100"
      }
    ]
  }
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	单条认证策略信息
data.uuid	string	认证策略UUID

data.name	string	策略名称 最小长度:1 最大长度:95
data.description	string	策略描述 最小长度:0 最大长度:95
data.position	int32	策略的位置信息 最小值:0 最大值:64
data.IpRangeList	object	策略适用的ip和mac列表
data.IpRangeList.ip	object[]	适用的IP地址信息
data.IpRangeList.ip.start	string	ip范围的起始位置 格式:ip-address
data.IpRangeList.ip.end	string	ip范围的终止位置 格式:ip-address
data.IpRangeList.ip.bits	uint8	ip掩码简写
data.IpRangeList.ip.mask	string	ip掩码 格式:ip-address
data.IpRangeList.mac	string[]	适用的MAC地址信息
data.authType	string	可选，认证方式 枚举值:[NOAUTH:不需要认证 LOCALAUTH:本地密码认证 SSOAUTH:SSO登录认证]
data.noAuth	string	认证方式选择不需要认证时，需要设置用户名 枚举值:[IPTOUSER:ip作为用户名 MACTOUSER:mac作为用户名 PCTOUSER:计算机名作为用户名]
data.exceptUser	string[]	认证方式选择单点登录，需要添加例外用户
data.newUserOption	string	新用户选项 枚举值:[LOCALUSER:添加的本地组，作为本地用户 TEMPUSER:临时账号 FORBIDUSER:不允许新用户认证]
data.localUser	object	添加到本地组用户设置
data.localUser.localPathAdd	string	添加到本地组路径，默认是/ 最小长度:1 最大长度:1440
data.localUser.ldapAutoAdd	bool	是否开启LDAP自动同步到组

data.localUser.multipleLogin	string	多人同时登录限制 枚举值: PUBLIC:可以多人同时登录 PRIVATE:只可以一人使用]
data.localUser.bindType	string	可选，地址绑定方式 枚举值: DUPLEX:地址和用户双向绑定 SIMPLEX:地址和用户单向绑定]
data.localUser.bindAddr	string	可选，绑定地址选择 枚举值: IP:绑定IP地址，MAC地址不绑定 MAC:绑定MAC地址，IP地址不绑定 IPMAC:绑定IP和MAC地址]
data.tempRightPath	string	作为临时账号时，分配的权限组路径,默认是/ 最小长度:1 最大长度:1440

请求成功样例：

```
{
  "data": {
    "name": "test",
    "IpRangeList": {
      "ip": [
        {
          "start": "192.168.1.10",
          "end": "192.168.1.100"
        }
      ]
    }
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.3.2.1.4 删除一条认证策略

功能说明：删除一条认证策略

1) API请求数据

DELETE https://{host}/api/v1/namespaces/@namespace/authpolicys/@name

2) 输入参数

请求样例：

删除一条认证策略test

DELETE https://192.168.1.1/api/v1/namespaces/public/authpolicys/test

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	单条认证策略信息
data.uuid	string	认证策略UUID
data.name	string	策略名称 最小长度:1 最大长度:95
data.description	string	策略描述 最小长度:0 最大长度:95
data.position	int32	策略的位置信息 最小值:0 最大值:64
data IpRangeList	object	策略适用的ip和mac列表
data IpRangeList.ip	object[]	适用的IP地址信息
data IpRangeList.ip.start	string	ip范围的起始位置 格式:ip-address
data IpRangeList.ip.end	string	ip范围的终止位置 格式:ip-address
data IpRangeList.ip.bits	uint8	ip掩码简写
data IpRangeList.ip.mask	string	ip掩码 格式:ip-address
data IpRangeList.mac	string[]	适用的MAC地址信息
data.authType	string	可选, 认证方式 枚举值:[NOAUTH:不需要认证 LOCALAUTH:本地密码认证 SSOAUTH:SSO登录认证]
data.noAuth	string	认证方式选择不需要认证时, 需要设置用户名 枚举值:[IPTOUSER:ip作为用户名 MACTOUSER:mac作为用户名 PCTOUSER:计算机名作为用户名]
data.exceptUser	string[]	认证方式选择单点登录, 需要添加例外用户
data.newUserOption	string	新用户选项 枚举值:[LOCALUSER:添加的本地组, 作为本地用户 TEMPUSER:临时账号 FORBIDUSER:不允许新用户认证]

data.localUser	object	添加到本地组用户设置
data.localUser.localPathAdd	string	添加到本地组路径，默认是/ 最小长度:1 最大长度:1440
data.localUser.ldapAutoAdd	bool	是否开启LDAP自动同步到组
data.localUser.multipleLogin	string	多人同时登录限制 枚举值:[PUBLIC:可以多人同时登录 PRIVATE:只可以一人使用]
data.localUser.bindType	string	可选，地址绑定方式 枚举值:[DUPLEX:地址和用户双向绑定 SIMPLEX:地址和用户单向绑定]
data.localUser.bindAddr	string	可选，绑定地址选择 枚举值:[IP:绑定IP地址，MAC地址不绑定 MAC:绑定MAC地址，IP地址不绑定 IPMAC:绑定IP和MAC地址]
data.tempRightPath	string	作为临时账号时，分配的权限组路径,默认是/ 最小长度:1 最大长度:1440

请求成功样例：

```
{
  "data": {
    "name": "test"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.3.2.1.5 批量删除认证策略

功能说明：批量删除认证策略

1) API请求数据

POST	https://{host}/api/batch/v1/namespaces/@namespace/authpolicys?_method=delete
------	--

2) 输入参数

• Body参数

参数	类型	是否必选	描述
obj	object[]	是	认证策略列表

obj.name	string	是	策略名称 最小长度:1 最大长度:95
obj.description	string	否	策略描述 最小长度:0 最大长度:95
obj.position	int32	否	策略的位置信息 最小值:0 最大值:64
obj.IpRangeList	object	否	策略适用的ip和mac列表
obj.IpRangeList.ip	object[]	是	适用的IP地址信息
obj.IpRangeList.ip.start	string	是	ip范围的起始位置 格式:ip-address
obj.IpRangeList.ip.end	string	否	ip范围的终止位置 格式:ip-address
obj.IpRangeList.ip.bits	uint8	否	ip掩码简写
obj.IpRangeList.ip.mask	string	否	ip掩码 格式:ip-address
obj.IpRangeList.mac	string[]	是	适用的MAC地址信息
obj.authType	string	否	可选, 认证方式 默认值:true 枚举值:[NOAUTH:不需要认证 LOCALAUTH:本地密码认证 SSOAUTH:SSO登录认证]
obj.noAuth	string	否	认证方式选择不需要认证时, 需要设置用户名 默认值:true 枚举值:[IPTOUSER:ip作为用户名 MACTOUSER:mac作为用户名 PCTOUSER:计算机名作为用户名]
obj.exceptUser	string[]	是	认证方式选择单点登录, 需要添加例外用户
obj.newUserOption	string	否	新用户选项 默认值:true 枚举值:[LOCALUSER:添加的本地组, 作为本地用户 TEMPUSER:临时账号 FORBIDUSER:不允许新用户认证]
obj.localUser	object	是	添加到本地组用户设置
obj.localUser.localPathAdd	string	是	添加到本地组路径, 默认是/ 最小长度:1 最大长度:1440

obj.localUser.ldapAutoAdd	bool	否	是否开启LDAP自动同步到组
obj.localUser.multipleLogin	string	否	多人同时登录限制 默认值:true 枚举值:[PUBLIC:可以多人同时登录 PRIVATE:只可以一人使用]
obj.localUser.bindType	string	否	可选, 地址绑定方式 默认值:true 枚举值:[DUPLEX:地址和用户双向绑定 SIMPLEX:地址和用户单向绑定]
obj.localUser.bindAddr	string	否	可选, 绑定地址选择 默认值:true 枚举值:[IP:绑定IP地址, MAC地址不绑定 MAC:绑定MAC地址, IP地址不绑定 IPMAC:绑定IP和MAC地址]
obj.tempRightPath	string	是	作为临时账号时, 分配的权限组路径,默认是/ 最小长度:1 最大长度:1440

请求样例:

批量删除认证策略

POST https://192.168.1.1/api/batch/v1/namespaces/public/authpolicys?_method=DELETE

```
[
  {
    "name": "test1"
  },
  {
    "name": "test2"
  },
  {
    "name": "test3"
  }
]
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	认证策略列表
data.uuid	string	认证策略UUID
data.name	string	策略名称 最小长度:1 最大长度:95

data.description	string	策略描述 最小长度:0 最大长度:95
data.position	int32	策略的位置信息 最小值:0 最大值:64
data.IpRangeList	object	策略适用的ip和mac列表
data.IpRangeList.ip	object[]	适用的IP地址信息
data.IpRangeList.ip.start	string	ip范围的起始位置 格式:ip-address
data.IpRangeList.ip.end	string	ip范围的终止位置 格式:ip-address
data.IpRangeList.ip.bits	uint8	ip掩码简写
data.IpRangeList.ip.mask	string	ip掩码 格式:ip-address
data.IpRangeList.mac	string[]	适用的MAC地址信息
data.authType	string	可选, 认证方式 枚举值:[NOAUTH:不需要认证 LOCALAUTH:本地密码认证 SSOAUTH:SSO登录认证]
data.noAuth	string	认证方式选择不需要认证时, 需要设置用户名 枚举值:[IPTOUSER:ip作为用户名 MACTOUSER:mac作为用户名 PCTOUSER:计算机名作为用户名]
data.exceptUser	string[]	认证方式选择单点登录, 需要添加例外用户
data.newUserOption	string	新用户选项 枚举值:[LOCALUSER:添加的本地组, 作为本地用户 TEMPUSER:临时账号 FORBIDUSER:不允许新用户认证]
data.localUser	object	添加到本地组用户设置
data.localUser.localPathAdd	string	添加到本地组路径, 默认是/ 最小长度:1 最大长度:1440
data.localUser.IdapAutoAdd	bool	是否开启LDAP自动同步到组
data.localUser.multipleLogin	string	多人同时登录限制 枚举值:[PUBLIC:可以多人同时登录 PRIVATE:只可以一人使用]

data.localUser.bindType	string	可选，地址绑定方式 枚举值:[DUPLEX:地址和用户双向绑定 SIMPLEX:地址和用户单向绑定]
data.localUser.bindAddr	string	可选，绑定地址选择 枚举值:[IP:绑定IP地址，MAC地址不绑定 MAC:绑定MAC地址，IP地址不绑定 IPMAC:绑定IP和MAC地址]
data.tempRightPath	string	作为临时账号时，分配的权限组路径,默认是/ 最小长度:1 最大长度:1440

请求成功样例：

```
{
  "data": [
    {
      "name": "test1"
    },
    {
      "name": "test2"
    },
    {
      "name": "test3"
    }
  ],
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.3.2.1.6 全量更新一条认证策略

功能说明：全量更新一条认证策略

1) API请求数据

PUT

https://{host}/api/v1/namespaces/@namespace/authpolicys/@name

2) 输入参数

• Body参数

参数	类型	是否必选	描述
obj	object	是	单条认证策略信息
obj.name	string	是	策略名称 最小长度:1 最大长度:95
obj.description	string	否	策略描述 最小长度:0 最大长度:95

obj.position	int32	否	策略的位置信息 最小值:0 最大值:64
obj.IpRangeList	object	否	策略适用的ip和mac列表
obj.IpRangeList.ip	object[]	是	适用的IP地址信息
obj.IpRangeList.ip.start	string	是	ip范围的起始位置 格式:ip-address
obj.IpRangeList.ip.end	string	否	ip范围的终止位置 格式:ip-address
obj.IpRangeList.ip.bits	uint8	否	ip掩码简写
obj.IpRangeList.ip.mask	string	否	ip掩码 格式:ip-address
obj.IpRangeList.mac	string[]	是	适用的MAC地址信息
obj.authType	string	否	可选, 认证方式 默认值:true 枚举值:[NOAUTH:不需要认证 LOCALAUTH:本地密码认证 SSOAUTH:SSO登录认证]
obj.noAuth	string	否	认证方式选择不需要认证时, 需要设置用户名 默认值:true 枚举值:[IPTOUSER:ip作为用户名 MACTOUSER:mac作为用户名 PCTOUSER:计算机名作为用户名]
obj.exceptUser	string[]	是	认证方式选择单点登录, 需要添加例外用户
obj.newUserOption	string	否	新用户选项 默认值:true 枚举值:[LOCALUSER:添加的本地组, 作为本地用户 TEMPUSER:临时账号 FORBIDUSER:不允许新用户认证]
obj.localUser	object	是	添加到本地组用户设置
obj.localUser.localPathAdd	string	是	添加到本地组路径, 默认是/ 最小长度:1 最大长度:1440
obj.localUser.ldapAutoAdd	bool	否	是否开启LDAP自动同步到组
obj.localUser.multipleLogin	string	否	多人同时登录限制 默认值:true 枚举值:[PUBLIC:可以多人同时登录 PRIVATE:只可以一人使用]

obj.localUser.bindType	string	否	可选，地址绑定方式 默认值:true 枚举值:[DUPLEX:地址和用户双向绑定 SIMPLEX:地址和用户单向绑定]
obj.localUser.bindAddr	string	否	可选，绑定地址选择 默认值:true 枚举值:[IP:绑定IP地址，MAC地址不绑定 MAC:绑定MAC地址，IP地址不绑定 IPMAC:绑定IP和MAC地址]
obj.tempRightPath	string	是	作为临时账号时，分配的权限组路径,默认是/ 最小长度:1 最大长度:1440

请求样例:

```
PUT https://192.168.1.1/api/v1/namespaces/public/authpolicys/admin
{
  "position": 0,
  "tempRightPath": "/test/xxx/",
  "IpRangeList": {
    "ip": [
      {
        "start": "192.168.1.1"
      }
    ],
    "mac": [
      "6a:54:a6:27:27:3a"
    ]
  },
  "exceptUser": [
    "test"
  ],
  "localUser": {
    "ldapAutoAdd": true,
    "bindType": "DUPLEX",
    "localPathAdd": "/test/xxx/",
    "multipleLogin": "PUBLIC",
    "bindAddr": "IP"
  },
  "newUserOption": "LOCALUSER",
  "noAuth": "IPTOUSER",
  "authType": "NOAUTH",
  "name": "test",
  "description": "description"
}
```

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	单条认证策略信息
data.uuid	string	认证策略UUID

data.name	string	策略名称 最小长度:1 最大长度:95
data.description	string	策略描述 最小长度:0 最大长度:95
data.position	int32	策略的位置信息 最小值:0 最大值:64
data.IpRangeList	object	策略适用的ip和mac列表
data.IpRangeList.ip	object[]	适用的IP地址信息
data.IpRangeList.ip.start	string	ip范围的起始位置 格式:ip-address
data.IpRangeList.ip.end	string	ip范围的终止位置 格式:ip-address
data.IpRangeList.ip.bits	uint8	ip掩码简写
data.IpRangeList.ip.mask	string	ip掩码 格式:ip-address
data.IpRangeList.mac	string[]	适用的MAC地址信息
data.authType	string	可选，认证方式 枚举值:[NOAUTH:不需要认证 LOCALAUTH:本地密码认证 SSOAUTH:SSO登录认证]
data.noAuth	string	认证方式选择不需要认证时，需要设置用户名 枚举值:[IPTOUSER:ip作为用户名 MACTOUSER:mac作为用户名 PCTOUSER:计算机名作为用户名]
data.exceptUser	string[]	认证方式选择单点登录，需要添加例外用户
data.newUserOption	string	新用户选项 枚举值:[LOCALUSER:添加的本地组，作为本地用户 TEMPUSER:临时账号 FORBIDUSER:不允许新用户认证]
data.localUser	object	添加到本地组用户设置
data.localUser.localPathAdd	string	添加到本地组路径，默认是/ 最小长度:1 最大长度:1440
data.localUser.IdapAutoAdd	bool	是否开启LDAP自动同步到组

data.localUser.multipleLogin	string	多人同时登录限制 枚举值:[PUBLIC:可以多人同时登录 PRIVATE:只可以一人使用]
data.localUser.bindType	string	可选，地址绑定方式 枚举值:[DUPLEX:地址和用户双向绑定 SIMPLEX:地址和用户单向绑定]
data.localUser.bindAddr	string	可选，绑定地址选择 枚举值:[IP:绑定IP地址，MAC地址不绑定 MAC:绑定MAC地址，IP地址不绑定 IPMAC:绑定IP和MAC地址]
data.tempRightPath	string	作为临时账号时，分配的权限组路径,默认是/ 最小长度:1 最大长度:1440

请求成功样例：

```
{
  "message": "",
  "data": {
    "position": 0,
    "tempRightPath": "/test/xxx/",
    "IpRangeList": {
      "ip": [
        {
          "start": "192.168.1.1"
        }
      ],
      "mac": [
        "6a:54:a6:27:27:3a"
      ]
    },
    "exceptUser": [
      "test"
    ],
    "localUser": {
      "ldapAutoAdd": true,
      "bindType": "DUPLEX",
      "localPathAdd": "/test/xxx/",
      "multipleLogin": "PUBLIC",
      "bindAddr": "IP"
    },
    "newUserOption": "LOCALUSER",
    "noAuth": "IPTOUSER",
    "authType": "NOAUTH",
    "uuid": "07150a520899461e956bd9f0a69ea82d",
    "name": "test",
    "description": "description"
  },
  "code": 0
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```


4.3.2.1.7 增量更新一条认证策略

功能说明：增量更新一条认证策略

1) API请求数据

PATCH <https://{host}/api/v1/namespaces/@namespace/authpolicys/@name>

2) 输入参数

• URL参数

参数	类型	是否必选	描述
_arrayop	string	否	对象属性是数组的操作控制 枚举值:[add:添加对应的数组元素 remove:删除对应的数组元素]
_key	string	否	目标位置关键字 枚举值:[position:指定目标位置关键字为 position name:指定目标位置关键字为 name]
_dest	string	否	目标位置关键字对应的值 最大长度:100
_where	string	否	指定位置, 用于移动/插入 枚举值:[top:顶部 bottom:底部 before:在目标位置之前 after:在目标位置之后]

• Body参数

参数	类型	是否必选	描述
obj	object	是	单条认证策略信息
obj.name	string	是	策略名称 最小长度:1 最大长度:95
obj.description	string	否	策略描述 最小长度:0 最大长度:95
obj.position	int32	否	策略的位置信息 最小值:0 最大值:64
obj.IpRangeList	object	否	策略适用的ip和mac列表
obj.IpRangeList.ip	object[]	是	适用的IP地址信息
obj.IpRangeList.ip.start	string	是	ip范围的起始位置 格式:ip-address
obj.IpRangeList.ip.end	string	否	ip范围的终止位置 格式:ip-address

obj.IpRangeList.ip.bits	uint8	否	ip掩码简写
obj.IpRangeList.ip.mask	string	否	ip掩码 格式:ip-address
obj.IpRangeList.mac	string[]	是	适用的MAC地址信息
obj.authType	string	否	可选, 认证方式 默认值:true 枚举值:[NOAUTH:不需要认证 LOCALAUTH:本地密码认证 SSOAUTH:SSO登录认证]
obj.noAuth	string	否	认证方式选择不需要认证时, 需要设置用户名 默认值:true 枚举值:[IPTOUSER:ip作为用户名 MACTOUSER:mac作为用户名 PCTOUSER:计算机名作为用户名]
obj.exceptUser	string[]	是	认证方式选择单点登录, 需要添加例外用户
obj.newUserOption	string	否	新用户选项 默认值:true 枚举值:[LOCALUSER:添加的本地组, 作为本地用户 TEMPUSER:临时账号 FORBIDUSER:不允许新用户认证]
obj.localUser	object	是	添加到本地组用户设置
obj.localUser.localPathAdd	string	是	添加到本地组路径, 默认是/ 最小长度:1 最大长度:1440
obj.localUser.ldapAutoAdd	bool	否	是否开启LDAP自动同步到组
obj.localUser.multipleLogin	string	否	多人同时登录限制 默认值:true 枚举值:[PUBLIC:可以多人同时登录 PRIVATE:只可以一人使用]
obj.localUser.bindType	string	否	可选, 地址绑定方式 默认值:true 枚举值:[DUPLEX:地址和用户双向绑定 SIMPLEX:地址和用户单向绑定]

obj.localUser.bindAddr	string	否	可选，绑定地址选择 默认值:true 枚举值:[IP:绑定IP地址，MAC地址不绑定 MAC:绑定MAC地址，IP地址不绑定 IPMAC:绑定IP和MAC地址]
obj.tempRightPath	string	是	作为临时账号时，分配的权限组路径,默认是/ 最小长度:1 最大长度:1440

请求样例:

增量修改认证策略test,把描述改为xxxxx,策略适用范围改为192.168.1.10~192.168.1.100,其他字段不修改

```
PATCH https://192.168.1.1/api/v1/namespaces/public/authpolicys/test
{
  "IpRangeList": {
    "ip": [
      {
        "start": "192.168.1.10",
        "end": "192.168.1.100"
      }
    ]
  },
  "name": "test",
  "description": "xxxxx"
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	单条认证策略信息
data.uuid	string	认证策略UUID
data.name	string	策略名称 最小长度:1 最大长度:95
data.description	string	策略描述 最小长度:0 最大长度:95
data.position	int32	策略的位置信息 最小值:0 最大值:64
data.IpRangeList	object	策略适用的ip和mac列表
data.IpRangeList.ip	object[]	适用的IP地址信息
data.IpRangeList.ip.start	string	ip范围的起始位置 格式:ip-address
data.IpRangeList.ip.end	string	ip范围的终止位置 格式:ip-address
data.IpRangeList.ip.bits	uint8	ip掩码简写

data.IpRangeList.ip.mask	string	ip掩码 格式:ip-address
data.IpRangeList.mac	string[]	适用的MAC地址信息
data.authType	string	可选，认证方式 枚举值:[NOAUTH:不需要认证 LOCALAUTH:本地密码认证 SSOAUTH:SSO登录认证]
data.noAuth	string	认证方式选择不需要认证时，需要设置用户名 枚举值:[IPTOUSER:ip作为用户名 MACTOUSER:mac作为用户名 PCTOUSER:计算机名作为用户名]
data.exceptUser	string[]	认证方式选择单点登录，需要添加例外用户
data.newUserOption	string	新用户选项 枚举值:[LOCALUSER:添加的本地组，作为本地用户 TEMPUSER:临时账号 FORBIDUSER:不允许新用户认证]
data.localUser	object	添加到本地组用户设置
data.localUser.localPathAdd	string	添加到本地组路径，默认是/ 最小长度:1 最大长度:1440
data.localUser.ldapAutoAdd	bool	是否开启LDAP自动同步到组
data.localUser.multipleLogin	string	多人同时登录限制 枚举值:[PUBLIC:可以多人同时登录 PRIVATE:只可以一人使用]
data.localUser.bindType	string	可选，地址绑定方式 枚举值:[DUPLEX:地址和用户双向绑定 SIMPLEX:地址和用户单向绑定]
data.localUser.bindAddr	string	可选，绑定地址选择 枚举值:[IP:绑定IP地址，MAC地址不绑定 MAC:绑定MAC地址，IP地址不绑定 IPMAC:绑定IP和MAC地址]
data.tempRightPath	string	作为临时账号时，分配的权限组路径,默认是/ 最小长度:1 最大长度:1440

请求成功样例：

```
{
  "data": {
    "ipRangeList": {
      "ip": [
        {
          "start": "192.168.1.10",
          "end": "192.168.1.100"
        }
      ]
    },
    "name": "test",
    "description": "xxxxx"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.3.2.1.8 批量更新认证策略

功能说明: 批量更新认证策略

1) API请求数据

PATCH <https://{host}/api/batch/v1/namespaces/@namespace/authpolicys>

2) 输入参数

• URL参数

参数	类型	是否必选	描述
_where	string	否	指定位置, 用于移动/插入 枚举值:[top:顶部 bottom:底部 before:在目标位置之前 after:在目标位置之后]
_dest	string	否	目标位置关键字对应的值 最大长度:100
_key	string	否	目标位置关键字 枚举值:[position:指定目标位置关键字为 position name:指定目标位置关键字为 name]

• Body参数

参数	类型	是否必选	描述
obj	object[]	是	认证策略列表
obj.name	string	是	策略名称 最小长度:1 最大长度:95

obj.description	string	否	策略描述 最小长度:0 最大长度:95
obj.position	int32	否	策略的位置信息 最小值:0 最大值:64
obj.IpRangeList	object	否	策略适用的ip和mac列表
obj.IpRangeList.ip	object[]	是	适用的IP地址信息
obj.IpRangeList.ip.start	string	是	ip范围的起始位置 格式:ip-address
obj.IpRangeList.ip.end	string	否	ip范围的终止位置 格式:ip-address
obj.IpRangeList.ip.bits	uint8	否	ip掩码简写
obj.IpRangeList.ip.mask	string	否	ip掩码 格式:ip-address
obj.IpRangeList.mac	string[]	是	适用的MAC地址信息
obj.authType	string	否	可选, 认证方式 默认值:true 枚举值:[NOAUTH:不需要认证 LOCALAUTH:本地密码认证 SSOAUTH:SSO登录认证]
obj.noAuth	string	否	认证方式选择不需要认证时, 需要设置用户名 默认值:true 枚举值:[IPTOUSER:ip作为用户名 MACTOUSER:mac作为用户名 PCTOUSER:计算机名作为用户名]
obj.exceptUser	string[]	是	认证方式选择单点登录, 需要添加例外用户
obj.newUserOption	string	否	新用户选项 默认值:true 枚举值:[LOCALUSER:添加的本地组, 作为本地用户 TEMPUSER:临时账号 FORBIDUSER:不允许新用户认证]
obj.localUser	object	是	添加到本地组用户设置
obj.localUser.localPathAdd	string	是	添加到本地组路径, 默认是/ 最小长度:1 最大长度:1440
obj.localUser.ldapAutoAdd	bool	否	是否开启LDAP自动同步到组

obj.localUser.multipleLogin	string	否	多人同时登录限制 默认值:true 枚举值:[PUBLIC:可以多人同时登录 PRIVATE:只可以一人使用]
obj.localUser.bindType	string	否	可选, 地址绑定方式 默认值:true 枚举值:[DUPLEX:地址和用户双向绑定 SIMPLEX:地址和用户单向绑定]
obj.localUser.bindAddr	string	否	可选, 绑定地址选择 默认值:true 枚举值:[IP:绑定IP地址, MAC地址不绑定 MAC:绑定MAC地址, IP地址不绑定 IPMAC:绑定IP和MAC地址]
obj.tempRightPath	string	是	作为临时账号时, 分配的权限组路径,默认是/ 最小长度:1 最大长度:1440

请求样例:

批量修改认证策略,修改内容: 认证方式改为不需要认证, 分别把IP、MAC、PC作为用户名, 其他字段不修改

PATCH https://192.168.1.1/api/batch/v1/namespaces/public/authpolicys

```
[
  {
    "noAuth": "IPTOUSER",
    "name": "test1",
    "authType": "NOAUTH"
  },
  {
    "noAuth": "MACTOUSER",
    "name": "test2",
    "authType": "NOAUTH"
  },
  {
    "noAuth": "PCTOUSER",
    "name": "test3",
    "authType": "NOAUTH"
  }
]
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	认证策略列表
data.uuid	string	认证策略UUID
data.name	string	策略名称 最小长度:1 最大长度:95

data.description	string	策略描述 最小长度:0 最大长度:95
data.position	int32	策略的位置信息 最小值:0 最大值:64
data.IpRangeList	object	策略适用的ip和mac列表
data.IpRangeList.ip	object[]	适用的IP地址信息
data.IpRangeList.ip.start	string	ip范围的起始位置 格式:ip-address
data.IpRangeList.ip.end	string	ip范围的终止位置 格式:ip-address
data.IpRangeList.ip.bits	uint8	ip掩码简写
data.IpRangeList.ip.mask	string	ip掩码 格式:ip-address
data.IpRangeList.mac	string[]	适用的MAC地址信息
data.authType	string	可选, 认证方式 枚举值:[NOAUTH:不需要认证 LOCALAUTH:本地密码认证 SSOAUTH:SSO登录认证]
data.noAuth	string	认证方式选择不需要认证时, 需要设置用户名 枚举值:[IPTOUSER:ip作为用户名 MACTOUSER:mac作为用户名 PCTOUSER:计算机名作为用户名]
data.exceptUser	string[]	认证方式选择单点登录, 需要添加例外用户
data.newUserOption	string	新用户选项 枚举值:[LOCALUSER:添加的本地组, 作为本地用户 TEMPUSER:临时账号 FORBIDUSER:不允许新用户认证]
data.localUser	object	添加到本地组用户设置
data.localUser.localPathAdd	string	添加到本地组路径, 默认是/ 最小长度:1 最大长度:1440
data.localUser.IdapAutoAdd	bool	是否开启LDAP自动同步到组
data.localUser.multipleLogin	string	多人同时登录限制 枚举值:[PUBLIC:可以多人同时登录 PRIVATE:只可以一人使用]

data.localUser.bindType	string	可选，地址绑定方式 枚举值:[DUPLEX:地址和用户双向绑定 SIMPLEX:地址和用户单向绑定]
data.localUser.bindAddr	string	可选，绑定地址选择 枚举值:[IP:绑定IP地址，MAC地址不绑定 MAC:绑定MAC地址，IP地址不绑定 IPMAC:绑定IP和MAC地址]
data.tempRightPath	string	作为临时账号时，分配的权限组路径,默认是/ 最小长度:1 最大长度:1440

请求成功样例：

```
{
  "data": [
    {
      "noAuth": "IPTOUSER",
      "name": "test1",
      "authType": "NOAUTH"
    },
    {
      "noAuth": "MACTOUSER",
      "name": "test2",
      "authType": "NOAUTH"
    },
    {
      "noAuth": "PCTOUSER",
      "name": "test3",
      "authType": "NOAUTH"
    }
  ],
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.3.2.1.9 设置用户认证开关状态

功能说明：设置用户认证开关状态

1) API请求数据

PATCH

https://{host}/api/v1/namespaces/@namespace/authstatus

2) 输入参数

- Body参数

参数	类型	是否必选	描述
obj	object	是	用于设置用户认证开关状态
obj.authEnable	bool	否	用户认证开关状态

请求样例：

```
设置用户认证开关状态为开启
PATCH https://192.168.1.1/api/v1/namespaces/public/authstatus
{
  "authEnable": true
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	用于设置用户认证开关状态
data.authEnable	bool	用户认证开关状态

请求成功样例：

```
{
  "data": {
    "authEnable": true
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.3.2.1.10 获取用户认证状态

功能说明：获取用户认证状态

1) API请求数据

GET

https://{host}/api/v1/namespaces/@namespace/authstatus

2) 输入参数

请求样例：

```
获取用户认证开关状态
GET https://192.168.1.1/api/v1/namespaces/public/authstatus
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	用于设置用户认证开关状态
data.authEnable	bool	用户认证开关状态

请求成功样例：

```
{
  "data": {
    "authEnable": true
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.3.2.1.11 更新认证区域

功能说明: 更新认证区域

1) API请求数据

PATCH https://{host}/api/v1/namespaces/@namespace/authzone

2) 输入参数

• URL参数

参数	类型	是否必选	描述
_arrayop	string	否	对象属性是数组的操作控制 枚举值: add:添加对应的数组元素 remove:删除对应的数组元素]

• Body参数

参数	类型	是否必选	描述
obj	object[]	是	认证区域的请求数据结构
obj.zone	string	是	认证区域的名称 最小长度:1 最大长度:95

请求样例:

编辑用户认证区域为L3_untrust_A,L3_untrust_B

PATCH https://192.168.1.1/api/v1/namespaces/public/authzone

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	认证区域的请求数据结构
data.zone	string	认证区域的名称 最小长度:1 最大长度:95

请求成功样例：

```
{
  "data": [
    {
      "zone": "L3_untrust_A"
    },
    {
      "zone": "L3_untrust_B"
    }
  ],
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.3.2.1.12 查看认证区域

功能说明：查看认证区域

1) API请求数据

GET

https://{host}/api/v1/namespaces/@namespace/authzone

2) 输入参数

请求样例：

获取用户认证区域信息
GET https://192.168.1.1/api/v1/namespaces/public/authzone

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	认证区域的请求数据结构
data.zone	string	认证区域的名称 最小长度:1 最大长度:95

请求成功样例：

```
{
  "data": [
    {
      "zone": "L3_untrust_A"
    },
    {
      "zone": "L3_untrust_B"
    }
  ],
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.3.3 用户认证状态

4.3.3.1 获取在线用户列表

功能说明：获取在线用户列表

1) API请求数据

POST https://{host}/api/v1/namespaces/@namespace/onlineusers?_method=get

2) 输入参数

• URL 参数

参数	类型	是否必选	描述
_length	int32	否	获取最大长度，限制最大值，如果数据不够可能返回小于 length 最小值:1 最大值:200 默认值:true
_sortby	string	否	指定排序字段 最大长度:100
_order	string	否	指定排序方式（正序/倒序） 默认值:true 枚举值:[asc:按升序排列 desc:按降序排列]
_start	int32	否	起始位置, 从 0 开始 最小值:0 最大值:2000000

• Body参数

参数	类型	是否必选	描述
obj	object	是	获取在线用户列表过滤参数
obj.userStatus	string	否	过滤用户状态(默认为ALL) 默认值:true 枚举值:[ACTIVE:活跃用户 FROZEN:冻结用户 ALL:所有状态用户]
obj.filterType	string	否	过滤方式(默认为ORG) 默认值:true 枚举值:[ORG:用户组过滤 IP:IP地址过滤 USER:在线用户]

obj.org	string	是	当前所属组(默认为/) 最小长度:1 最大长度:1440
obj.users	string[]	是	用户名
obj.ipRanges	object[]	是	IP范围列表
obj.ipRanges.start	string	是	ip范围的起始位置 格式:ip-address
obj.ipRanges.end	string	否	ip范围的终止位置 格式:ip-address
obj.ipRanges.bits	uint8	否	ip掩码简写
obj.ipRanges.mask	string	否	ip掩码 格式:ip-address

请求样例:

获取在线用户列表, 过滤条件: 用户状态为全部, 当前所属组为/

```
POST https://192.168.1.1/api/v1/namespaces/public/onlineusers?_method=GET
{
  "userStatus": "ALL",
  "filterType": "ORG",
  "org": "/"
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	分页数据
data.totalItems	int32	总共多少项目
data.totalPages	int32	总共多少页
data.pageNumber	int32	当前页码, 从 1 开始
data.pageSize	int32	每页多大
data.itemsOffset	int32	当前条目偏移, 从 0 开始
data.itemLength	int32	数据列表长度
data.items	object[]	有效数据列表
data.items.auth	string	认证方式
data.items.org	string	当前所属组(默认为/) 最小长度:1 最大长度:1440
data.items.userName	string	用户名
data.items.ip	string	ip地址 格式:ip-address
data.items.onlineTime	string	在线时长
data.items.enable	bool	是否可上网 (未冻结)

data.items.loginTime	string	登录或冻结时间
----------------------	--------	---------

请求成功样例:

```
{
  "data": {
    "itemsOffset": 0,
    "items": [
      {
        "enable": true,
        "loginTime": "2020-7-15 10:58:28",
        "ip": "200.200.152.191",
        "org": "/",
        "auth": "密码认证",
        "onlineTime": "0:06:23",
        "userName": "xxx"
      }
    ],
    "totalItems": 1,
    "pageNumber": 1,
    "itemLength": 1,
    "totalPages": 1,
    "pageSize": 100
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.3.3.2 注销在线用户

功能说明: 注销在线用户, 从在线用户列表中删除

1) API请求数据

POST	https://{host}/api/batch/v1/namespaces/@namespace/onlineusers?_method=delete
------	--

2) 输入参数

- Body参数

参数	类型	是否必选	描述
obj	string[]	是	在线用户列表

请求样例:

注销在线用户, 该用户登录ip为192.168.1.1

```
POST https://192.168.1.1/api/batch/v1/namespaces/public/onlineusers?_method=DELETE
[
  "192.168.1.1"
]
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码

message	string	错误信息
data	string[]	在线用户列表

请求成功样例:

```
{
  "data": [
    "192.168.1.1"
  ],
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

4.3.3.3 修改在线用户状态

功能说明: 修改在线用户状态, 指定为冻结或解冻

1) API请求数据

POST <https://{host}/api/v1/namespaces/@namespace/onlineusers>

2) 输入参数

• Body参数

参数	类型	是否必选	描述
obj	object	是	修改在线用户请求数据
obj.operate	string	是	修改在线用户请求数据 枚举值: BLOCK:冻结用户 UNBLOCK:解冻用户]
obj.blockTime	int32	是	冻结时间(单位分钟,范围1-1440),需指定o perate为BLOCK才生效 最小值:1 最大值:1440
obj.ip	string[]	是	在线用户列表

请求样例:

```
解冻登录ip为192.168.1.1的在线用户
POST https://192.168.1.1/api/v1/namespaces/public/onlineusers
{
  "operate": "UNBLOCK",
  "ip": [
    "192.168.1.1"
  ]
}
```

3) 输出参数

• API返回数据

参数	类型	描述
----	----	----

code	int	错误码
message	string	错误信息
data	object	修改在线用户请求数据
data.operate	string	修改在线用户请求数据 枚举值:[BLOCK:冻结用户 UNBLOCK:解冻用户]
data.blockTime	int32	冻结时间(单位分钟,范围1-1440),需指定operate为BLOCK才生效 最小值:1 最大值:1440
data.ip	string[]	在线用户列表

请求成功样例:

```
{
  "data": {
    "operate": "UNBLOCK",
    "ip": [
      "192.168.1.1"
    ]
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

5 系统

5.1 管理员账户

5.1.1 查询所有已创建的管理员账户信息

功能说明：通过本接口，可以查询所有已创建的管理员账户信息

1) API请求数据

GET

https://{host}/api/v1/namespaces/@namespace/account

2) 输入参数

• URL参数

参数	类型	是否必选	描述
_start	int32	否	起始位置, 从 0 开始 最小值:0 最大值:2000000
_length	int32	否	获取最大长度, 限制最大值, 如果数据不够可能返回小于 length 最小值:1 最大值:200 默认值:true

请求样例:

查询管理员账户信息, 分页起始值0, 每页3个账户的信息

GET https://192.168.1.1/api/v1/namespaces/public/account?_start=0&_length=3

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	分页数据
data.totalItems	int32	总共多少项目
data.totalPages	int32	总共多少页
data.pageNumber	int32	当前页码, 从 1 开始
data.pageSize	int32	每页多大
data.itemsOffset	int32	当前条目偏移, 从 0 开始
data.itemLength	int32	数据列表长度
data.items	object[]	有效数据列表
data.items.name	string	用户名 最小长度:1 最大长度:60

data.items.enable	bool	账户状态，启用/禁用
data.items.description	string	描述 最小长度:0 最大长度:95
data.items.roleName	string	账户角色 枚举值: ADMINISTRATOR:超级管理员 COMMON:普通管理员 SAFE:三权分立账户，安全管理员 LOG:三权分立账户，审计员 SYSTEM:三权分立账户，系统管理员 REMOTE:远程认证账户]
data.items.authentication	object	认证策略
data.items.authentication.usbKeyEnable	bool	usb-key认证是否启用
data.items.authentication.fingerprintEnable	bool	指纹校验是否启用
data.items.authentication.createUsbKeyStatus	bool	是否创建了usbkey证书
data.items.authentication.dKey	string	采集的指纹 最大长度:24
data.items.authentication.manageMode	string[]	管理方式
data.items.modulePermissions	object[]	模块权限
data.items.modulePermissions.name	string	模块名 最大长度:60
data.items.modulePermissions.leaf	bool	模块是否为子项
data.items.modulePermissions.moduleClass	string	模块所属大类名 最大长度:60
data.items.modulePermissions.editPermission	object	模块的编辑权限
data.items.modulePermissions.editPermission.authorizeStatus	bool	是否有权限
data.items.modulePermissions.editPermission.modifyEnable	bool	能否修改权限
data.items.modulePermissions.viewPermission	object	模块的查看权限
data.items.modulePermissions.viewPermission.authorizeStatus	bool	是否有权限

data.items.modulePermissions.viewPermission.modifyEnable	bool	能否修改权限
--	------	--------

请求成功样例:

```
{
  "data": {
    "itemsOffset": 0,
    "items": [
      {
        "enable": true,
        "authentication": {
          "usbKeyEnable": false,
          "fingerprintEnable": false,
          "manageMode": [
            "WEBCONSOLE",
            "APIINTERFACE",
            "TECHNICALSUPPORT"
          ]
        },
        "modulePermissions": [
          {
            "name": "首页",
            "leaf": false
          },
          {
            "editPermission": {
              "modifyEnable": false,
              "authorizeStatus": true
            },
            "viewPermission": {
              "modifyEnable": false,
              "authorizeStatus": true
            },
            "leaf": true,
            "name": "首页",
            "moduleClass": "首页"
          },
          {
            "name": "安全运营",
            "leaf": false
          },
          {
            "editPermission": {
              "modifyEnable": false,
              "authorizeStatus": true
            },
            "viewPermission": {
              "modifyEnable": false,
              "authorizeStatus": true
            },
            "leaf": true,
            "name": "安全运营中心",
            "moduleClass": "安全运营"
          },
          {
            "editPermission": {
              "modifyEnable": false,
              "authorizeStatus": true
            },
            "viewPermission": {
              "modifyEnable": false,
              "authorizeStatus": true
            },
            "leaf": true,
            "name": "业务安全",
            "moduleClass": "安全运营"
          },
          {
            "editPermission": {
              "modifyEnable": false,
              "authorizeStatus": true
            },
            "viewPermission": {
              "modifyEnable": false,
```

```
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "用户安全",
      "moduleClass": "安全运营"
    },
    {
      "editPermission": {
        "modifyEnable": false,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": false,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "勒索专项防护",
      "moduleClass": "安全运营"
    },
    {
      "editPermission": {
        "modifyEnable": false,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": false,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "业务资产管理",
      "moduleClass": "安全运营"
    },
    {
      "editPermission": {
        "modifyEnable": false,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": false,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "黑白名单",
      "moduleClass": "安全运营"
    },
    {
      "editPermission": {
        "modifyEnable": false,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": false,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "热点事件预警",
      "moduleClass": "安全运营"
    },
    {
      "editPermission": {
        "modifyEnable": false,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": false,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "下一代安全体系",
      "moduleClass": "安全运营"
    },
    {
      "name": "监控",
      "leaf": false
    },
    {
      "editPermission": {
```

```
        "modifyEnable": false,
        "authorizeStatus": true
    },
    "viewPermission": {
        "modifyEnable": false,
        "authorizeStatus": true
    },
    "leaf": true,
    "name": "日志",
    "moduleClass": "监控"
},
{
    "editPermission": {
        "modifyEnable": false,
        "authorizeStatus": true
    },
    "viewPermission": {
        "modifyEnable": false,
        "authorizeStatus": true
    },
    "leaf": true,
    "name": "会话",
    "moduleClass": "监控"
},
{
    "editPermission": {
        "modifyEnable": false,
        "authorizeStatus": true
    },
    "viewPermission": {
        "modifyEnable": false,
        "authorizeStatus": true
    },
    "leaf": true,
    "name": "统计",
    "moduleClass": "监控"
},
{
    "editPermission": {
        "modifyEnable": false,
        "authorizeStatus": false
    },
    "viewPermission": {
        "modifyEnable": false,
        "authorizeStatus": true
    },
    "leaf": true,
    "name": "报表",
    "moduleClass": "监控"
},
{
    "editPermission": {
        "modifyEnable": false,
        "authorizeStatus": true
    },
    "viewPermission": {
        "modifyEnable": false,
        "authorizeStatus": true
    },
    "leaf": true,
    "name": "设置",
    "moduleClass": "监控"
},
{
    "name": "策略",
    "leaf": false
},
{
    "editPermission": {
        "modifyEnable": false,
        "authorizeStatus": true
    },
    "viewPermission": {
        "modifyEnable": false,
        "authorizeStatus": true
    },
    "leaf": true,
```

```
{
  "name": "地址转换",
  "moduleClass": "策略"
},
{
  "editPermission": {
    "modifyEnable": false,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": false,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "访问控制",
  "moduleClass": "策略"
},
{
  "editPermission": {
    "modifyEnable": false,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": false,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "安全策略",
  "moduleClass": "策略"
},
{
  "editPermission": {
    "modifyEnable": false,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": false,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "流控",
  "moduleClass": "策略"
},
{
  "editPermission": {
    "modifyEnable": false,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": false,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "认证",
  "moduleClass": "策略"
},
{
  "editPermission": {
    "modifyEnable": false,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": false,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "解密",
  "moduleClass": "策略"
},
{
  "editPermission": {
    "modifyEnable": false,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": false,
    "authorizeStatus": true
  }
}
```

```
    "leaf": true,
    "name": "页面定制",
    "moduleClass": "策略"
  },
  {
    "name": "对象",
    "leaf": false
  },
  {
    "editPermission": {
      "modifyEnable": false,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": false,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "网络对象",
    "moduleClass": "对象"
  },
  {
    "editPermission": {
      "modifyEnable": false,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": false,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "服务",
    "moduleClass": "对象"
  },
  {
    "editPermission": {
      "modifyEnable": false,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": false,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "安全策略模板",
    "moduleClass": "对象"
  },
  {
    "editPermission": {
      "modifyEnable": false,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": false,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "安全防护规则库",
    "moduleClass": "对象"
  },
  {
    "editPermission": {
      "modifyEnable": false,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": false,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "内容识别库",
    "moduleClass": "对象"
  },
  {
    "editPermission": {
      "modifyEnable": false,
      "authorizeStatus": true
```



```

    },
    "viewPermission": {
      "modifyEnable": false,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "IP地址库",
    "moduleClass": "对象"
  },
  {
    "editPermission": {
      "modifyEnable": false,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": false,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "时间计划",
    "moduleClass": "对象"
  },
  {
    "editPermission": {
      "modifyEnable": false,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": false,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "信任的证书颁发机构",
    "moduleClass": "对象"
  },
  {
    "name": "网络",
    "leaf": false
  },
  {
    "editPermission": {
      "modifyEnable": false,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": false,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "网络",
    "moduleClass": "网络"
  },
  {
    "name": "系统",
    "leaf": false
  },
  {
    "editPermission": {
      "modifyEnable": false,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": false,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "系统",
    "moduleClass": "系统"
  }
},
"description": "Administrator",
"name": "admin",
"roleName": "ADMINISTRATOR"
},
{
  "enable": true,
  "authentication": {

```

```
"usbKeyEnable": false,
"fingerprintEnable": false,
"manageMode": [
  "APIINTERFACE"
],
},
"modulePermissions": [
  {
    "name": "首页",
    "leaf": false
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "首页",
    "moduleClass": "首页"
  },
  {
    "name": "安全运营",
    "leaf": false
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "安全运营中心",
    "moduleClass": "安全运营"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "业务安全",
    "moduleClass": "安全运营"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "用户安全",
    "moduleClass": "安全运营"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
```

```

    },
    {
      "name": "勒索专项防护",
      "moduleClass": "安全运营",
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "业务资产管理",
      "moduleClass": "安全运营"
    },
    {
      "name": "黑白名单",
      "moduleClass": "安全运营",
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "热点事件预警",
      "moduleClass": "安全运营"
    },
    {
      "name": "下一代安全体系",
      "moduleClass": "安全运营",
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "下一代安全体系",
      "moduleClass": "安全运营"
    },
    {
      "name": "监控",
      "leaf": false,
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "日志",
      "moduleClass": "监控"
    },
    {
      "name": "监控",
      "leaf": false,
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "日志",
      "moduleClass": "监控"
    }
  ],
  "name": "安全运营",
  "moduleClass": "安全运营",
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "安全运营",
  "moduleClass": "安全运营"
}

```

```
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "会话",
    "moduleClass": "监控"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "统计",
    "moduleClass": "监控"
  },
  {
    "editPermission": {
      "modifyEnable": false,
      "authorizeStatus": false
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "报表",
    "moduleClass": "监控"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "设置",
    "moduleClass": "监控"
  },
  {
    "name": "策略",
    "leaf": false
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "地址转换",
    "moduleClass": "策略"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "访问控制",
    "moduleClass": "策略"
  },
}
```

```
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "安全策略",
  "moduleClass": "策略"
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "流控",
  "moduleClass": "策略"
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "认证",
  "moduleClass": "策略"
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "解密",
  "moduleClass": "策略"
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "页面定制",
  "moduleClass": "策略"
},
{
  "name": "对象",
  "leaf": false
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  }
}
```

```
    },
    "leaf": true,
    "name": "网络对象",
    "moduleClass": "对象"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "服务",
    "moduleClass": "对象"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "安全策略模板",
    "moduleClass": "对象"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "安全防护规则库",
    "moduleClass": "对象"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "内容识别库",
    "moduleClass": "对象"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "IP地址库",
    "moduleClass": "对象"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
```

```

        "authorizeStatus": true
      },
      "leaf": true,
      "name": "时间计划",
      "moduleClass": "对象"
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "信任的证书颁发机构",
      "moduleClass": "对象"
    },
    {
      "name": "网络",
      "leaf": false
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "网络",
      "moduleClass": "网络"
    },
    {
      "name": "系统",
      "leaf": false
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "系统",
      "moduleClass": "系统"
    }
  ],
  "description": "1",
  "name": "test1",
  "roleName": "COMMON"
},
{
  "enable": true,
  "authentication": {
    "usbKeyEnable": false,
    "fingerprintEnable": false,
    "manageMode": [
      "APIINTERFACE"
    ]
  },
  "modulePermissions": [
    {
      "name": "首页",
      "leaf": false
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },

```

```
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "首页",
    "moduleClass": "首页"
  },
  {
    "name": "安全运营",
    "leaf": false
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "安全运营中心",
    "moduleClass": "安全运营"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "业务安全",
    "moduleClass": "安全运营"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "用户安全",
    "moduleClass": "安全运营"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "勒索专项防护",
    "moduleClass": "安全运营"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "业务资产管理",
    "moduleClass": "安全运营"
  },
}
```



```
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "黑白名单",
  "moduleClass": "安全运营"
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "热点事件预警",
  "moduleClass": "安全运营"
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "下一代安全体系",
  "moduleClass": "安全运营"
},
{
  "name": "监控",
  "leaf": false
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "日志",
  "moduleClass": "监控"
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "会话",
  "moduleClass": "监控"
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  }
}
```

```
    },
    "leaf": true,
    "name": "统计",
    "moduleClass": "监控"
  },
  {
    "editPermission": {
      "modifyEnable": false,
      "authorizeStatus": false
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "报表",
    "moduleClass": "监控"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "设置",
    "moduleClass": "监控"
  },
  {
    "name": "策略",
    "leaf": false
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "地址转换",
    "moduleClass": "策略"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "访问控制",
    "moduleClass": "策略"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "安全策略",
    "moduleClass": "策略"
  },
  {
    "editPermission": {
      "modifyEnable": true,
```

```

        "authorizeStatus": true
    },
    "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "leaf": true,
    "name": "流控",
    "moduleClass": "策略"
},
{
    "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "leaf": true,
    "name": "认证",
    "moduleClass": "策略"
},
{
    "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "leaf": true,
    "name": "解密",
    "moduleClass": "策略"
},
{
    "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "leaf": true,
    "name": "页面定制",
    "moduleClass": "策略"
},
{
    "name": "对象",
    "leaf": false
},
{
    "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "leaf": true,
    "name": "网络对象",
    "moduleClass": "对象"
},
{
    "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "leaf": true,
    "name": "服务",

```

```
    "moduleClass": "对象"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "安全策略模板",
    "moduleClass": "对象"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "安全防护规则库",
    "moduleClass": "对象"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "内容识别库",
    "moduleClass": "对象"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "IP地址库",
    "moduleClass": "对象"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "时间计划",
    "moduleClass": "对象"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
```

```

        "name": "信任的证书颁发机构",
        "moduleClass": "对象"
    },
    {
        "name": "网络",
        "leaf": false
    },
    {
        "editPermission": {
            "modifyEnable": true,
            "authorizeStatus": true
        },
        "viewPermission": {
            "modifyEnable": true,
            "authorizeStatus": true
        },
        "leaf": true,
        "name": "网络",
        "moduleClass": "网络"
    },
    {
        "name": "系统",
        "leaf": false
    },
    {
        "editPermission": {
            "modifyEnable": true,
            "authorizeStatus": true
        },
        "viewPermission": {
            "modifyEnable": true,
            "authorizeStatus": true
        },
        "leaf": true,
        "name": "系统",
        "moduleClass": "系统"
    }
],
"description": "1",
"name": "test2",
"roleName": "COMMON"
}
],
"totalItems": 4,
"pageNumber": 0,
"itemLength": 3,
"totalPages": 2,
"pageSize": 3
},
"code": 0,
"message": "成功"
}

```

请求失败样例:

```

{
  "data": "",
  "code": 22,
  "message": "无效参数"
}

```

5.1.2 查询一个已创建的管理员账户信息

功能说明: 通过本接口, 可以查询一个已创建的管理员账户信息

1) API请求数据

GET	https://{host}/api/v1/namespaces/@namespace/account/@name
-----	---

2) 输入参数

- URL参数

参数	类型	是否必选	描述
role	string	否	账户角色，主要用于判断是否为远程账户 枚举值: ADMINISTRATOR:超级管理员 COMMON:普通管理员 SAFE:三权分立账户，安全管理员 LOG:三权分立账户，审计员 SYSTEM:三权分立账户，系统管理员 REMOTE:远程认证账户]

请求样例：

查询一个管理员账户信息，管理员角色为远程认证账户

GET https://192.168.1.1/api/v1/namespaces/public/account/test?role=REMOTE

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	管理员基本信息
data.name	string	用户名 最小长度:1 最大长度:60
data.enable	bool	账户状态，启用/禁用
data.description	string	描述 最小长度:0 最大长度:95
data.roleName	string	账户角色 枚举值: ADMINISTRATOR:超级管理员 COMMON:普通管理员 SAFE:三权分立账户，安全管理员 LOG:三权分立账户，审计员 SYSTEM:三权分立账户，系统管理员 REMOTE:远程认证账户]
data.authentication	object	认证策略
data.authentication.usbKeyEnable	bool	usb-key认证是否启用
data.authentication.fingerPrintEnable	bool	指纹校验是否启用
data.authentication.createUsbKeyStatus	bool	是否创建了usbkey证书

data.authentication.dKey	string	采集的指纹 最大长度:24
data.authentication.manageMode	string[]	管理方式
data.modulePermissions	object[]	模块权限
data.modulePermissions.name	string	模块名 最大长度:60
data.modulePermissions.leaf	bool	模块是否为子项
data.modulePermissions.moduleClass	string	模块所属大类名 最大长度:60
data.modulePermissions.editPermission	object	模块的编辑权限
data.modulePermissions.editPermission.authorizeStatus	bool	是否有权限
data.modulePermissions.editPermission.modifyEnable	bool	能否修改权限
data.modulePermissions.viewPermission	object	模块的查看权限
data.modulePermissions.viewPermission.authorizeStatus	bool	是否有权限
data.modulePermissions.viewPermission.modifyEnable	bool	能否修改权限

请求成功样例:

```
{
  "data": {
    "enable": true,
    "authentication": {
      "usbKeyEnable": false,
      "fingerprintEnable": false,
      "manageMode": [
        "WEBCONSOLE"
      ]
    }
  },
  "modulePermissions": [
    {
      "name": "首页",
      "leaf": false
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "首页",
      "moduleClass": "首页"
    }
  ],
}
```

```
{
  "name": "安全运营",
  "leaf": false
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "安全运营中心",
  "moduleClass": "安全运营"
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "业务安全",
  "moduleClass": "安全运营"
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "用户安全",
  "moduleClass": "安全运营"
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "勒索专项防护",
  "moduleClass": "安全运营"
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "业务资产管理",
  "moduleClass": "安全运营"
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "业务资产管理",
  "moduleClass": "安全运营"
```



```
    },
    "leaf": true,
    "name": "黑白名单",
    "moduleClass": "安全运营"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "热点事件预警",
    "moduleClass": "安全运营"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "下一代安全体系",
    "moduleClass": "安全运营"
  },
  {
    "name": "监控",
    "leaf": false
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "日志",
    "moduleClass": "监控"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "会话",
    "moduleClass": "监控"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "统计",
    "moduleClass": "监控"
  },
  {
    "editPermission": {
      "modifyEnable": false,
```

```
        "authorizeStatus": false
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "报表",
      "moduleClass": "监控"
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "设置",
      "moduleClass": "监控"
    },
    {
      "name": "策略",
      "leaf": false
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "地址转换",
      "moduleClass": "策略"
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "访问控制",
      "moduleClass": "策略"
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "安全策略",
      "moduleClass": "策略"
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "流控",
```

```
    "moduleClass": "策略"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "认证",
    "moduleClass": "策略"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "解密",
    "moduleClass": "策略"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "页面定制",
    "moduleClass": "策略"
  },
  {
    "name": "对象",
    "leaf": false
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "网络对象",
    "moduleClass": "对象"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "服务",
    "moduleClass": "对象"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
```

```

        "modifyEnable": true,
        "authorizeStatus": true
    },
    "leaf": true,
    "name": "安全策略模板",
    "moduleClass": "对象"
},
{
    "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "leaf": true,
    "name": "安全防护规则库",
    "moduleClass": "对象"
},
{
    "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "leaf": true,
    "name": "内容识别库",
    "moduleClass": "对象"
},
{
    "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "leaf": true,
    "name": "IP地址库",
    "moduleClass": "对象"
},
{
    "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "leaf": true,
    "name": "时间计划",
    "moduleClass": "对象"
},
{
    "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "leaf": true,
    "name": "信任的证书颁发机构",
    "moduleClass": "对象"
},
{
    "name": "网络",
    "leaf": false
},
{
    "name": "网络",
    "leaf": false
}

```

```
        "editPermission": {
            "modifyEnable": true,
            "authorizeStatus": true
        },
        "viewPermission": {
            "modifyEnable": true,
            "authorizeStatus": true
        },
        "leaf": true,
        "name": "网络",
        "moduleClass": "网络"
    },
    {
        "name": "系统",
        "leaf": false
    },
    {
        "editPermission": {
            "modifyEnable": true,
            "authorizeStatus": true
        },
        "viewPermission": {
            "modifyEnable": true,
            "authorizeStatus": true
        },
        "leaf": true,
        "name": "系统",
        "moduleClass": "系统"
    }
],
"description": "remote",
"name": "test",
"roleName": "REMOTE"
},
"code": 0,
"message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

5.1.3 查询管理员账户的不同角色的默认权限模板

功能说明：通过本接口，可以查询管理员账户的不同角色的默认权限模板，用于设置管理员账户权限

1) API请求数据

GET

https://{host}/api/v1/namespaces/@namespace/modulepermissiontpl

2) 输入参数

- URL 参数

参数	类型	是否必选	描述
role	string	否	账户角色，主要用于判断是否为远程账户 枚举值:[ADMINISTRATOR:超级管理员 COMMON:普通管理员 SAFE:三权分立账户，安全管理员 LOG:三权分立账户，审计员 SYSTEM:三权分立账户，系统管理员 REMOTE:远程认证账户]

请求样例:

查询管理员账户默认权限模板, 管理员角色为普通用户

GET https://192.168.1.1/api/v1/namespaces/public/modulepermissiontpl?role=COMMON

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	模块权限模板
data.role	string	账户角色 枚举值:[ADMINISTRATOR:超级管理员 COMMON:普通管理员 SAFE:三权分立账户, 安全管理员 LOG:三权分立账户, 审计员 SYSTEM:三权分立账户, 系统管理员 REMOTE:远程认证账户]
data.modulePermissions	object[]	模块权限
data.modulePermissions.name	string	模块名 最大长度:60
data.modulePermissions.leaf	bool	模块是否为子项
data.modulePermissions.moduleClass	string	模块所属大类名 最大长度:60
data.modulePermissions.editPermission	object	模块的编辑权限
data.modulePermissions.editPermission.authorizeStatus	bool	是否有权限
data.modulePermissions.editPermission.modifyEnable	bool	能否修改权限
data.modulePermissions.viewPermission	object	模块的查看权限
data.modulePermissions.viewPermission.authorizeStatus	bool	是否有权限
data.modulePermissions.viewPermission.modifyEnable	bool	能否修改权限

请求成功样例:

```
{
  "data": {
```

```
"modulePermissions": [
  {
    "name": "首页",
    "leaf": false
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "首页",
    "moduleClass": "首页"
  },
  {
    "name": "安全运营",
    "leaf": false
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "安全运营中心",
    "moduleClass": "安全运营"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "业务安全",
    "moduleClass": "安全运营"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "用户安全",
    "moduleClass": "安全运营"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "勒索专项防护",
    "moduleClass": "安全运营"
  },
  {
    "editPermission": {
      "modifyEnable": true,
```

```
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "业务资产管理",
      "moduleClass": "安全运营"
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "黑白名单",
      "moduleClass": "安全运营"
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "热点事件预警",
      "moduleClass": "安全运营"
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "下一代安全体系",
      "moduleClass": "安全运营"
    },
    {
      "name": "监控",
      "leaf": false
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "日志",
      "moduleClass": "监控"
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "会话",
```



```
    "moduleClass": "监控"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "统计",
    "moduleClass": "监控"
  },
  {
    "editPermission": {
      "modifyEnable": false,
      "authorizeStatus": false
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "报表",
    "moduleClass": "监控"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "设置",
    "moduleClass": "监控"
  },
  {
    "name": "策略",
    "leaf": false
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "地址转换",
    "moduleClass": "策略"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "访问控制",
    "moduleClass": "策略"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
```

```
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "leaf": true,
    "name": "安全策略",
    "moduleClass": "策略"
},
{
    "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "leaf": true,
    "name": "流控",
    "moduleClass": "策略"
},
{
    "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "leaf": true,
    "name": "认证",
    "moduleClass": "策略"
},
{
    "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "leaf": true,
    "name": "解密",
    "moduleClass": "策略"
},
{
    "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "leaf": true,
    "name": "页面定制",
    "moduleClass": "策略"
},
{
    "name": "对象",
    "leaf": false
},
{
    "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "viewPermission": {
        "modifyEnable": false,
        "authorizeStatus": true
    },
    "leaf": true,
    "name": "网络对象",
    "moduleClass": "对象"
},
{
    }
```

```
"editPermission": {
  "modifyEnable": true,
  "authorizeStatus": true
},
"viewPermission": {
  "modifyEnable": false,
  "authorizeStatus": true
},
"leaf": true,
"name": "服务",
"moduleClass": "对象"
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": false,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "安全策略模板",
  "moduleClass": "对象"
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": false,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "安全防护规则库",
  "moduleClass": "对象"
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": false,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "内容识别库",
  "moduleClass": "对象"
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": false,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "IP地址库",
  "moduleClass": "对象"
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": false,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "时间计划",
  "moduleClass": "对象"
},
},
```

```

{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": false,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "信任的证书颁发机构",
  "moduleClass": "对象"
},
{
  "name": "网络",
  "leaf": false
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "网络",
  "moduleClass": "网络"
},
{
  "name": "系统",
  "leaf": false
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "系统",
  "moduleClass": "系统"
}
],
"role": "COMMON"
},
"code": 0,
"message": "成功"
}

```

请求失败样例:

```

{
  "data": "",
  "code": 22,
  "message": "无效参数"
}

```

5.1.4 新增一个管理员账户

功能说明: 通过本接口, 可以新增一个管理员账户

1) API请求数据

POST <https://{host}/api/v1/namespaces/@namespace/account>

2) 输入参数

• Body参数

参数	类型	是否必选	描述
obj	object	是	管理员基本信息
obj.name	string	是	用户名 最小长度:1 最大长度:60
obj.enable	bool	是	账户状态, 启用/禁用 默认值:true
obj.description	string	否	描述 最小长度:0 最大长度:95
obj.roleName	string	是	账户角色 枚举值:[ADMINISTRATOR:超级管理员 COMMON:普通管理员 SAFE:三权分立账户, 安全管理员 LOG:三权分立账户, 审计员 SYSTEM:三权分立账户, 系统管理员 REMOTE:远程认证账户]
obj.authentication	object	是	认证策略
obj.authentication.password	string	是	新密码, 创建或修改其他账户密码时使用 最小长度:1 最大长度:512
obj.authentication.manageMode	string[]	是	管理方式
obj.modulePermissions	object[]	是	模块权限
obj.modulePermissions.name	string	是	模块名 最大长度:60
obj.modulePermissions.leaf	bool	是	模块是否为子项
obj.modulePermissions.moduleClass	string	是	模块所属大类名 最大长度:60
obj.modulePermissions.editPermission	object	是	模块的编辑权限
obj.modulePermissions.editPermission.authorizeStatus	bool	是	是否有权限
obj.modulePermissions.editPermission.modifyEnable	bool	是	能否修改权限
obj.modulePermissions.viewPermission	object	是	模块的查看权限

obj.modulePermissions.viewPermission.authorizeStatus	bool	是	是否有权限
obj.modulePermissions.viewPermission.modifyEnable	bool	是	能否修改权限

请求样例:

添加普通管理员账户, 管理方式只有WEB控制台

POST https://192.168.1.1/api/v1/namespaces/public/account

```
{
  "enable": true,
  "authentication": {
    "password": "sangfor123.",
    "manageMode": [
      "WEBCONSOLE"
    ]
  },
  "modulePermissions": [
    {
      "name": "首页",
      "leaf": false
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": false
      },
      "leaf": true,
      "name": "首页",
      "moduleClass": "首页"
    },
    {
      "name": "安全运营",
      "leaf": false
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "安全运营中心",
      "moduleClass": "安全运营"
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "业务安全",
      "moduleClass": "安全运营"
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {

```

```
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "leaf": true,
    "name": "用户安全",
    "moduleClass": "安全运营"
},
{
    "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "leaf": true,
    "name": "勒索专项防护",
    "moduleClass": "安全运营"
},
{
    "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "leaf": true,
    "name": "业务资产管理",
    "moduleClass": "安全运营"
},
{
    "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "leaf": true,
    "name": "黑白名单",
    "moduleClass": "安全运营"
},
{
    "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "leaf": true,
    "name": "热点事件预警",
    "moduleClass": "安全运营"
},
{
    "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "leaf": true,
    "name": "下一代安全体系",
    "moduleClass": "安全运营"
},
{
    "name": "监控",
    "leaf": false
},
{
    "name": "安全运营",
    "leaf": true,
    "moduleClass": "安全运营"
}
```

```
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "日志",
    "moduleClass": "监控"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "会话",
    "moduleClass": "监控"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "统计",
    "moduleClass": "监控"
  },
  {
    "editPermission": {
      "modifyEnable": false,
      "authorizeStatus": false
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "报表",
    "moduleClass": "监控"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "设置",
    "moduleClass": "监控"
  },
  {
    "name": "策略",
    "leaf": false
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    }
  },
}
```



```
"leaf": true,
"name": "地址转换",
"moduleClass": "策略"
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "访问控制",
  "moduleClass": "策略"
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "安全策略",
  "moduleClass": "策略"
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "流控",
  "moduleClass": "策略"
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "认证",
  "moduleClass": "策略"
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "解密",
  "moduleClass": "策略"
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "解密",
  "moduleClass": "策略"
}
```

```
    },
    "leaf": true,
    "name": "页面定制",
    "moduleClass": "策略"
  },
  {
    "name": "对象",
    "leaf": false
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": false,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "网络对象",
    "moduleClass": "对象"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": false,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "服务",
    "moduleClass": "对象"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": false,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "安全策略模板",
    "moduleClass": "对象"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": false,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "安全防护规则库",
    "moduleClass": "对象"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": false,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "内容识别库",
    "moduleClass": "对象"
  },
  {
    "editPermission": {
      "modifyEnable": true,
```

```
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": false,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "IP地址库",
      "moduleClass": "对象"
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": false,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "时间计划",
      "moduleClass": "对象"
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": false,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "信任的证书颁发机构",
      "moduleClass": "对象"
    },
    {
      "name": "网络",
      "leaf": false
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "网络",
      "moduleClass": "网络"
    },
    {
      "name": "系统",
      "leaf": false
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "系统",
      "moduleClass": "系统"
    }
  ],
  "description": "",
  "name": "111",
  "roleName": "COMMON"
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	管理员基本信息
data.name	string	用户名 最小长度:1 最大长度:60
data.enable	bool	账户状态, 启用/禁用
data.description	string	描述 最小长度:0 最大长度:95
data.roleName	string	账户角色 枚举值: ADMINISTRATOR:超级管理员 COMMON:普通管理员 SAFE:三权分立账户, 安全管理员 LOG:三权分立账户, 审计员 SYSTEM:三权分立账户, 系统管理员 REMOTE:远程认证账户]
data.authentication	object	认证策略
data.authentication.usbKeyEnable	bool	usb-key认证是否启用
data.authentication.fingerPrintEnable	bool	指纹校验是否启用
data.authentication.createUsbKeyStatus	bool	是否创建了usbkey证书
data.authentication.dKey	string	采集的指纹 最大长度:24
data.authentication.manageMode	string[]	管理方式
data.modulePermissions	object[]	模块权限
data.modulePermissions.name	string	模块名 最大长度:60
data.modulePermissions.leaf	bool	模块是否为子项
data.modulePermissions.moduleClass	string	模块所属大类名 最大长度:60
data.modulePermissions.editPermission	object	模块的编辑权限

data.modulePermissions.editPermission.authorizeStatus	bool	是否有权限
data.modulePermissions.editPermission.modifyEnable	bool	能否修改权限
data.modulePermissions.viewPermission	object	模块的查看权限
data.modulePermissions.viewPermission.authorizeStatus	bool	是否有权限
data.modulePermissions.viewPermission.modifyEnable	bool	能否修改权限

请求成功样例：

```
{
  "data": {
    "enable": true,
    "authentication": {
      "usbKeyEnable": false,
      "fingerprintEnable": false,
      "manageMode": [
        "WEBCONSOLE"
      ]
    },
    "modulePermissions": [
      {
        "name": "首页",
        "leaf": false
      },
      {
        "editPermission": {
          "modifyEnable": true,
          "authorizeStatus": true
        },
        "viewPermission": {
          "modifyEnable": true,
          "authorizeStatus": true
        },
        "leaf": true,
        "name": "首页",
        "moduleClass": "首页"
      },
      {
        "name": "安全运营",
        "leaf": false
      },
      {
        "editPermission": {
          "modifyEnable": true,
          "authorizeStatus": true
        },
        "viewPermission": {
          "modifyEnable": true,
          "authorizeStatus": true
        },
        "leaf": true,
        "name": "安全运营中心",
        "moduleClass": "安全运营"
      },
      {
        "editPermission": {
          "modifyEnable": true,
          "authorizeStatus": true
        },
        "viewPermission": {
          "modifyEnable": true,
```

```
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "业务安全",
      "moduleClass": "安全运营"
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "用户安全",
      "moduleClass": "安全运营"
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "勒索专项防护",
      "moduleClass": "安全运营"
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "业务资产管理",
      "moduleClass": "安全运营"
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "黑白名单",
      "moduleClass": "安全运营"
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "热点事件预警",
      "moduleClass": "安全运营"
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
```

```

    "modifyEnable": true,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "下一代安全体系",
  "moduleClass": "安全运营"
},
{
  "name": "监控",
  "leaf": false
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "日志",
  "moduleClass": "监控"
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "会话",
  "moduleClass": "监控"
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "统计",
  "moduleClass": "监控"
},
{
  "editPermission": {
    "modifyEnable": false,
    "authorizeStatus": false
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "报表",
  "moduleClass": "监控"
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "设置",
  "moduleClass": "监控"
},
{

```



```
    "leaf": true,
    "name": "解密",
    "moduleClass": "策略"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "页面定制",
    "moduleClass": "策略"
  },
  {
    "name": "对象",
    "leaf": false
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "网络对象",
    "moduleClass": "对象"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "服务",
    "moduleClass": "对象"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "安全策略模板",
    "moduleClass": "对象"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "安全防护规则库",
    "moduleClass": "对象"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
```

```
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "内容识别库",
    "moduleClass": "对象"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "IP地址库",
    "moduleClass": "对象"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "时间计划",
    "moduleClass": "对象"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "信任的证书颁发机构",
    "moduleClass": "对象"
  },
  {
    "name": "网络",
    "leaf": false
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "网络",
    "moduleClass": "网络"
  },
  {
    "name": "系统",
    "leaf": false
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    }
```

```

    },
    "leaf": true,
    "name": "系统",
    "moduleClass": "系统"
  }
},
"description": "",
"name": "111",
"roleName": "COMMON"
},
"code": 0,
"message": "成功"
}

```

请求失败样例：

```

{
  "data": "",
  "code": 22,
  "message": "无效参数"
}

```

5.1.5 修改一个管理员账户的信息

功能说明：通过本接口，可以修改一个管理员账户的信息，包括密码、描述、模块权限、启用禁用状态和登录管理方式

1) API请求数据

PATCH <https://{host}/api/v1/namespaces/@namespace/account/@name>

2) 输入参数

• Body参数

参数	类型	是否必选	描述
obj	object	是	管理员基本信息
obj.name	string	是	用户名 最小长度:1 最大长度:60
obj.enable	bool	否	账户状态，启用/禁用 默认值:true
obj.description	string	否	描述 最小长度:0 最大长度:95
obj.roleName	string	是	账户角色 枚举值:[ADMINISTRATOR:超级管理员 COMMON:普通管理员 SAFE:三权分立账户，安全管理员 LOG:三权分立账户，审计员 SYSTEM:三权分立账户，系统管理员 REMOTE:远程认证账户]
obj.authentication	object	否	认证策略
obj.authentication.oldPassword	string	否	当前密码，修改本账户密码，或者修改默认账户用户名及密码时使用 最小长度:1 最大长度:512

obj.authentication.pass word	string	否	新密码，创建或修改其他账户密码时使用 最小长度:1 最大长度:512
obj.authentication.manageMode	string[]	否	管理方式
obj.modulePermissions	object[]	否	模块权限
obj.modulePermissions.name	string	是	模块名 最大长度:60
obj.modulePermissions.leaf	bool	是	模块是否为子项
obj.modulePermissions.moduleClass	string	是	模块所属大类名 最大长度:60
obj.modulePermissions.editPermission	object	是	模块的编辑权限
obj.modulePermissions.editPermission.authorizeStatus	bool	是	是否有权限
obj.modulePermissions.editPermission.modifyEnable	bool	是	能否修改权限
obj.modulePermissions.viewPermission	object	是	模块的查看权限
obj.modulePermissions.viewPermission.authorizeStatus	bool	是	是否有权限
obj.modulePermissions.viewPermission.modifyEnable	bool	是	能否修改权限

请求样例:

修改角色为普通用户的管理员账户信息，修改描述字段为test，修改首页模块权限为不勾选（模块权限字段需要传入整个结构）

PATCH https://192.168.1.1/api/v1/namespaces/public/account/111

```
{
  "enable": true,
  "modulePermissions": [
    {
      "name": "首页",
      "leaf": false
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": false
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": false
      },
      "leaf": true,
      "name": "首页",
      "moduleClass": "首页"
    },
    {
      "name": "安全运营",
      "leaf": false
    }
  ],
}
```

```
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "安全运营中心",
  "moduleClass": "安全运营"
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "业务安全",
  "moduleClass": "安全运营"
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "用户安全",
  "moduleClass": "安全运营"
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "勒索专项防护",
  "moduleClass": "安全运营"
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "业务资产管理",
  "moduleClass": "安全运营"
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "黑白名单",
  "moduleClass": "安全运营"
}
```

```
{
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "热点事件预警",
    "moduleClass": "安全运营"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "下一代安全体系",
    "moduleClass": "安全运营"
  },
  {
    "name": "监控",
    "leaf": false
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "日志",
    "moduleClass": "监控"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "会话",
    "moduleClass": "监控"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "统计",
    "moduleClass": "监控"
  },
  {
    "editPermission": {
      "modifyEnable": false,
      "authorizeStatus": false
    },
    "viewPermission": {
      "modifyEnable": true,
```

```
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "报表",
      "moduleClass": "监控"
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "设置",
      "moduleClass": "监控"
    },
    {
      "name": "策略",
      "leaf": false
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "地址转换",
      "moduleClass": "策略"
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "访问控制",
      "moduleClass": "策略"
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "安全策略",
      "moduleClass": "策略"
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "流控",
      "moduleClass": "策略"
    },
    {
      "editPermission": {
```

```
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "leaf": true,
    "name": "认证",
    "moduleClass": "策略"
},
{
    "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "leaf": true,
    "name": "解密",
    "moduleClass": "策略"
},
{
    "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "leaf": true,
    "name": "页面定制",
    "moduleClass": "策略"
},
{
    "name": "对象",
    "leaf": false
},
{
    "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "leaf": true,
    "name": "网络对象",
    "moduleClass": "对象"
},
{
    "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "leaf": true,
    "name": "服务",
    "moduleClass": "对象"
},
{
    "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "leaf": true,
```



```
    "name": "安全策略模板",
    "moduleClass": "对象"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "安全防护规则库",
    "moduleClass": "对象"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "内容识别库",
    "moduleClass": "对象"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "IP地址库",
    "moduleClass": "对象"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "时间计划",
    "moduleClass": "对象"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "信任的证书颁发机构",
    "moduleClass": "对象"
  },
  {
    "name": "网络",
    "leaf": false
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
```

```
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "网络",
    "moduleClass": "网络"
  },
  {
    "name": "系统",
    "leaf": false
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "系统",
    "moduleClass": "系统"
  }
],
"description": "test",
"name": "111",
"roleName": "COMMON"
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	管理员基本信息
data.name	string	用户名 最小长度:1 最大长度:60
data.enable	bool	账户状态，启用/禁用
data.description	string	描述 最小长度:0 最大长度:95
data.roleName	string	账户角色 枚举值: ADMINISTRATOR:超级管理员 COMMON:普通管理员 SAFE:三权分立账户，安全管理员 LOG:三权分立账户，审计员 SYSTEM:三权分立账户，系统管理员 REMOTE:远程认证账户]
data.authentication	object	认证策略
data.authentication.usb KeyEnable	bool	usb-key认证是否启用
data.authentication.fing erprintEnable	bool	指纹校验是否启用

data.authentication.createUsbKeyStatus	bool	是否创建了usbkey证书
data.authentication.dKey	string	采集的指纹 最大长度:24
data.authentication.manageMode	string[]	管理方式
data.modulePermissions	object[]	模块权限
data.modulePermissions.name	string	模块名 最大长度:60
data.modulePermissions.leaf	bool	模块是否为子项
data.modulePermissions.moduleClass	string	模块所属大类名 最大长度:60
data.modulePermissions.editPermission	object	模块的编辑权限
data.modulePermissions.editPermission.authorizeStatus	bool	是否有权限
data.modulePermissions.editPermission.modifyEnable	bool	能否修改权限
data.modulePermissions.viewPermission	object	模块的查看权限
data.modulePermissions.viewPermission.authorizeStatus	bool	是否有权限
data.modulePermissions.viewPermission.modifyEnable	bool	能否修改权限

请求成功样例:

```
{
  "data": {
    "enable": true,
    "authentication": {
      "usbKeyEnable": false,
      "fingerprintEnable": false,
      "manageMode": [
        "WEBCONSOLE"
      ]
    },
    "modulePermissions": [
      {
        "name": "首页",
        "leaf": false
      },
      {
        "editPermission": {
          "modifyEnable": true,
          "authorizeStatus": false
        },
        "viewPermission": {
          "modifyEnable": true,
          "authorizeStatus": false
        },
        "leaf": true,

```

```
{
  "name": "首页",
  "moduleClass": "首页"
},
{
  "name": "安全运营",
  "leaf": false
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "安全运营中心",
  "moduleClass": "安全运营"
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "业务安全",
  "moduleClass": "安全运营"
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "用户安全",
  "moduleClass": "安全运营"
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "勒索专项防护",
  "moduleClass": "安全运营"
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "业务资产管理",
  "moduleClass": "安全运营"
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  }
}
```

```
"viewPermission": {
  "modifyEnable": true,
  "authorizeStatus": true
},
"leaf": true,
"name": "黑白名单",
"moduleClass": "安全运营"
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "热点事件预警",
  "moduleClass": "安全运营"
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "下一代安全体系",
  "moduleClass": "安全运营"
},
{
  "name": "监控",
  "leaf": false
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "日志",
  "moduleClass": "监控"
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "会话",
  "moduleClass": "监控"
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "统计",
  "moduleClass": "监控"
},
},
}
```

```
{
  "editPermission": {
    "modifyEnable": false,
    "authorizeStatus": false
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "报表",
  "moduleClass": "监控"
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "设置",
  "moduleClass": "监控"
},
{
  "name": "策略",
  "leaf": false
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "地址转换",
  "moduleClass": "策略"
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "访问控制",
  "moduleClass": "策略"
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "安全策略",
  "moduleClass": "策略"
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "安全策略",
  "moduleClass": "策略"
}
```

```

    },
    "leaf": true,
    "name": "流控",
    "moduleClass": "策略"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "认证",
    "moduleClass": "策略"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "解密",
    "moduleClass": "策略"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "页面定制",
    "moduleClass": "策略"
  },
  {
    "name": "对象",
    "leaf": false
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "网络对象",
    "moduleClass": "对象"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "服务",
    "moduleClass": "对象"
  },
  {
    "editPermission": {
      "modifyEnable": true,

```

```
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "安全策略模板",
      "moduleClass": "对象"
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "安全防护规则库",
      "moduleClass": "对象"
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "内容识别库",
      "moduleClass": "对象"
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "IP地址库",
      "moduleClass": "对象"
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "时间计划",
      "moduleClass": "对象"
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "信任的证书颁发机构",
      "moduleClass": "对象"
    },
    {
      "name": "网络",
```



```
    "leaf": false
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "网络",
    "moduleClass": "网络"
  },
  {
    "name": "系统",
    "leaf": false
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "系统",
    "moduleClass": "系统"
  },
  ],
  "description": "test",
  "name": "111",
  "roleName": "COMMON"
},
"code": 0,
"message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

5.1.6 删除一个管理员账户

功能说明: 通过本接口, 可以删除一个管理员账户

1) API请求数据

POST

https://{host}/api/v1/namespaces/@namespace/account?_method=delete

2) 输入参数

- Body参数

参数	类型	是否必选	描述
obj	object	是	管理员基本信息
obj.name	string	是	用户名 最小长度:1 最大长度:60

obj.roleName	string	是	账户角色 枚举值: ADMINISTRATOR:超级管理员 COMMON:普通管理员 SAFE:三权分立账户, 安全管理员 LOG:三权分立账户, 审计员 SYSTEM:三权分立账户, 系统管理员 REMOTE:远程认证账户]
--------------	--------	---	--

请求样例:

删除一个管理员账户, 管理员角色为远程认证账户

```
POST https://192.168.1.1/api/v1/namespaces/public/account?_method=DELETE
{
  "name": "test",
  "roleName": "REMOTE"
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	管理员基本信息
data.name	string	用户名 最小长度:1 最大长度:60
data.enable	bool	账户状态, 启用/禁用
data.description	string	描述 最小长度:0 最大长度:95
data.roleName	string	账户角色 枚举值: ADMINISTRATOR:超级管理员 COMMON:普通管理员 SAFE:三权分立账户, 安全管理员 LOG:三权分立账户, 审计员 SYSTEM:三权分立账户, 系统管理员 REMOTE:远程认证账户]
data.authentication	object	认证策略
data.authentication.usbKeyEnable	bool	usb-key认证是否启用
data.authentication.fingerprntEnable	bool	指纹校验是否启用
data.authentication.createUsbKeyStatus	bool	是否创建了usbkey证书
data.authentication.dKey	string	采集的指纹 最大长度:24

data.authentication.manageMode	string[]	管理方式
data.modulePermissions	object[]	模块权限
data.modulePermissions.name	string	模块名 最大长度:60
data.modulePermissions.leaf	bool	模块是否为子项
data.modulePermissions.moduleClass	string	模块所属大类名 最大长度:60
data.modulePermissions.editPermission	object	模块的编辑权限
data.modulePermissions.editPermission.authorizeStatus	bool	是否有权限
data.modulePermissions.editPermission.modifyEnable	bool	能否修改权限
data.modulePermissions.viewPermission	object	模块的查看权限
data.modulePermissions.viewPermission.authorizeStatus	bool	是否有权限
data.modulePermissions.viewPermission.modifyEnable	bool	能否修改权限

请求成功样例:

```
{
  "data": {
    "enable": true,
    "authentication": {
      "usbKeyEnable": false,
      "fingerprintEnable": false,
      "manageMode": [
        "WEBCONSOLE"
      ]
    }
  },
  "modulePermissions": [
    {
      "name": "首页",
      "leaf": false
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "首页",
      "moduleClass": "首页"
    }
  ],
  "name": "安全运营",
  "leaf": false
}
```

```
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "安全运营中心",
      "moduleClass": "安全运营"
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "业务安全",
      "moduleClass": "安全运营"
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "用户安全",
      "moduleClass": "安全运营"
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "勒索专项防护",
      "moduleClass": "安全运营"
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "业务资产管理",
      "moduleClass": "安全运营"
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "黑白名单",
```

```
    "moduleClass": "安全运营"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "热点事件预警",
    "moduleClass": "安全运营"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "下一代安全体系",
    "moduleClass": "安全运营"
  },
  {
    "name": "监控",
    "leaf": false
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "日志",
    "moduleClass": "监控"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "会话",
    "moduleClass": "监控"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "统计",
    "moduleClass": "监控"
  },
  {
    "editPermission": {
      "modifyEnable": false,
      "authorizeStatus": false
    },
    "viewPermission": {
```

```
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "leaf": true,
    "name": "报表",
    "moduleClass": "监控"
},
{
    "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "leaf": true,
    "name": "设置",
    "moduleClass": "监控"
},
{
    "name": "策略",
    "leaf": false
},
{
    "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "leaf": true,
    "name": "地址转换",
    "moduleClass": "策略"
},
{
    "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "leaf": true,
    "name": "访问控制",
    "moduleClass": "策略"
},
{
    "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "leaf": true,
    "name": "安全策略",
    "moduleClass": "策略"
},
{
    "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "leaf": true,
    "name": "流控",
    "moduleClass": "策略"
},
{
```

```
"editPermission": {
  "modifyEnable": true,
  "authorizeStatus": true
},
"viewPermission": {
  "modifyEnable": true,
  "authorizeStatus": true
},
"leaf": true,
"name": "认证",
"moduleClass": "策略"
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "解密",
  "moduleClass": "策略"
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "页面定制",
  "moduleClass": "策略"
},
{
  "name": "对象",
  "leaf": false
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "网络对象",
  "moduleClass": "对象"
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "服务",
  "moduleClass": "对象"
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "对象",
  "moduleClass": "对象"
}
```

```
    "leaf": true,
    "name": "安全策略模板",
    "moduleClass": "对象"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "安全防护规则库",
    "moduleClass": "对象"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "内容识别库",
    "moduleClass": "对象"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "IP地址库",
    "moduleClass": "对象"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "时间计划",
    "moduleClass": "对象"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "信任的证书颁发机构",
    "moduleClass": "对象"
  },
  {
    "name": "网络",
    "leaf": false
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
```



```
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "网络",
    "moduleClass": "网络"
  },
  {
    "name": "系统",
    "leaf": false
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "系统",
    "moduleClass": "系统"
  }
],
"description": "remote",
"name": "test",
"roleName": "REMOTE"
},
"code": 0,
"message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

5.1.7 批量删除管理员账户

功能说明: 通过本接口, 可以批量删除管理员账户

1) API请求数据

POST

https://{host}/api/batch/v1/namespaces/@namespace/account?_method=delete

2) 输入参数

- Body参数

参数	类型	是否必选	描述
obj	object[]	是	管理员账户信息列表
obj.name	string	是	用户名 最小长度:1 最大长度:60

obj.roleName	string	是	账户角色 枚举值: ADMINISTRATOR:超级管理员 COMMON:普通管理员 SAFE:三权分立账户, 安全管理员 LOG:三权分立账户, 审计员 SYSTEM:三权分立账户, 系统管理员 REMOTE:远程认证账户]
--------------	--------	---	--

请求样例:

批量删除管理员账户

POST https://192.168.1.1/api/batch/v1/namespaces/public/account?_method=DELETE

```
[
  {
    "name": "test",
    "roleName": "REMOTE"
  },
  {
    "name": "test1",
    "roleName": "COMMON"
  }
]
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	管理员账户信息列表
data.name	string	用户名 最小长度:1 最大长度:60
data.enable	bool	账户状态, 启用/禁用
data.description	string	描述 最小长度:0 最大长度:95
data.roleName	string	账户角色 枚举值: ADMINISTRATOR:超级管理员 COMMON:普通管理员 SAFE:三权分立账户, 安全管理员 LOG:三权分立账户, 审计员 SYSTEM:三权分立账户, 系统管理员 REMOTE:远程认证账户]
data.authentication	object	认证策略
data.authentication.usbKeyEnable	bool	usb-key认证是否启用
data.authentication.fingerPrintEnable	bool	指纹校验是否启用

data.authentication.createUsbKeyStatus	bool	是否创建了usbkey证书
data.authentication.dKey	string	采集的指纹 最大长度:24
data.authentication.manageMode	string[]	管理方式
data.modulePermissions	object[]	模块权限
data.modulePermissions.name	string	模块名 最大长度:60
data.modulePermissions.leaf	bool	模块是否为子项
data.modulePermissions.moduleClass	string	模块所属大类名 最大长度:60
data.modulePermissions.editPermission	object	模块的编辑权限
data.modulePermissions.editPermission.authorizeStatus	bool	是否有权限
data.modulePermissions.editPermission.modifyEnable	bool	能否修改权限
data.modulePermissions.viewPermission	object	模块的查看权限
data.modulePermissions.viewPermission.authorizeStatus	bool	是否有权限
data.modulePermissions.viewPermission.modifyEnable	bool	能否修改权限

请求成功样例：

```
{
  "data": [
    {
      "enable": true,
      "authentication": {
        "usbKeyEnable": false,
        "fingerprintEnable": false,
        "manageMode": [
          "WEBCONSOLE"
        ]
      },
      "modulePermissions": [
        {
          "name": "首页",
          "leaf": false
        },
        {
          "editPermission": {
            "modifyEnable": true,
            "authorizeStatus": true
          },
          "viewPermission": {
            "modifyEnable": true,
            "authorizeStatus": true
          }
        }
      ]
    }
  ]
}
```

```
    "leaf": true,
    "name": "首页",
    "moduleClass": "首页"
  },
  {
    "name": "安全运营",
    "leaf": false
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "安全运营中心",
    "moduleClass": "安全运营"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "业务安全",
    "moduleClass": "安全运营"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "用户安全",
    "moduleClass": "安全运营"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "勒索专项防护",
    "moduleClass": "安全运营"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "业务资产管理",
    "moduleClass": "安全运营"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    }
  }
```

```
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "黑白名单",
    "moduleClass": "安全运营"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "热点事件预警",
    "moduleClass": "安全运营"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "下一代安全体系",
    "moduleClass": "安全运营"
  },
  {
    "name": "监控",
    "leaf": false
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "日志",
    "moduleClass": "监控"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "会话",
    "moduleClass": "监控"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "统计",
    "moduleClass": "监控"
  }
```

```
    },
    {
      "editPermission": {
        "modifyEnable": false,
        "authorizeStatus": false
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "报表",
      "moduleClass": "监控"
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "设置",
      "moduleClass": "监控"
    },
    {
      "name": "策略",
      "leaf": false
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "地址转换",
      "moduleClass": "策略"
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "访问控制",
      "moduleClass": "策略"
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "安全策略",
      "moduleClass": "策略"
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
```

```
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "流控",
      "moduleClass": "策略"
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "认证",
      "moduleClass": "策略"
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "解密",
      "moduleClass": "策略"
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "页面定制",
      "moduleClass": "策略"
    },
    {
      "name": "对象",
      "leaf": false
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "网络对象",
      "moduleClass": "对象"
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "服务",
      "moduleClass": "对象"
    },
    {
      "editPermission": {
```

```
{
    "modifyEnable": true,
    "authorizeStatus": true
},
{
    "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "leaf": true,
    "name": "安全策略模板",
    "moduleClass": "对象"
},
{
    "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "leaf": true,
    "name": "安全防护规则库",
    "moduleClass": "对象"
},
{
    "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "leaf": true,
    "name": "内容识别库",
    "moduleClass": "对象"
},
{
    "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "leaf": true,
    "name": "IP地址库",
    "moduleClass": "对象"
},
{
    "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "leaf": true,
    "name": "时间计划",
    "moduleClass": "对象"
},
{
    "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "leaf": true,
    "name": "信任的证书颁发机构",
    "moduleClass": "对象"
}
```



```
        "name": "网络",
        "leaf": false
    },
    {
        "editPermission": {
            "modifyEnable": true,
            "authorizeStatus": true
        },
        "viewPermission": {
            "modifyEnable": true,
            "authorizeStatus": true
        },
        "leaf": true,
        "name": "网络",
        "moduleClass": "网络"
    },
    {
        "name": "系统",
        "leaf": false
    },
    {
        "editPermission": {
            "modifyEnable": true,
            "authorizeStatus": true
        },
        "viewPermission": {
            "modifyEnable": true,
            "authorizeStatus": true
        },
        "leaf": true,
        "name": "系统",
        "moduleClass": "系统"
    }
],
"description": "remote",
"name": "test",
"roleName": "REMOTE"
},
{
    "enable": true,
    "authentication": {
        "usbKeyEnable": false,
        "fingerprintEnable": false,
        "manageMode": [
            "APIINTERFACE"
        ]
    },
    "modulePermissions": [
        {
            "name": "首页",
            "leaf": false
        },
        {
            "editPermission": {
                "modifyEnable": true,
                "authorizeStatus": true
            },
            "viewPermission": {
                "modifyEnable": true,
                "authorizeStatus": true
            },
            "leaf": true,
            "name": "首页",
            "moduleClass": "首页"
        },
        {
            "name": "安全运营",
            "leaf": false
        },
        {
            "editPermission": {
                "modifyEnable": true,
                "authorizeStatus": true
            },
            "viewPermission": {
                "modifyEnable": true,
                "authorizeStatus": true
            }
        }
    ]
}
```

```
    },
    "leaf": true,
    "name": "安全运营中心",
    "moduleClass": "安全运营"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "业务安全",
    "moduleClass": "安全运营"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "用户安全",
    "moduleClass": "安全运营"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "勒索专项防护",
    "moduleClass": "安全运营"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "业务资产管理",
    "moduleClass": "安全运营"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "黑白名单",
    "moduleClass": "安全运营"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
```

```
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "热点事件预警",
      "moduleClass": "安全运营"
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "下一代安全体系",
      "moduleClass": "安全运营"
    },
    {
      "name": "监控",
      "leaf": false
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "日志",
      "moduleClass": "监控"
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "会话",
      "moduleClass": "监控"
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "统计",
      "moduleClass": "监控"
    },
    {
      "editPermission": {
        "modifyEnable": false,
        "authorizeStatus": false
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "报表",
      "moduleClass": "监控"
    },
    {
      "editPermission": {
```

```
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "leaf": true,
    "name": "设置",
    "moduleClass": "监控"
},
{
    "name": "策略",
    "leaf": false
},
{
    "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "leaf": true,
    "name": "地址转换",
    "moduleClass": "策略"
},
{
    "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "leaf": true,
    "name": "访问控制",
    "moduleClass": "策略"
},
{
    "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "leaf": true,
    "name": "安全策略",
    "moduleClass": "策略"
},
{
    "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "leaf": true,
    "name": "流控",
    "moduleClass": "策略"
},
{
    "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "leaf": true,
```

```
    "name": "认证",
    "moduleClass": "策略"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "解密",
    "moduleClass": "策略"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "页面定制",
    "moduleClass": "策略"
  },
  {
    "name": "对象",
    "leaf": false
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "网络对象",
    "moduleClass": "对象"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "服务",
    "moduleClass": "对象"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "安全策略模板",
    "moduleClass": "对象"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
```

```
"viewPermission": {
  "modifyEnable": true,
  "authorizeStatus": true
},
"leaf": true,
"name": "安全防护规则库",
"moduleClass": "对象"
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "内容识别库",
  "moduleClass": "对象"
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "IP地址库",
  "moduleClass": "对象"
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "时间计划",
  "moduleClass": "对象"
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "信任的证书颁发机构",
  "moduleClass": "对象"
},
{
  "name": "网络",
  "leaf": false
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "网络",
  "moduleClass": "网络"
},
}
```

```
{
  "name": "系统",
  "leaf": false
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "系统",
  "moduleClass": "系统"
}
],
"description": "1",
"name": "test1",
"roleName": "COMMON"
}
],
"code": 0,
"message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

5.1.8 批量修改管理员账户启用、禁用状态

功能说明: 通过本接口, 可以批量启用、禁用管理员账户

1) API请求数据

PATCH

https://{host}/api/batch/v1/namespaces/@namespace/account

2) 输入参数

• Body参数

参数	类型	是否必选	描述
obj	object[]	是	管理员账户信息列表
obj.name	string	是	用户名 最小长度:1 最大长度:60
obj.enable	bool	是	账户状态, 启用/禁用 默认值:true
obj.roleName	string	是	账户角色 枚举值:[ADMINISTRATOR:超级管理员 COMMON:普通管理员 SAFE:三权分立账户, 安全管理员 LOG:三权分立账户, 审计员 SYSTEM:三权分立账户, 系统管理员 REMOTE:远程认证账户]

请求样例:

```
批量修改管理员账户启用禁用状态为禁用
PATCH https://192.168.1.1/api/batch/v1/namespaces/public/logout
[
  {
    "enable": false,
    "name": "test",
    "roleName": "REMOTE"
  },
  {
    "enable": false,
    "name": "test1",
    "roleName": "COMMON"
  }
]
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	管理员账户信息列表
data.name	string	用户名 最小长度:1 最大长度:60
data.enable	bool	账户状态，启用/禁用
data.description	string	描述 最小长度:0 最大长度:95
data.roleName	string	账户角色 枚举值:[ADMINISTRATOR:超级管理员 COMMON:普通管理员 SAFE:三权分立账户，安全管理员 LOG:三权分立账户，审计员 SYSTEM:三权分立账户，系统管理员 REMOTE:远程认证账户]
data.authentication	object	认证策略
data.authentication.usbKeyEnable	bool	usb-key认证是否启用
data.authentication.fingerprntEnable	bool	指纹校验是否启用
data.authentication.createUsbKeyStatus	bool	是否创建了usbkey证书
data.authentication.dKey	string	采集的指纹 最大长度:24
data.authentication.manageMode	string[]	管理方式
data.modulePermissions	object[]	模块权限

data.modulePermissions.name	string	模块名 最大长度:60
data.modulePermissions.leaf	bool	模块是否为子项
data.modulePermissions.moduleClass	string	模块所属大类名 最大长度:60
data.modulePermissions.editPermission	object	模块的编辑权限
data.modulePermissions.editPermission.authorizeStatus	bool	是否有权限
data.modulePermissions.editPermission.modifyEnable	bool	能否修改权限
data.modulePermissions.viewPermission	object	模块的查看权限
data.modulePermissions.viewPermission.authorizeStatus	bool	是否有权限
data.modulePermissions.viewPermission.modifyEnable	bool	能否修改权限

请求成功样例：

```
{
  "data": [
    {
      "enable": false,
      "authentication": {
        "usbKeyEnable": false,
        "fingerprintEnable": false,
        "manageMode": [
          "WEBCONSOLE"
        ]
      },
      "modulePermissions": [
        {
          "name": "首页",
          "leaf": false
        },
        {
          "editPermission": {
            "modifyEnable": true,
            "authorizeStatus": true
          },
          "viewPermission": {
            "modifyEnable": true,
            "authorizeStatus": true
          },
          "leaf": true,
          "name": "首页",
          "moduleClass": "首页"
        },
        {
          "name": "安全运营",
          "leaf": false
        },
        {
          "editPermission": {
            "modifyEnable": true,
            "authorizeStatus": true
          }
        }
      ]
    }
  ]
}
```

```
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "安全运营中心",
    "moduleClass": "安全运营"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "业务安全",
    "moduleClass": "安全运营"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "用户安全",
    "moduleClass": "安全运营"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "勒索专项防护",
    "moduleClass": "安全运营"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "业务资产管理",
    "moduleClass": "安全运营"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "黑白名单",
    "moduleClass": "安全运营"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
```

```
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "热点事件预警",
    "moduleClass": "安全运营"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "下一代安全体系",
    "moduleClass": "安全运营"
  },
  {
    "name": "监控",
    "leaf": false
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "日志",
    "moduleClass": "监控"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "会话",
    "moduleClass": "监控"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "统计",
    "moduleClass": "监控"
  },
  {
    "editPermission": {
      "modifyEnable": false,
      "authorizeStatus": false
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "报表",
    "moduleClass": "监控"
  }
```

```
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "设置",
      "moduleClass": "监控"
    },
    {
      "name": "策略",
      "leaf": false
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "地址转换",
      "moduleClass": "策略"
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "访问控制",
      "moduleClass": "策略"
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "安全策略",
      "moduleClass": "策略"
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "流控",
      "moduleClass": "策略"
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
```

```
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "认证",
      "moduleClass": "策略"
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "解密",
      "moduleClass": "策略"
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "页面定制",
      "moduleClass": "策略"
    },
    {
      "name": "对象",
      "leaf": false
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "网络对象",
      "moduleClass": "对象"
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "服务",
      "moduleClass": "对象"
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "安全策略模板",
      "moduleClass": "对象"
    },
    {
      "editPermission": {
```

```
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "leaf": true,
    "name": "安全防护规则库",
    "moduleClass": "对象"
},
{
    "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "leaf": true,
    "name": "内容识别库",
    "moduleClass": "对象"
},
{
    "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "leaf": true,
    "name": "IP地址库",
    "moduleClass": "对象"
},
{
    "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "leaf": true,
    "name": "时间计划",
    "moduleClass": "对象"
},
{
    "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "leaf": true,
    "name": "信任的证书颁发机构",
    "moduleClass": "对象"
},
{
    "name": "网络",
    "leaf": false
},
{
    "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
    },
    "leaf": true,
```

```
        "name": "网络",
        "moduleClass": "网络"
    },
    {
        "name": "系统",
        "leaf": false
    },
    {
        "editPermission": {
            "modifyEnable": true,
            "authorizeStatus": true
        },
        "viewPermission": {
            "modifyEnable": true,
            "authorizeStatus": true
        },
        "leaf": true,
        "name": "系统",
        "moduleClass": "系统"
    }
],
"description": "remote",
"name": "test",
"roleName": "REMOTE"
},
{
    "enable": false,
    "authentication": {
        "usbKeyEnable": false,
        "fingerprintEnable": false,
        "manageMode": [
            "WEBCONSOLE"
        ]
    },
    "modulePermissions": [
        {
            "name": "首页",
            "leaf": false
        },
        {
            "editPermission": {
                "modifyEnable": true,
                "authorizeStatus": true
            },
            "viewPermission": {
                "modifyEnable": true,
                "authorizeStatus": true
            },
            "leaf": true,
            "name": "首页",
            "moduleClass": "首页"
        },
        {
            "name": "安全运营",
            "leaf": false
        },
        {
            "editPermission": {
                "modifyEnable": true,
                "authorizeStatus": true
            },
            "viewPermission": {
                "modifyEnable": true,
                "authorizeStatus": true
            },
            "leaf": true,
            "name": "安全运营中心",
            "moduleClass": "安全运营"
        },
        {
            "editPermission": {
                "modifyEnable": true,
                "authorizeStatus": true
            },
            "viewPermission": {
                "modifyEnable": true,
                "authorizeStatus": true
            }
        }
    ]
}
```

```
    },
    "leaf": true,
    "name": "业务安全",
    "moduleClass": "安全运营"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "用户安全",
    "moduleClass": "安全运营"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "勒索专项防护",
    "moduleClass": "安全运营"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "业务资产管理",
    "moduleClass": "安全运营"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "黑白名单",
    "moduleClass": "安全运营"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "热点事件预警",
    "moduleClass": "安全运营"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
```



```
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "下一代安全体系",
      "moduleClass": "安全运营"
    },
    {
      "name": "监控",
      "leaf": false
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "日志",
      "moduleClass": "监控"
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "会话",
      "moduleClass": "监控"
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "统计",
      "moduleClass": "监控"
    },
    {
      "editPermission": {
        "modifyEnable": false,
        "authorizeStatus": false
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "报表",
      "moduleClass": "监控"
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "设置",
      "moduleClass": "监控"
    },
    {
      "name": "策略",
```

```
    "leaf": false
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "地址转换",
    "moduleClass": "策略"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "访问控制",
    "moduleClass": "策略"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "安全策略",
    "moduleClass": "策略"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "流控",
    "moduleClass": "策略"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
    "name": "认证",
    "moduleClass": "策略"
  },
  {
    "editPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "viewPermission": {
      "modifyEnable": true,
      "authorizeStatus": true
    },
    "leaf": true,
```

```

    },
    {
      "name": "解密",
      "moduleClass": "策略"
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "页面定制",
      "moduleClass": "策略"
    },
    {
      "name": "对象",
      "leaf": false
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "网络对象",
      "moduleClass": "对象"
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "服务",
      "moduleClass": "对象"
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "安全策略模板",
      "moduleClass": "对象"
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "viewPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },
      "leaf": true,
      "name": "安全防护规则库",
      "moduleClass": "对象"
    },
    {
      "editPermission": {
        "modifyEnable": true,
        "authorizeStatus": true
      },

```

```
"viewPermission": {
  "modifyEnable": true,
  "authorizeStatus": true
},
"leaf": true,
"name": "内容识别库",
"moduleClass": "对象"
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "IP地址库",
  "moduleClass": "对象"
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "时间计划",
  "moduleClass": "对象"
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "信任的证书颁发机构",
  "moduleClass": "对象"
},
{
  "name": "网络",
  "leaf": false
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "网络",
  "moduleClass": "网络"
},
{
  "name": "系统",
  "leaf": false
},
{
  "editPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "viewPermission": {
    "modifyEnable": true,
    "authorizeStatus": true
  },
  "leaf": true,
  "name": "系统",
  "moduleClass": "系统"
}
```

```
        "leaf": true,
        "name": "系统",
        "moduleClass": "系统"
      },
      "description": "",
      "name": "test1",
      "roleName": "COMMON"
    }
  ],
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

5.1.9 获取账户密码安全策略

功能说明：通过本接口，可以获取账户密码安全策略

1) API请求数据

GET

https://{host}/api/v1/namespaces/@namespace/accountpasswdpolicy

2) 输入参数

请求样例：

获取密码安全策略

GET https://192.168.1.1/api/v1/namespaces/public/accountpasswdpolicy

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	密码安全策略
data.changeInitialPasswd	bool	下次登录是否需要修改密码，true:是，false:否
data.passwdExpiredChange	bool	密码存在有效期，过期是否需要修改密码，true:是，false:否
data.passwdValidity	int32	密码的有效期

请求成功样例：

```
{
  "data": {
    "passwdValidity": 30,
    "changeInitialPasswd": false,
    "passwdExpiredChange": false
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

5.1.10 修改账户密码安全策略

功能说明：通过本接口，可以修改账户密码安全策略

1) API请求数据

PATCH https://{host}/api/v1/namespaces/@namespace/accountpasswdpolicy

2) 输入参数

- Body参数

参数	类型	是否必选	描述
obj	object	是	密码安全策略
obj.changeInitialPasswd	bool	否	下次登录是否需要修改密码，true:是，false:否
obj.passwdExpiredChange	bool	否	密码存在有效期，过期是否需要修改密码，true:是，false:否
obj.passwdValidity	int32	是	密码的有效期

请求样例：

修改密码安全策略，启用密码有效期并设置为15天

```
PATCH https://192.168.1.1/api/v1/namespaces/public/accountpasswdpolicy
{
  "passwdValidity": 15,
  "changeInitialPasswd": false,
  "passwdExpiredChange": true
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	密码安全策略
data.changeInitialPasswd	bool	下次登录是否需要修改密码，true:是，false:否
data.passwdExpiredChange	bool	密码存在有效期，过期是否需要修改密码，true:是，false:否
data.passwdValidity	int32	密码的有效期

请求成功样例：

```
{
  "data": {
    "passwdValidity": 15,
    "changeInitialPasswd": false,
    "passwdExpiredChange": true
  },
  "code": 0,
}
```

```
    "message": "成功"
  }
  请求失败样例:
  {
    "data": "",
    "code": 22,
    "message": "无效参数"
  }
```

5.2 排障

5.2.1 获取系统模块列表

功能说明：获取系统模块列表

1) API请求数据

GET

https://{host}/api/v1/namespaces/@namespace/sysmodule

2) 输入参数

- URL参数

参数	类型	是否必选	描述
_start	int32	否	起始位置, 从 0 开始 最小值:0 最大值:2000000
_length	int32	否	获取最大长度, 限制最大值, 如果数据不够可能返回小于 length 最小值:1 最大值:200 默认值:true

请求样例:

获取系统功能模块列表,从第1条开始, 展示1条
GET https://192.168.1.1/api/v1/namespaces/public/sysmodule?_start=1&_length=1

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	分页数据
data.totalItems	int32	总共多少项目
data.totalPages	int32	总共多少页
data.pageNumber	int32	当前页码, 从 1 开始
data.pageSize	int32	每页多大
data.itemsOffset	int32	当前条目偏移, 从 0 开始
data.itemLength	int32	数据列表长度

data.items	object[]	有效数据列表
data.items.name	string	模块名称 最小长度:1 最大长度:95
data.items.description	string	模块名称描述 最小长度:0 最大长度:95

请求成功样例:

```
{
  "data": {
    "itemsOffset": 0,
    "items": [
      {
        "name": "VPNServ",
        "description": "DLAN总部"
      }
    ],
    "totalItems": 71,
    "pageNumber": 1,
    "itemLength": 1,
    "totalPages": 72,
    "pageSize": 1
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

5.2.2 获取系统故障日志

功能说明: 获取系统故障日志

1) API请求数据

POST

https://{host}/api/v1/namespaces/@namespace/syslog

2) 输入参数

- URL参数

参数	类型	是否必选	描述
_start	int32	否	起始位置, 从 0 开始 最小值:0 最大值:2000000
_length	int32	否	获取最大长度, 限制最大值, 如果数据不够可能返回小于 length 最小值:1 最大值:200 默认值:true

- Body参数

参数	类型	是否必选	描述
obj	object	是	返回系统模块名称

obj.filter	object	是	命令任务执行需要的body参数
obj.filter.levels	string[]	否	日志等级:info warn error debug
obj.filter.date	string	否	日期,例:20191115 最小长度:8 最大长度:8
obj.filter.name	string[]	否	模块名,对应/v1/namespaces/@namespace/sysmodule中返回的name字段

请求样例:

获取指定模块的系统故障日志 (name:[SSLVPN]),从第1条开始,展示100条

POST https://192.168.1.1/api/v1/namespaces/public/syslog?_start=1&_length=100

```
{
  "filter": {
    "name": [
      "SSLVPN"
    ],
    "date": "20191225"
  }
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	分页数据
data.totalItems	int32	总共多少项目
data.totalPages	int32	总共多少页
data.pageNumber	int32	当前页码, 从 1 开始
data.pageSize	int32	每页多大
data.itemsOffset	int32	当前条目偏移, 从 0 开始
data.itemLength	int32	数据列表长度
data.items	object[]	有效数据列表
data.items.moduleName	string	模块名称 最小长度:1 最大长度:95
data.items.logLevel	string	日志类型 枚举值: info:信息 warn:告警 error:错误 debug:调试]
data.items.time	string	时间
data.items.logInfo	string	详细信息 最大长度:1024

请求成功样例:

```
{
  "data": {
    "itemsOffset": 0,
    "items": [],
    "totalItems": 0,
    "pageNumber": 1,
    "itemLength": 0,
    "totalPages": 1,
    "pageSize": 100
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

5.3 高可用

5.3.1 获取所有组双机的状态

功能说明：获取所有组双机的状态

1) API请求数据

GET

https://{host}/api/v1/namespaces/@namespace/ha/groupstate

2) 输入参数

请求样例：

获取组双机热备列表信息

GET https://192.168.1.1/api/v1/namespaces/public/groupstate

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	分页数据
data.totalItems	int32	总共多少项目
data.totalPages	int32	总共多少页
data.pageNumber	int32	当前页码，从 1 开始
data.pageSize	int32	每页多大
data.itemsOffset	int32	当前条目偏移，从 0 开始
data.itemLength	int32	数据列表长度
data.items	object[]	有效数据列表
data.items.groupId	uint16	双机虚拟组id 最小值:0 最大值:255

data.items.local	object	本地状态信息
data.items.local.priority	uint16	优先级 最小值:1 最大值:255
data.items.local.preempt	string	抢占模式 枚举值:[enable:抢占开关:开 disable:抢占开关:关]
data.items.local.state	string	状态 枚举值:[MASTER:主 SLAVE:备 INIT:未知]

请求成功样例：

```
{
  "data": {
    "itemsOffset": 0,
    "items": [
      {
        "local": {
          "state": "INIT",
          "priority": 2,
          "preempt": "disable"
        },
        "groupId": 100
      },
      {
        "local": {
          "state": "INIT",
          "priority": 100,
          "preempt": "enable"
        },
        "groupId": 101
      }
    ],
    "totalItems": 2,
    "pageNumber": 1,
    "itemLength": 2,
    "totalPages": 1,
    "pageSize": 100
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

5.3.2 获取指定组双机的状态

功能说明：获取指定组双机的状态

1) API请求数据

GET

https://{host}/api/v1/namespaces/@namespace/ha/groupstate/@name

2) 输入参数

请求样例:

```
获取指定组双机test热备信息

GET https://192.168.1.1/api/v1/namespaces/public/groupstate/test
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	双机状态信息
data.groupId	uint16	双机虚拟组id 最小值:0 最大值:255
data.local	object	本地状态信息
data.local.priority	uint16	优先级 最小值:1 最大值:255
data.local.preempt	string	抢占模式 枚举值:[enable:抢占开关:开 disable:抢占开关:关]
data.local.state	string	状态 枚举值:[MASTER:主 SLAVE:备 INIT:未知]

请求成功样例:

```
{
  "data": {
    "local": {
      "state": "INIT",
      "priority": 2,
      "preempt": "disable"
    },
    "groupId": 100
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

5.4 安全能力更新

5.4.1 获取安全能力更新库列表

功能说明：通过此接口，查询规则库具体信息，包括版本号

1) API请求数据

GET

https://{host}/api/v1/namespaces/@namespace/updatelib

2) 输入参数

- URL参数

参数	类型	是否必选	描述
_start	int32	否	起始位置, 从 0 开始 最小值:0 最大值:2000000
_length	int32	否	获取最大长度，限制最大值，如果数据不够可能返回小于 length 最小值:1 最大值:200 默认值:true

请求样例：

获取安全能力更新库列表

GET https://192.168.1.1/api/v1/namespaces/public/updatelib

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	分页数据
data.totalItems	int32	总共多少项目
data.totalPages	int32	总共多少页
data.pageNumber	int32	当前页码，从 1 开始
data.pageSize	int32	每页多大
data.itemsOffset	int32	当前条目偏移, 从 0 开始
data.itemLength	int32	数据列表长度
data.items	object[]	有效数据列表
data.items.name	string	库名称 最大长度:64
data.items.groupName	string	所属组名称 最大长度:64
data.items.curVersion	string	当前版本 最大长度:32

data.items.newVersion	string	最新版本 最大长度:64
data.items.updateInterval	string	更新间隔 最大长度:32
data.items.expire	string	有效期 最大长度:32
data.items.autoUpdate	bool	是否自动更新
data.items.allowRollback	bool	是否允许回滚

请求成功样例:

```
{
  "data": {
    "itemsOffset": 0,
    "items": [
      {
        "curVersion": "2020-03-02 11:00:00",
        "autoUpdate": true,
        "newVersion": "2020-03-02 11:00:00",
        "allowRollback": true,
        "updateInterval": "1个月",
        "expire": "2020-08-04",
        "name": "SAVE安全智能文件检测模型库",
        "groupName": "防病毒模型库"
      },
      {
        "curVersion": "2020-05-07 14:45:32",
        "autoUpdate": true,
        "newVersion": "2020-05-07 14:45:32",
        "allowRollback": true,
        "updateInterval": "14天",
        "expire": "2020-08-04",
        "name": "URL库",
        "groupName": "云脑-云智最新威胁防护库"
      },
      {
        "curVersion": "2020-06-24 15:00:00",
        "autoUpdate": true,
        "newVersion": "2020-06-24 15:00:00",
        "allowRollback": true,
        "updateInterval": "14天",
        "expire": "2020-08-04",
        "name": "漏洞攻击特征识别库",
        "groupName": "云脑-云智最新威胁防护库"
      },
      {
        "curVersion": "--",
        "autoUpdate": true,
        "newVersion": "2020-06-29 00:00:00",
        "allowRollback": true,
        "updateInterval": "",
        "expire": "永不过期",
        "name": "软件优化",
        "groupName": "基础更新库"
      },
      {
        "curVersion": "2019-09-02 12:34:56",
        "autoUpdate": true,
        "newVersion": "2020-05-09 11:55:55",
        "allowRollback": true,
        "updateInterval": "14天",
        "expire": "2020-08-04",
        "name": "应用识别库",
        "groupName": "云脑-云智最新威胁防护库"
      },
      {
        "curVersion": "2016-10-19 17:00:00",
        "autoUpdate": false,
        "newVersion": "2016-10-19 17:00:00",
        "allowRollback": true,
      }
    ]
  }
}
```

```
"updateInterval": "14天",
"expire": "2020-08-04",
"name": "WEB应用防护库",
"groupName": "云脑-云智最新威胁防护库"
},
{
  "curVersion": "2016-10-16 16:03:19",
  "autoUpdate": false,
  "newVersion": "2016-10-16 16:03:19",
  "allowRollback": true,
  "updateInterval": "14天",
  "expire": "2020-08-04",
  "name": "数据泄密防护库",
  "groupName": "云脑-云智最新威胁防护库"
},
{
  "curVersion": "2016-07-06 17:00:00",
  "autoUpdate": false,
  "newVersion": "2016-09-27 17:00:00",
  "allowRollback": true,
  "updateInterval": "14天",
  "expire": "2020-08-04",
  "name": "实时漏洞分析识别库",
  "groupName": "云脑-云智最新威胁防护库"
},
{
  "curVersion": "2020-05-19 10:00:00",
  "autoUpdate": true,
  "newVersion": "2020-05-19 10:00:00",
  "allowRollback": true,
  "updateInterval": "",
  "expire": "永不过期",
  "name": "IP地址库",
  "groupName": "基础更新库"
},
{
  "curVersion": "2016-09-23 16:30:01",
  "autoUpdate": true,
  "newVersion": "2016-09-23 16:30:01",
  "allowRollback": false,
  "updateInterval": "",
  "expire": "永不过期",
  "name": "热点事件预警与处置库",
  "groupName": "基础更新库"
},
{
  "curVersion": "2019-01-14 00:00:00",
  "autoUpdate": true,
  "newVersion": "2019-01-14 00:00:00",
  "allowRollback": true,
  "updateInterval": "14天",
  "expire": "2020-08-04",
  "name": "僵尸网络与病毒防护库",
  "groupName": "云脑-云智最新威胁防护库"
},
{
  "curVersion": "2018-07-31 16:17:29",
  "autoUpdate": true,
  "newVersion": "2018-07-31 16:17:29",
  "allowRollback": true,
  "updateInterval": "14天",
  "expire": "2020-08-04",
  "name": "热点事件库",
  "groupName": "云脑-云智最新威胁防护库"
},
{
  "curVersion": "2020-03-02 11:00:00",
  "autoUpdate": true,
  "newVersion": "2020-03-02 11:00:00",
  "allowRollback": true,
  "updateInterval": "1个月",
  "expire": "2020-08-04",
  "name": "SAVE安全智能文件检测模型库",
  "groupName": "防病毒模型库"
},
{
  "curVersion": "2020-05-07 14:45:32",
```

```
"autoUpdate": true,
"newVersion": "2020-05-07 14:45:32",
"allowRollback": true,
"updateInterval": "14天",
"expire": "2020-08-04",
"name": "URL库",
"groupName": "云脑-云智最新威胁防护库"
},
{
  "curVersion": "2020-06-24 15:00:00",
  "autoUpdate": true,
  "newVersion": "2020-06-24 15:00:00",
  "allowRollback": true,
  "updateInterval": "14天",
  "expire": "2020-08-04",
  "name": "IPS漏洞特征识别库",
  "groupName": "云脑-云智最新威胁防护库"
},
{
  "curVersion": "--",
  "autoUpdate": true,
  "newVersion": "2020-06-29 00:00:00",
  "allowRollback": true,
  "updateInterval": "",
  "expire": "永不过期",
  "name": "软件优化",
  "groupName": "基础更新库"
},
{
  "curVersion": "2019-09-02 12:34:56",
  "autoUpdate": true,
  "newVersion": "2020-05-09 11:55:55",
  "allowRollback": true,
  "updateInterval": "14天",
  "expire": "2020-08-04",
  "name": "应用识别库",
  "groupName": "云脑-云智最新威胁防护库"
},
{
  "curVersion": "2016-10-19 17:00:00",
  "autoUpdate": true,
  "newVersion": "2016-10-19 17:00:00",
  "allowRollback": true,
  "updateInterval": "14天",
  "expire": "2020-08-04",
  "name": "WEB应用防护库",
  "groupName": "云脑-云智最新威胁防护库"
},
{
  "curVersion": "2016-10-16 16:03:19",
  "autoUpdate": true,
  "newVersion": "2016-10-16 16:03:19",
  "allowRollback": true,
  "updateInterval": "14天",
  "expire": "2020-08-04",
  "name": "数据泄密防护库",
  "groupName": "云脑-云智最新威胁防护库"
},
{
  "curVersion": "2016-07-06 17:00:00",
  "autoUpdate": true,
  "newVersion": "2016-09-27 17:00:00",
  "allowRollback": true,
  "updateInterval": "14天",
  "expire": "2020-08-04",
  "name": "实时漏洞分析识别库",
  "groupName": "云脑-云智最新威胁防护库"
},
{
  "curVersion": "2020-05-19 10:00:00",
  "autoUpdate": true,
  "newVersion": "2020-05-19 10:00:00",
  "allowRollback": true,
  "updateInterval": "",
  "expire": "永不过期",
  "name": "IP地址库",
  "groupName": "基础更新库"
```



```

    },
    {
      "curVersion": "2016-09-23 16:30:01",
      "autoUpdate": true,
      "newVersion": "2016-09-23 16:30:01",
      "allowRollback": false,
      "updateInterval": "",
      "expire": "永不过期",
      "name": "热点事件预警与处置库",
      "groupName": "基础更新库"
    },
    {
      "curVersion": "2019-01-14 00:00:00",
      "autoUpdate": true,
      "newVersion": "2019-01-14 00:00:00",
      "allowRollback": true,
      "updateInterval": "14天",
      "expire": "2020-08-04",
      "name": "僵尸网络与病毒防护库",
      "groupName": "云脑-云智最新威胁防护库"
    },
    {
      "curVersion": "2018-07-31 16:17:29",
      "autoUpdate": true,
      "newVersion": "2018-07-31 16:17:29",
      "allowRollback": true,
      "updateInterval": "14天",
      "expire": "2020-08-04",
      "name": "热点事件库",
      "groupName": "云脑-云智最新威胁防护库"
    },
    {
      "curVersion": "2020-06-29 03:20:57",
      "autoUpdate": true,
      "newVersion": "2020-06-29 03:20:57",
      "allowRollback": false,
      "updateInterval": "5分钟",
      "expire": "2020-08-04",
      "name": "云鉴检测威胁情报",
      "groupName": "云脑-云鉴情报库"
    }
  ],
  "totalItems": 25,
  "pageNumber": 1,
  "itemLength": 25,
  "totalPages": 1,
  "pageSize": 100
},
"code": 0,
"message": "成功"
}

```

请求失败样例：

```

{
  "data": "",
  "code": 22,
  "message": "无效参数"
}

```

5.4.2 升级规则库

功能说明：通过此接口，升级指定的规则库

1) API请求数据

POST https://{host}/api/v1/namespaces/@namespace/updatelib

2) 输入参数

- Body参数

参数	类型	是否必选	描述
----	----	------	----

obj	object[]	是	安全能力更新库列表
obj.name	string	是	库名称 最大长度:64
obj.groupName	string	是	所属组名称 最大长度:64
obj.curVersion	string	是	当前版本 最大长度:32
obj.newVersion	string	是	最新版本 最大长度:64
obj.updateInterval	string	是	更新间隔 最大长度:32
obj.expire	string	是	有效期 最大长度:32
obj.autoUpdate	bool	是	是否自动更新
obj.allowRollback	bool	是	是否允许回滚

请求样例:

```
升级规则库
POST https://192.168.1.1/api/v1/namespaces/public/updatelib
[
  {
    "name": "flux_inspect"
  }
]
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	string	

请求成功样例:

```
{
  "data": "当前库已升至最新",
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

5.5 系统维护

5.5.1 系统升级

5.5.1.1 获取系统升级日志

功能说明：获取系统升级日志

1) API请求数据

GET

https://{host}/api/v1/namespaces/@namespace/updatelog

2) 输入参数

• URL参数

参数	类型	是否必选	描述
_start	int32	否	起始位置, 从 0 开始 最小值:0 最大值:2000000
_length	int32	否	获取最大长度, 限制最大值, 如果数据不够可能返回小于 length 最小值:1 最大值:200 默认值:true

请求样例:

获取系统升级日志, 从第1条开始, 展示10条

GET https://192.168.1.1/api/v1/namespaces/public/updatelog?_start=1&_length=10

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	分页数据
data.totalItems	int32	总共多少项目
data.totalPages	int32	总共多少页
data.pageNumber	int32	当前页码, 从 1 开始
data.pageSize	int32	每页多大
data.itemsOffset	int32	当前条目偏移, 从 0 开始
data.itemLength	int32	数据列表长度
data.items	object[]	有效数据列表
data.items.version	string	版本号 最小长度:1 最大长度:32
data.items.packageName	string	升级包 最小长度:1 最大长度:95
data.items.time	string	升级时间

data.items.result	string	升级结果 最大长度:16
data.items.tool	string	升级方式 最小长度:1 最大长度:95
data.items.rollbackSupport	bool	是否支持回滚

请求成功样例:

```
{
  "data": {
    "itemsOffset": 1,
    "items": [],
    "totalItems": 1,
    "pageNumber": 1,
    "itemLength": 0,
    "totalPages": 1,
    "pageSize": 10
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

5.6 导入导出

5.6.1 ftp文件上传下载

功能说明: ftp导入导出

1) API请求数据

POST

https://{host}/api/v1/namespaces/@namespace/ftp

2) 输入参数

• Body参数

参数	类型	是否必选	描述
obj	object	是	ftp请求参数
obj.opr	string	是	文件操作类型 枚举值:[GET:ftp下载文件 PUT:ftp上传文件]
obj.host	string	是	ftp服务器IP地址
obj.userName	string	是	ftp服务器用户名 最小长度:1 最大长度:95

obj.password	string	是	ftp服务器密码 最小长度:1 最大长度:95
obj.localFile	string	否	本地文件名称(ftp上传才起作用) 最小长度:1 最大长度:255
obj.remoteFile	string	是	远程文件名称 最小长度:1 最大长度:255

请求样例:

ftp文件上传下载, 上传条件: 从ftp服务192.168.1.10上传文件test.csv

```
POST https://192.168.1.1/api/v1/namespaces/public/ftp
{
  "host": "192.168.1.10",
  "opr": "GET",
  "remoteFile": "test.csv",
  "password": "test",
  "userName": "test"
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	ftp上传下载文件返回信息
data.fileName	string	操作的文件名字 最小长度:1 最大长度:255
data.fileSize	string	上传或者下载的文件大小 最小长度:1 最大长度:95

请求成功样例:

```
{
  "data": {
    "fileSize": "795.1K",
    "fileName": "1592270888626"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

6 运营中心

6.1 黑白名单

6.1.1 获取符合过滤条件的全部黑白名单

功能说明：通过此接口可以获取全部黑白名单

1) API请求数据

GET

https://{host}/api/v1/namespaces/@namespace/whiteblacklist

2) 输入参数

- URL参数

参数	类型	是否必选	描述
_start	int32	否	起始位置, 从 0 开始 最小值:0 最大值:2000000
_sortby	string	否	指定排序字段 最大长度:100
type	string	否	黑名单 BLACK, 白名单 WHITE 枚举值:[BLACK:黑名单 WHITE:白名单]
_search	string	否	模糊搜索关键字 最大长度:95
_select	string	否	选择字段, 字段用逗号分隔 最大长度:1000
url	string	否	IP,IP网段或者域名,url 最小长度:1 最大长度:4096
_order	string	否	指定排序方式 (正序/倒序) 默认值:true 枚举值:[asc:按升序排列 desc:按降序排列]
_length	int32	否	获取最大长度, 限制最大值, 如果数据不够 可能返回小于 length 最小值:1 最大值:200 默认值:true
description	string	否	描述信息 最小长度:0 最大长度:95

请求样例：

获取所有黑白名单
GET https://192.168.1.1/api/v1/namespaces/public/whiteblacklist?type=WHITE

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	分页数据
data.totalItems	int32	总共多少项目
data.totalPages	int32	总共多少页
data.pageNumber	int32	当前页码, 从 1 开始
data.pageSize	int32	每页多大
data.itemsOffset	int32	当前条目偏移, 从 0 开始
data.itemLength	int32	数据列表长度
data.items	object[]	有效数据列表
data.items.url	string	主要是IP, 域名, URL 最小长度:1 最大长度:1024
data.items.enable	bool	启用 true, 禁用 false
data.items.type	string	黑名单 BLACK, 白名单 WHITE 枚举值: BLACK:黑名单 WHITE:白名单]
data.items.domain	uint32	0 IP, 1 域名, 3 URL
data.items.isDefault	bool	内置 true 自定义 false
data.items.createTime	string	创建时间
data.items.description	string	描述 最小长度:0 最大长度:95

请求成功样例:

```
{
  "data": {
    "itemsOffset": 0,
    "items": [
      {
        "enable": true,
        "type": "WHITE",
        "createTime": "2011-07-01 08:30:00",
        "isDefault": true,
        "url": "www.sangfor.com",
        "domain": true,
        "uuid": "07150a520899461e956bd9f0a69ea82d",
        "description": "www.sangfor.com"
      }
    ],
    "totalItems": 1,
    "pageNumber": 0,
    "totalPages": 1,
  }
}
```

```
        "itemLength": 1,
        "pageSize": 200
    },
    "code": 0,
    "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

6.1.2 获取单个黑白名单详细信息

功能说明：通过此接口可以查询单个黑白名单

1) API请求数据

GET

https://{host}/api/v1/namespaces/@namespace/whiteblacklist/@url

2) 输入参数

• URL参数

参数	类型	是否必选	描述
type	string	否	黑名单BLACK，白名单WHITE 枚举值:[BLACK:黑名单 WHITE:白名单]

请求样例:

获取单个黑白Ip详细信息

GET https://192.168.1.1/api/v1/namespaces/public/whiteblacklist/www.sangfor.com?type=WHITE

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	黑白IP信息
data.url	string	主要是IP，域名，URL 最小长度:1 最大长度:1024
data.enable	bool	启用 true, 禁用 false
data.type	string	黑名单 BLACK, 白名单 WHITE 枚举值:[BLACK:黑名单 WHITE:白名单]
data.domain	uint32	0 IP, 1 域名, 3 URL
data.isDefault	bool	内置 true 自定义 false

data.createTime	string	创建时间
data.description	string	描述 最小长度:0 最大长度:95

请求成功样例:

```
{
  "data": {
    "enable": true,
    "url": "www.sangfor.com",
    "createTime": "2011-07-01 08:30:00",
    "isDefault": true,
    "type": "WHITE",
    "domain": true,
    "uuid": "07150a520899461e956bd9f0a69ea82d",
    "description": "www.sangfor.com"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

6.1.3 添加黑白名单

功能说明: 添加黑白名单

1) API请求数据

POST https://{host}/api/v1/namespaces/@namespace/whiteblacklist

2) 输入参数

• URL参数

参数	类型	是否必选	描述
type	string	否	黑名单BLACK, 白名单WHITE 枚举值: BLACK:黑名单 WHITE:白名单]

• Body参数

参数	类型	是否必选	描述
obj	object	是	黑白IP信息
obj.url	string	是	主要是IP, 域名, URL 最小长度:1 最大长度:1024
obj.enable	bool	否	启用 true, 禁用 false 默认值:true

obj.type	string	是	黑名单 BLACK, 白名单 WHITE 枚举值: BLACK:黑名单 WHITE:白名单]
obj.description	string	否	描述 最小长度:0 最大长度:95

请求样例:

添加黑白名单

```
POST https://192.168.1.1/api/v1/namespaces/public/whiteblacklist
{
  "enable": true,
  "type": "WHITE",
  "url": "www.baidu.com",
  "description": "www.baidu.com"
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	黑白IP信息
data.url	string	主要是IP, 域名, URL 最小长度:1 最大长度:1024
data.enable	bool	启用 true, 禁用 false
data.type	string	黑名单 BLACK, 白名单 WHITE 枚举值: BLACK:黑名单 WHITE:白名单]
data.domain	uint32	0 IP, 1 域名, 3 URL
data.isDefault	bool	内置 true 自定义 false
data.createTime	string	创建时间
data.description	string	描述 最小长度:0 最大长度:95

请求成功样例:

```
{
  "data": {
    "enable": true,
    "url": "www.baidu.com",
    "createTime": "2011-07-01 08:30:00",
    "isDefault": true,
    "type": "WHITE",
    "domain": true,
    "uuid": "07150a520899461e956bd9f0a69ea82d",
    "description": "www.baidu.com"
  },
}
```

```
"code": 0,  
"message": "成功"  
}
```

请求失败样例:

```
{  
  "data": "",  
  "code": 22,  
  "message": "无效参数"  
}
```

6.1.4 批量添加黑白名单信息

功能说明: 批量添加黑白名单

1) API请求数据

POST https://{host}/api/batch/v1/namespaces/@namespace/whiteblacklist

2) 输入参数

• URL参数

参数	类型	是否必选	描述
override	string	否	冲突类型 枚举值:[PROMPT:冲突报错提示, 默认项 REPLACE:冲突替换 SKIP:冲突跳过]

• Body参数

参数	类型	是否必选	描述
obj	object[]	是	黑白名单列表
obj.url	string	是	主要是IP, 域名, URL 最小长度:1 最大长度:1024
obj.enable	bool	否	启用 true, 禁用 false 默认值:true
obj.type	string	是	黑名单 BLACK, 白名单 WHITE 枚举值:[BLACK:黑名单 WHITE:白名单]
obj.description	string	否	描述 最小长度:0 最大长度:95

请求样例:

批量添加黑白名单

```
POST https://192.168.1.1/api/batch/v1/namespaces/public/whiteblacklist  
[  
  {  
    "enable": true,  
    "type": "WHITE",  
    "url": "www.baidu.com",  
    "description": "www.baidu.com"
```

```
}  
]
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	黑白名单列表
data.url	string	主要是IP， 域名, URL 最小长度:1 最大长度:1024
data.enable	bool	启用 true, 禁用 false
data.type	string	黑名单 BLACK, 白名单 WHITE 枚举值: BLACK:黑名单 WHITE:白名单]
data.domain	uint32	0 IP, 1 域名, 3 URL
data.isDefault	bool	内置 true 自定义 false
data.createTime	string	创建时间
data.description	string	描述 最小长度:0 最大长度:95

请求成功样例:

```
{  
  "data": {  
    "itemsOffset": 0,  
    "items": [  
      {  
        "enable": true,  
        "url": "www.baidu.com",  
        "createTime": "2011-07-01 08:30:00",  
        "isDefault": true,  
        "type": "WHITE",  
        "domain": true,  
        "uuid": "07150a520899461e956bd9f0a69ea82d",  
        "description": "www.baidu.com"  
      }  
    ],  
    "totalItems": 1,  
    "pageNumber": 0,  
    "pageSize": 200,  
    "itemLength": 1,  
    "totalPages": 1  
  },  
  "code": 0,  
  "message": "成功"  
}
```

请求失败样例:

```
{  
  "data": "",  
  "code": 22,  
  "message": "无效参数"  
}
```

6.1.5 删除单个黑白名单

功能说明：删除黑白名单

1) API请求数据

DELETE https://{host}/api/v1/namespaces/@namespace/whiteblacklist/@url

2) 输入参数

- URL 参数

参数	类型	是否必选	描述
type	string	否	黑名单BLACK，白名单WHITE 枚举值: BLACK:黑名单 WHITE:白名单]

请求样例：

删除单个黑白名单

DELETE https://192.168.1.1/api/v1/namespaces/public/whiteblacklist/www.baidu.com?type=WHITE

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	黑白IP信息
data.url	string	主要是IP，域名，URL 最小长度:1 最大长度:1024
data.enable	bool	启用 true，禁用 false
data.type	string	黑名单 BLACK，白名单 WHITE 枚举值: BLACK:黑名单 WHITE:白名单]
data.domain	uint32	0 IP, 1 域名, 3 URL
data.isDefault	bool	内置 true 自定义 false
data.createTime	string	创建时间
data.description	string	描述 最小长度:0 最大长度:95

请求成功样例：

```
{
  "data": {
    "enable": true,
    "url": "www.baidu.com",
    "createTime": "2011-07-01 08:30:00",
```

```
{
  "isDefault": true,
  "type": "WHITE",
  "domain": true,
  "uuid": "07150a520899461e956bd9f0a69ea82d",
  "description": "www.baidu.com"
},
{
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

6.1.6 删除自定义黑白名单

功能说明: 删除自定义黑白名单

1) API请求数据

DELETE https://{host}/api/v1/namespaces/@namespace/whiteblacklist

2) 输入参数

- URL 参数

参数	类型	是否必选	描述
deleteAll	bool	否	黑名单BLACK, 白名单WHITE

请求样例:

清除所有自定义黑名单

DELETE https://192.168.1.1/api/v1/namespaces/public/whiteblacklist?deleteAll=true

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	黑白IP信息
data.url	string	主要是IP, 域名, URL 最小长度:1 最大长度:1024
data.enable	bool	启用 true, 禁用 false
data.type	string	黑名单 BLACK, 白名单 WHITE 枚举值: BLACK:黑名单 WHITE:白名单]
data.domain	uint32	0 IP, 1 域名, 3 URL
data.isDefault	bool	内置 true 自定义 false
data.createTime	string	创建时间

data.description	string	描述 最小长度:0 最大长度:95
------------------	--------	-------------------------

请求成功样例:

```
{
  "data": [],
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

6.1.7 批量删除黑白名单信息

功能说明: 批量删除黑白名单

1) API请求数据

POST

https://{host}/api/batch/v1/namespaces/@namespace/whiteblacklist?_method=delete

2) 输入参数

• Body参数

参数	类型	是否必选	描述
obj	object[]	是	黑白名单列表
obj.url	string	是	主要是IP, 域名, URL 最小长度:1 最大长度:1024
obj.enable	bool	否	启用 true, 禁用 false 默认值:true
obj.type	string	是	黑名单 BLACK, 白名单 WHITE 枚举值:[BLACK:黑名单 WHITE:白名单]
obj.description	string	否	描述 最小长度:0 最大长度:95

请求样例:

批量删除黑白名单

```
DELETE https://192.168.1.1/api/batch/v1/namespaces/public/whiteblacklist
[
  {
    "url": "www.baidu.com",
    "type": "WHITE"
  }
]
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	黑白名单列表
data.url	string	主要是IP， 域名, URL 最小长度:1 最大长度:1024
data.enable	bool	启用 true, 禁用 false
data.type	string	黑名单 BLACK, 白名单 WHITE 枚举值:[BLACK:黑名单 WHITE:白名单]
data.domain	uint32	0 IP, 1 域名, 3 URL
data.isDefault	bool	内置 true 自定义 false
data.createTime	string	创建时间
data.description	string	描述 最小长度:0 最大长度:95

请求成功样例:

```
{
  "data": {
    "itemsOffset": 0,
    "items": [
      {
        "enable": true,
        "url": "www.baidu.com",
        "createTime": "2011-07-01 08:30:00",
        "isDefault": true,
        "type": "WHITE",
        "domain": true,
        "uuid": "07150a520899461e956bd9f0a69ea82d",
        "description": "www.baidu.com"
      }
    ],
    "totalItems": 1,
    "pageNumber": 0,
    "pageSize": 200,
    "itemLength": 1,
    "totalPages": 1
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```


6.1.8 修改黑名单配置

功能说明：修改黑白名单

1) API请求数据

PATCH https://{host}/api/v1/namespaces/@namespace/whiteblacklist/@url

2) 输入参数

- Body参数

参数	类型	是否必选	描述
obj	object	是	黑白IP信息
obj.url	string	是	主要是IP，域名，URL 最小长度:1 最大长度:1024
obj.enable	bool	否	启用 true, 禁用 false 默认值:true
obj.type	string	是	黑名单 BLACK, 白名单 WHITE 枚举值:[BLACK:黑名单 WHITE:白名单]
obj.description	string	否	描述 最小长度:0 最大长度:95

请求样例：

修改单个黑白名单

```
PATCH https://192.168.1.1/api/v1/namespaces/public/whiteblacklist/www.baidu.com
{
  "enable": true,
  "type": "WHITE",
  "url": "www.baidu.com",
  "description": "baidu"
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	黑白IP信息
data.url	string	主要是IP，域名，URL 最小长度:1 最大长度:1024
data.enable	bool	启用 true, 禁用 false

data.type	string	黑名单 BLACK, 白名单 WHITE 枚举值: BLACK:黑名单 WHITE:白名单]
data.domain	uint32	0 IP, 1 域名, 3 URL
data.isDefault	bool	内置 true 自定义 false
data.createTime	string	创建时间
data.description	string	描述 最小长度:0 最大长度:95

请求成功样例:

```
{
  "data": {
    "enable": true,
    "url": "www.baidu.com",
    "createTime": "2011-07-01 08:30:00",
    "isDefault": true,
    "type": "WHITE",
    "domain": true,
    "uuid": "07150a520899461e956bd9f0a69ea82d",
    "description": "baidu"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

6.1.9 批量修改黑白名单信息

功能说明: 通过此接口, 批量修改黑白名单信息

1) API请求数据

PATCH https://{host}/api/batch/v1/namespaces/@namespace/whiteblacklist

2) 输入参数

• Body参数

参数	类型	是否必选	描述
obj	object[]	是	黑白名单列表
obj.url	string	是	主要是IP, 域名, URL 最小长度:1 最大长度:1024
obj.enable	bool	否	启用 true, 禁用 false 默认值:true

obj.type	string	是	黑名单 BLACK, 白名单 WHITE 枚举值: BLACK:黑名单 WHITE:白名单]
obj.description	string	否	描述 最小长度:0 最大长度:95

请求样例:

批量修改黑白名单

PATCH https://192.168.1.1/api/batch/v1/namespaces/public/whiteblacklist

```
[  
  {  
    "enable": true,  
    "type": "WHITE",  
    "url": "www.baidu.com",  
    "description": "baidu"  
  }  
]
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	黑白名单列表
data.url	string	主要是IP, 域名, URL 最小长度:1 最大长度:1024
data.enable	bool	启用 true, 禁用 false
data.type	string	黑名单 BLACK, 白名单 WHITE 枚举值: BLACK:黑名单 WHITE:白名单]
data.domain	uint32	0 IP, 1 域名, 3 URL
data.isDefault	bool	内置 true 自定义 false
data.createTime	string	创建时间
data.description	string	描述 最小长度:0 最大长度:95

请求成功样例:

```
{  
  "data": {  
    "itemsOffset": 0,  
    "items": [  
      {  
        "enable": true,  
        "url": "www.baidu.com",  
        "createTime": "2011-07-01 08:30:00",  
        "isDefault": true,  
      }  
    ]  
  }  
}
```

```

      "type": "WHITE",
      "domain": true,
      "uuid": "07150a520899461e956bd9f0a69ea82d",
      "description": "baidu"
    },
    "totalItems": 1,
    "pageNumber": 0,
    "pageSize": 200,
    "itemLength": 1,
    "totalPages": 1
  },
  "code": 0,
  "message": "成功"
}

```

请求失败样例:

```

{
  "data": "",
  "code": 22,
  "message": "无效参数"
}

```

6.2 封锁攻击者IP

6.2.1 获取封锁攻击者IP列表

功能说明: 获取封锁攻击者IP列表

1) API请求数据

GET <https://{host}/api/v1/namespaces/@namespace/blockip>

2) 输入参数

- URL参数

参数	类型	是否必选	描述
_sortby	string	否	指定排序字段 最大长度:100
creator	string	否	限制访问身份(默认为AF) 默认值:true 枚举值:[SIP:SIP身份调用 AF:AF身份调用]
_order	string	否	指定排序方式(正序/倒序) 默认值:true 枚举值:[asc:按升序排列 desc:按降序排列]
fuzzyIP	string	否	模糊搜索过滤IP, 若不填该字段, 则不进行过滤 最大长度:15

_length	int32	否	获取最大长度, 限制最大值, 如果数据不够可能返回小于 length 最小值:1 最大值:200 默认值:true
_start	int32	否	起始位置, 从 0 开始 最小值:0 最大值:2000000

请求样例:

获取封锁攻击者IP列表

GET https://192.168.1.1/api/v1/namespaces/public/blockip

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	分页数据
data.totalItems	int32	总共多少项目
data.totalPages	int32	总共多少页
data.pageNumber	int32	当前页码, 从 1 开始
data.pageSize	int32	每页多大
data.itemsOffset	int32	当前条目偏移, 从 0 开始
data.itemLength	int32	数据列表长度
data.items	object[]	有效数据列表
data.items.srcIP	string	源IP 格式:ip-address
data.items.dstIP	string	目的IP 格式:ip-address
data.items.dstPort	int32	目的端口 最小值:0 最大值:65535
data.items.blockTime	string	封锁时间
data.items.deblockTime	string	剩余封锁时间
data.items.module	string	触发封锁的安全模块

data.items.attack	string	触发封锁的攻击类型 枚举值: NULL:手动添加, 无攻击类型 WAF:WAF攻击 IPS:IPS攻击 UTM:UTM攻击 IP-CT:IP-CT攻击 DOS:DOS攻击 PLT-AUTO:安全感知平台自动策略 PLT-MANUAL:安全感知平台手动策略 CLOUD-IP-PROTECT:stateCenter:block CLOUD-IP-PROTECT PLT-HONEYPOT:stateCenter:block PLT-HONEYPOT PLT-SIP:stateCenter:block PLT-SIP PLT-MSS:stateCenter:block PLT-MSS CLOUD-IP-QUERY:stateCenter:block CLOUD-IP-QUERY WAF-CDN:stateCenter:block WAF-CDN]
data.items.policy	string	触发封锁的策略
data.items.enableLog	bool	是否有日志详情
data.items.blockType	string	stateCenter:block addr type
data.items.blockAddr	string	stateCenter:block addr
data.items.blockTimeLen	int32	stateCenter:block time len
data.items.blockScope	string	stateCenter:block scope

请求成功样例:

```
{
  "data": {
    "itemsOffset": 0,
    "items": [
      {
        "deblockTime": "23:56:55",
        "policy": "",
        "attack": "NULL",
        "dstIP": "0.0.0.0",
        "module": "手动添加规则",
        "enableLog": false,
        "srcIP": "192.168.1.1",
        "blockTime": "2020-07-15 10:08:47",
        "dstPort": 0
      }
    ],
    "totalItems": 1,
    "pageNumber": 1,
    "itemLength": 1,
    "totalPages": 1,
    "pageSize": 100
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

6.2.2 批量添加封锁攻击者

功能说明：批量添加封锁攻击者

1) API请求数据

POST https://{host}/api/batch/v1/namespaces/@namespace/blockip

2) 输入参数

• URL参数

参数	类型	是否必选	描述
aifwType	string	否	安全感知平台自动添加或手动添加,creator为SIP才生效(默认为MANUAL) 默认值:true 枚举值:[AUTO:自动 MANUAL:手动]
creator	string	否	限制访问身份(默认为AF) 默认值:true 枚举值:[SIP:SIP身份调用 AF:AF身份调用]

• Body参数

参数	类型	是否必选	描述
obj	object	是	批量添加封锁IP
obj.ipType	string	否	选择以源IP或目的IP进行封锁(默认为SRC) 默认值:true 枚举值:[SRC:源IP DST:目的IP DNS:stateCenter:block dns URL:stateCenter:block url]
obj.srcIP	string[]	是	以源IP地址列表添加封锁,需指定ipType为SRC才生效
obj.dstIP	string[]	是	以目的IP地址列表添加封锁,需指定ipType为DST才生效
obj.blockTime	string	是	时间(带时间单位m/h/d)
obj.dns	string[]	是	stateCenter:block dns
obj.url	string[]	是	stateCenter:block url

请求样例:

批量添加封锁攻击者IP, 添加源ip192.168.1.2、192.168.1.3, 封锁时间为3天

```
POST https://192.168.1.1/api/batch/v1/namespaces/public/blockip
{
  "ipType": "SRC",
  "srcIP": [
```

```

    "192.168.1.2",
    "192.168.1.3"
  ],
  "blockTime": "3d"
}

```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	批量添加封锁IP
data.ipType	string	选择以源IP或目的IP进行封锁(默认为SRC) 枚举值:[SRC:源IP DST:目的IP DNS:stateCenter:block dns URL:stateCenter:block url]
data.srcIP	string[]	以源IP地址列表添加封锁,需指定ipType为SRC才生效
data.dstIP	string[]	以目的IP地址列表添加封锁,需指定ipType为DST才生效
data.blockTime	string	时间(带时间单位m/h/d)
data.dns	string[]	stateCenter:block dns
data.url	string[]	stateCenter:block url

请求成功样例:

```

{
  "data": {
    "ipType": "SRC",
    "srcIP": [
      "192.168.1.2",
      "192.168.1.3"
    ],
    "blockTime": "3d"
  },
  "code": 0,
  "message": "成功"
}

```

请求失败样例:

```

{
  "data": "",
  "code": 22,
  "message": "无效参数"
}

```

6.2.3 批量删除封锁攻击者

功能说明: 批量删除封锁攻击者

1) API请求数据

POST

https://{host}/api/batch/v1/namespaces/@namespace/blockip?_method=delete

2) 输入参数

• URL参数

参数	类型	是否必选	描述
creator	string	否	限制访问身份(默认为AF) 默认值:true 枚举值:[SIP:SIP身份调用 AF:AF身份调用]

• Body参数

参数	类型	是否必选	描述
obj	object[]	是	要删除的封锁IP列表
obj.srcIP	string	否	源IP 格式:ip-address
obj.dstIP	string	否	目的IP 格式:ip-address
obj.dstPort	int32	否	目的端口 最小值:0 最大值:65535
obj.attack	string	否	触发封锁的攻击类型 默认值:true 枚举值:[NULL:手动添加, 无攻击类型 WAF:WAF攻击 IPS:IPS攻击 UTM:UTM攻击 IP-CT:IP-CT攻击 DOS:DOS攻击 PLT-AUTO:安全感知平台自动策略 PLT-MANUAL:安全感知平台手动策略 CLOUD-IP-PROTECT:stateCenter:block CLOUD-IP-PROTECT PLT-HONEYPOT:stateCenter:block PLT-HONEYPOT PLT-SIP:stateCenter:block PLT-SIP PLT-MSS:stateCenter:block PLT-MSS CLOUD-IP-QUERY:stateCenter:block CLOUD-IP-QUERY WAF-CDN:stateCenter:block WAF-CDN]
obj.dns	string	否	stateCenter:block dns 最小长度:1 最大长度:4096
obj.url	string	否	stateCenter:block url 最小长度:1 最大长度:4096

obj.scope	string	否	stateCenter:block scope 枚举值:[GLOBAL:stateCenter:global block BUSINESS:stateCenter:business block]
-----------	--------	---	--

请求样例:

批量删除封锁攻击者IP, 192.168.1.1、192.168.1.2、192.168.1.3

POST https://192.168.1.1/api/batch/v1/namespaces/public/blockip?_method=DELETE

```
[
  {
    "srcIP": "192.168.1.1",
    "attack": "PLT-MANUAL"
  },
  {
    "srcIP": "192.168.1.2",
    "attack": "PLT-MANUAL"
  },
  {
    "srcIP": "192.168.1.3",
    "attack": "PLT-MANUAL"
  }
]
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	要删除的封锁IP列表
data.srcIP	string	源IP 格式:ip-address
data.dstIP	string	目的IP 格式:ip-address
data.dstPort	int32	目的端口 最小值:0 最大值:65535

data.attack	string	触发封锁的攻击类型 枚举值: NULL:手动添加, 无攻击类型 WAF:WAF攻击 IPS:IPS攻击 UTM:UTM攻击 IP-CT:IP-CT攻击 DOS:DOS攻击 PLT-AUTO:安全感知平台自动策略 PLT-MANUAL:安全感知平台手动策略 CLOUD-IP-PROTECT:stateCenter:block CLOUD-IP-PROTECT PLT-HONEYPOT:stateCenter:block PLT-HONEYPOT PLT-SIP:stateCenter:block PLT-SIP PLT-MSS:stateCenter:block PLT-MSS CLOUD-IP-QUERY:stateCenter:block CLOUD-IP-QUERY WAF-CDN:stateCenter:block WAF-CDN]
data.dns	string	stateCenter:block dns 最小长度:1 最大长度:4096
data.url	string	stateCenter:block url 最小长度:1 最大长度:4096
data.scope	string	stateCenter:block scope 枚举值: GLOBAL:stateCenter:global block BUSINESS:stateCenter:business block]

请求成功样例:

```
{
  "data": [
    {
      "srcIP": "192.168.1.1",
      "attack": "PLT-MANUAL"
    },
    {
      "srcIP": "192.168.1.2",
      "attack": "PLT-MANUAL"
    },
    {
      "srcIP": "192.168.1.3",
      "attack": "PLT-MANUAL"
    }
  ],
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

6.2.4 清空封锁攻击者

功能说明：清空封锁攻击者

1) API请求数据

DELETE

https://{host}/api/v1/namespaces/@namespace/blockipclear

2) 输入参数

- URL 参数

参数	类型	是否必选	描述
creator	string	否	限制访问身份(默认为AF) 默认值:true 枚举值:[SIP:SIP身份调用 AF:AF身份调用]

请求样例：

清空封锁攻击者IP

DELETE https://192.168.1.1/api/v1/namespaces/public/blockipclear

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	分页数据
data.totalItems	int32	总共多少项目
data.totalPages	int32	总共多少页
data.pageNumber	int32	当前页码，从 1 开始
data.pageSize	int32	每页多大
data.itemsOffset	int32	当前条目偏移, 从 0 开始
data.itemLength	int32	数据列表长度
data.items	object[]	有效数据列表
data.items.srcIP	string	源IP 格式:ip-address
data.items.dstIP	string	目的IP 格式:ip-address
data.items.dstPort	int32	目的端口 最小值:0 最大值:65535
data.items.blockTime	string	封锁时间
data.items.deblockTime	string	剩余封锁时间
data.items.module	string	触发封锁的安全模块

data.items.attack	string	触发封锁的攻击类型 枚举值: NULL:手动添加, 无攻击类型 WAF:WAF攻击 IPS:IPS攻击 UTM:UTM攻击 IP-CT:IP-CT攻击 DOS:DOS攻击 PLT-AUTO:安全感知平台自动策略 PLT-MANUAL:安全感知平台手动策略 CLOUD-IP-PROTECT:stateCenter:block CLOUD-IP-PROTECT PLT-HONEYPOT:stateCenter:block PLT-HONEYPOT PLT-SIP:stateCenter:block PLT-SIP PLT-MSS:stateCenter:block PLT-MSS CLOUD-IP-QUERY:stateCenter:block CLOUD-IP-QUERY WAF-CDN:stateCenter:block WAF-CDN]
data.items.policy	string	触发封锁的策略
data.items.enableLog	bool	是否有日志详情
data.items.blockType	string	stateCenter:block addr type
data.items.blockAddr	string	stateCenter:block addr
data.items.blockTimeLen	int32	stateCenter:block time len
data.items.blockScope	string	stateCenter:block scope

请求成功样例:

```
{
  "data": {
    "itemsOffset": 0,
    "items": [],
    "totalItems": 0,
    "pageNumber": 1,
    "itemLength": 0,
    "totalPages": 1,
    "pageSize": 50
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

6.2.5 全量修改自动封锁攻击者时间

功能说明: 设置自动封锁攻击者时间

1) API请求数据

PATCH https://{host}/api/v1/namespaces/@namespace/blockiptime

2) 输入参数

- Body参数

参数	类型	是否必选	描述
obj	object	是	封锁攻击者IP时间配置
obj.blockTime	string	是	时间值
obj.minutes	uint32	否	分钟数(自动生成)
obj.bruteBlockTime	string	否	暴力破解

请求样例:

修改自动封锁攻击者时间为1天

```
PATCH https://192.168.1.1/api/v1/namespaces/public/blockiptime
{
  "blockTime": "1d"
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	封锁攻击者IP时间配置
data.blockTime	string	时间值
data.minutes	uint32	分钟数(自动生成)
data.bruteBlockTime	string	暴力破解

请求成功样例:

```
{
  "data": {
    "blockTime": "1d",
    "minutes": 1440
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

6.2.6 获取自动封锁攻击者时间

功能说明: 获取自动封锁攻击者时间

1) API请求数据

GET

https://{host}/api/v1/namespaces/@namespace/blockiptime

2) 输入参数

请求样例:

获取自动封锁攻击者时间

PATCH https://192.168.1.1/api/v1/namespaces/public/blockiptime

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	封锁攻击者IP时间配置
data.blockTime	string	时间值
data.minutes	uint32	分钟数(自动生成)
data.bruteBlockTime	string	暴力破解

请求成功样例:

```
{
  "data": {
    "blockTime": "1d",
    "minutes": 1440
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

7 状态中心

7.1 总览

7.1.1 设备信息

7.1.1.1 获取内存使用率

功能说明：获取内存使用率

1) API请求数据

GET

https://{host}/api/v1/namespaces/@namespace/memoryusage

2) 输入参数

请求样例：

获取内存使用率

GET https://192.168.1.1/api/v1/namespaces/public/memoryusage

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	内存利用率
data.memoryUsage	uint32	内存利用率 最小值:0 最大值:100

请求成功样例：

```
{
  "data": {
    "memoryUsage": 25
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

7.1.1.2 获取内存使用率变化曲线数据

功能说明：获取内存使用率变化曲线数据

1) API请求数据

GET `https://{host}/api/v1/namespaces/@namespace/memoryusages`

2) 输入参数

- URL 参数

参数	类型	是否必选	描述
timeFilter	string	否	最近多长时间段内的内存实时使用率(默认为REAL-TIME) 默认值:true 枚举值:[REAL-TIME:最近的内存实时使用率(时间间隔为10秒) 24HOUR:最近24小时的内存实时使用率(时间间隔为10分钟) 7DAY:最近7天的内存实时使用率(时间间隔为72分钟)]

请求样例:

获取实时内存使用率变化曲线

GET `https://192.168.1.1/api/v1/namespaces/public/memoryusages`

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	内存使用率曲线数据
data.memoryUsages	object[]	内存实时使用率数组
data.memoryUsages.usage	uint32	内存实时使用率 最小值:0 最大值:100
data.memoryUsages.time	string	日期时间

请求成功样例:

```
{
  "data": {
    "memoryUsages": [
      {
        "time": "2020-06-03 20:24:40",
        "usage": 25
      },
      {
        "time": "2020-06-03 20:24:30",
        "usage": 25
      },
      {
        "time": "2020-06-03 20:24:20",
        "usage": 25
      },
      {
        "time": "2020-06-03 20:24:10",

```

```

      "usage": 25
    },
    {
      "time": "2020-06-03 20:24:00",
      "usage": 25
    },
    {
      "time": "2020-06-03 20:23:50",
      "usage": 25
    },
    {
      "time": "2020-06-03 20:23:40",
      "usage": 25
    },
    {
      "time": "2020-06-03 20:23:30",
      "usage": 25
    },
    {
      "time": "2020-06-03 20:23:20",
      "usage": 25
    },
    {
      "time": "2020-06-03 20:23:10",
      "usage": 25
    }
  ],
  "code": 0,
  "message": "成功"
}

```

请求失败样例：

```

{
  "data": "",
  "code": 22,
  "message": "无效参数"
}

```

7.1.1.3 获取cpu使用率

功能说明：获取cpu使用率

1) API请求数据

GET <https://{host}/api/v1/namespaces/@namespace/cpuusage>

2) 输入参数

请求样例：

获取cpu使用率

GET <https://192.168.1.1/api/v1/namespaces/public/cpuusage>

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	cpu使用率数据
data.cpuAverage	string	总CPU平均使用率
data.cpuCurrent	string	总CPU当前使用率
data.cpuDetail	string[]	各个CPU使用率

data.cpu1minute	string	总CPU1分钟平均使用率
data.cpu5minute	string	总CPU5分钟平均使用率
data.cpu15minute	string	总CPU15分钟平均使用率

请求成功样例:

```
{
  "data": {
    "cpu5minute": "9.10",
    "cpu15minute": "11.27",
    "cpu1minute": "12.33",
    "cpuAverage": "10.81",
    "cpuDetail": [
      "16.16",
      "13.28"
    ],
    "cpuCurrent": "14.71"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

7.1.1.4 获取cpu使用率变化曲线数据

功能说明: 获取cpu使用率变化曲线数据

1) API请求数据

GET

https://{host}/api/v1/namespaces/@namespace/cpuusages

2) 输入参数

• URL参数

参数	类型	是否必选	描述
timeFilter	string	否	最近多长时间段内的cpu实时使用率(默认为REAL-TIME) 默认值:true 枚举值:[REAL-TIME:最近的cpu实时使用率(时间间隔为10秒) 24HOUR:最近24小时的cpu实时使用率(时间间隔为10分钟) 7DAY:最近7天的cpu实时使用率(时间间隔为72分钟)]

请求样例:

获取实时cpu使用率变化曲线数据

GET https://192.168.1.1/api/v1/namespaces/public/cpuusages

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	cpu使用率变化曲线数据
data.cpuUsages	object[]	cpu使用率曲线数据
data.cpuUsages.usage	uint32	cpu实时使用率 最小值:0 最大值:100
data.cpuUsages.time	string	日期时间

请求成功样例:

```
{
  "data": {
    "cpuUsages": [
      {
        "time": "2020-06-03 20:24:40",
        "usage": 9
      },
      {
        "time": "2020-06-03 20:24:30",
        "usage": 9
      },
      {
        "time": "2020-06-03 20:24:20",
        "usage": 10
      },
      {
        "time": "2020-06-03 20:24:10",
        "usage": 9
      },
      {
        "time": "2020-06-03 20:24:00",
        "usage": 8
      },
      {
        "time": "2020-06-03 20:23:50",
        "usage": 9
      },
      {
        "time": "2020-06-03 20:23:40",
        "usage": 9
      },
      {
        "time": "2020-06-03 20:23:30",
        "usage": 9
      },
      {
        "time": "2020-06-03 20:23:20",
        "usage": 10
      },
      {
        "time": "2020-06-03 20:23:10",
        "usage": 10
      }
    ]
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
```

```
{
  "code": 22,
  "message": "无效参数"
}
```

7.1.1.5 获取设备运行时间

功能说明：获取设备运行时间

1) API请求数据

GET `https://{host}/api/v1/namespaces/@namespace/uptimes`

2) 输入参数

请求样例：

获取设备运行时间

GET `https://192.168.1.1/api/v1/namespaces/public/uptimes`

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	机器持续运行时间数据
data.upTimes	string	机器持续运行时间数据

请求成功样例：

```
{
  "data": {
    "upTimes": "3小时"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

7.1.1.6 获取当前时间

功能说明：获取当前系统时间，显示格式如：2019-11-11 23:59:59

1) API请求数据

GET `https://{host}/api/v1/namespaces/@namespace/currentdatetime`

2) 输入参数

请求样例：

获取当前时间

GET `https://192.168.1.1/api/v1/namespaces/public/currentdatetime`

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	系统时间
data.date	string	系统日期
data.time	string	系统时间
data.weekday	string	星期几
data.zone	string	时区
data.server	string	时间服务器地址

请求成功样例:

```
{
  "data": {
    "server": "pool.ntp.org",
    "weekday": "周三",
    "date": "2020-06-03",
    "time": "20:25:36",
    "zone": "(GMT+08:00)北京,上海,香港"
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

7.1.1.7 获取磁盘信息数据

功能说明: 获取磁盘信息数据, 包括磁盘已使用、未使用空间大小, 磁盘状态, 磁盘使用率等信息

1) API请求数据

GET <https://{host}/api/v1/namespaces/@namespace/diskstatus>

2) 输入参数

请求样例:

获取磁盘信息数据

GET <https://192.168.1.1/api/v1/namespaces/public/diskstatus>

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息

data	object	磁盘信息
data.diskStatus	object	设备磁盘状态
data.diskStatus.code	uint32	设备磁盘状态id 0: 硬盘稳定 1: L1级别告警 2: L2级别告警 3: L3级别告警 4: 硬盘不存在 5: 硬盘没有初始化 6: 硬盘不支持smart技术 7: 系统没有运行在CF卡上 8: fwlog分区没有挂载成功 9: ssd磁盘寿命低于告警门限标志 10: 硬盘IO占用率过高
data.diskStatus.msg	string	设备磁盘状态信息
data.rootDiskUnusedSize	uint32	根磁盘(/dev/root)未使用空间大小
data.diskUsage	uint32	返回硬盘使用率 最小值:0 最大值:100
data.diskUnusedSize	uint32	总磁盘未使用空间大小
data.diskUsedSize	uint32	总磁盘已使用空间大小

请求成功样例：

```
{
  "data": {
    "diskStatus": {
      "code": 7,
      "msg": "系统没有运行在CF卡上"
    },
    "rootDiskUnusedSize": 7383576,
    "diskUsage": 13,
    "diskUsedSize": 8753743,
    "diskUnusedSize": 56910728
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

7.1.1.8 获取硬盘使用率

功能说明：获取硬盘使用率，返回数值为百分率，如：50代表50%

1) API请求数据

GET

https://{host}/api/v1/namespaces/@namespace/diskusage

2) 输入参数

请求样例:

```
获取硬盘使用率
GET https://192.168.1.1/api/v1/namespaces/public/diskusage
```

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	返回硬盘使用率
data.diskUsage	uint32	返回硬盘使用率 最小值:0 最大值:100

请求成功样例:

```
{
  "data": {
    "diskUsage": 13
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

7.1.1.9 获取系统版本信息

功能说明: 获取系统版本信息

1) API请求数据

```
GET https://{host}/api/v1/namespaces/@namespace/systemversion
```

2) 输入参数

• URL参数

参数	类型	是否必选	描述
----	----	------	----

filter	string	否	版本信息选择(默认为ALL) 默认值:true 枚举值:[ALL:显示所有版本信息 EN:判断是否为EN版本 R:获取R版本号 HF:判断是否为HF版本 B:判断是否为Beta版本 BUILD:显示创建日期 ADD:显示附加版本信息 MAJOR:显示主版本号 MINOR:显示次版本号 INCREASE:显示增版本号 FULL:显示完整版本号]
--------	--------	---	--

请求样例:

获取硬盘使用率

GET https://192.168.1.1/api/v1/namespaces/public/systemversion

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	系统版本信息
data.full	string	AF版本全名
data.major	uint32	主版本号
data.minor	uint32	次版本号
data.increase	uint32	增版本号
data.build	string	创建日期
data.EN	bool	是否为英文版本
data.B	bool	是否为Beta版本
data.HF	bool	是否为HF版本
data.R	string	R版本号
data.addInfo	string[]	附加版本信息列表

请求成功样例:

```
{
  "data": {
    "EN": false,
    "build": "Build20200603",
    "R": "",
    "full": "AF8.0.25.4401",
    "addInfo": [],
    "major": 8,
    "increase": 25,
    "minor": 0,
    "B": false,
    "HF": false
  },
}
```

```
    "code": 0,
    "message": "成功"
  }
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

7.1.1.10 获取接口状态信息

功能说明: 提供网口名, 获取对应接口状态信息, 信息包括流速和网口连接状态

1) API请求数据

GET	https://{host}/api/v1/namespaces/@namespace/interfacestatus/@interfaceNames
-----	---

2) 输入参数

- URL参数

参数	类型	是否必选	描述
interfaceNames	string	否	接口名列表, 用逗号分隔, 如eth0,eth1,eth2 最大长度:1024

请求样例:

获取eth0网口状态

GET https://192.168.1.1/api/v1/namespaces/public/interfacestatus/eth0

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	接口状态列表
data.interfaceName	string	接口名,如eth0
data.information	object	接口信息
data.information.connectionStatus	bool	接口连接状态,true为连接,false为断开
data.information.speed	object	接口吞吐率
data.information.speed.recv	uint64	接收速率
data.information.speed.send	uint64	发送速率
data.information.bagSpeed	object	接口包收发速率
data.information.bagSpeed.recvbag	uint64	包接收速率

data.information.bagSpeed.sendbag	uint64	包发送速率
data.information.duplexSpeed	uint64	双工速率
data.information.duplex	string	双工类型
data.information.supportedPorts	string	支持光口或电口 枚举值:[TP:电口 FIBRE:光口]
data.information.counter	object	接口统计信息
data.information.counter.tx_packets	uint64	发包数
data.information.counter.rx_packets	uint64	收包数
data.information.counter.rx_bytes	uint64	接收字节
data.information.counter.tx_bytes	uint64	发送字节
data.information.counter.tx_dropped	uint64	发送丢包数
data.information.counter.rx_dropped	uint64	接收丢包数

请求成功样例：

```
{
  "data": [
    {
      "information": {
        "duplex": "full",
        "counter": {
          "tx_dropped": 0,
          "rx_packets": 12977295,
          "tx_packets": 146167,
          "tx_bytes": 44265791,
          "rx_dropped": 0,
          "rx_bytes": 854615690
        },
        "bagSpeed": {
          "sendbag": 27,
          "recvbag": 852
        },
        "connectStatus": true,
        "supportedPorts": "TP",
        "speed": {
          "send": 8440,
          "recv": 55007
        },
        "duplexSpeed": 1000
      },
      "interfaceName": "eth0"
    }
  ],
  "code": 0,
  "message": "成功"
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

7.1.2 网络活动状态

7.1.2.1 获取接口吞吐率折线图

功能说明：获取接口吞吐率折线图数据，可选择流速单位、时间段、网口名进行数据过滤

1) API请求数据

GET https://{host}/api/v1/namespaces/@namespace/interfacetraffics

2) 输入参数

• URL 参数

参数	类型	是否必选	描述
flowUnit	string	否	流量单位(BYTES或BITS),默认为BITS 默认值:true 枚举值:[BYTES:流量单位(比特每秒) BITS:流量单位(比特每秒)]
timeFilter	string	否	请求获取最近多长时间的数据(默认为REAL-TIME) 默认值:true 枚举值:[REAL-TIME:最近的接口吞吐率趋势 24HOUR:最近24小时的接口吞吐率趋势(时间间隔为10分钟) 7DAY:最近7天的接口吞吐率趋势(时间间隔为72分钟)]
interfaceName	string	否	要过滤的网口(默认为ALL-WAN) 默认值:true

请求样例：

获取实时接口吞吐率折线图

GET https://192.168.1.1/api/v1/namespaces/public/interfacetraffics

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	接口吞吐率折线图数据
data.speeds	object[]	接口吞吐率列表

data.speeds.sendSpeed	uint64	发送速率
data.speeds.recvSpeed	uint64	接收速率
data.speeds.time	string	日期时间
data.unit	string	接口吞吐率单位(MB/s KB/s B/s)
data.realTimeSpeed	object	实时速率
data.realTimeSpeed.sendSpeed	string	实时发送速率, 如1024Kbps
data.realTimeSpeed.recvSpeed	string	实时接收速率, 如1024Kbps

请求成功样例:

```
{
  "data": {
    "speeds": [
      {
        "recvSpeed": 0,
        "time": "2020-06-03 20:25:27",
        "sendSpeed": 0
      },
      {
        "recvSpeed": 0,
        "time": "2020-06-03 20:25:25",
        "sendSpeed": 0
      },
      {
        "recvSpeed": 0,
        "time": "2020-06-03 20:25:23",
        "sendSpeed": 0
      },
      {
        "recvSpeed": 0,
        "time": "2020-06-03 20:25:21",
        "sendSpeed": 0
      },
      {
        "recvSpeed": 0,
        "time": "2020-06-03 20:25:19",
        "sendSpeed": 0
      },
      {
        "recvSpeed": 0,
        "time": "2020-06-03 20:25:17",
        "sendSpeed": 0
      },
      {
        "recvSpeed": 0,
        "time": "2020-06-03 20:25:15",
        "sendSpeed": 0
      },
      {
        "recvSpeed": 0,
        "time": "2020-06-03 20:25:13",
        "sendSpeed": 0
      },
      {
        "recvSpeed": 0,
        "time": "2020-06-03 20:25:11",
        "sendSpeed": 0
      },
      {
        "recvSpeed": 0,
        "time": "2020-06-03 20:25:09",
        "sendSpeed": 0
      }
    ],
    "realTimeSpeed": {
      "sendSpeed": "0bps",

```

```
        "recvSpeed": "0bps"
      },
      "unit": "bps"
    },
    "code": 0,
    "message": "成功"
  }
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

7.2 监控

7.2.1 流量排行

7.2.1.1 获取应用流量排行

功能说明: 获取应用流量排行

1) API请求数据

GET

https://{host}/api/v1/namespaces/@namespace/topapptraffics

2) 输入参数

• URL参数

参数	类型	是否必选	描述
_sortBy	string	否	指定排序字段 最大长度:100
_length	int32	否	获取最大长度, 限制最大值, 如果数据不够可能返回小于 length 最小值:1 最大值:200 默认值:true
line	uint16	否	选择线路(0-256), 选择全部则值为0, 选择线路n则值为n(默认所有线路) 最小值:0 最大值:256
_order	string	否	指定排序方式 (正序/倒序) 默认值:true 枚举值:[asc:按升序排列 desc:按降序排列]
topNumber	int32	否	显示前N名的排行(默认top8,范围1-60) 最小值:1 最大值:60 默认值:true
_start	int32	否	起始位置, 从 0 开始 最小值:0 最大值:2000000

请求样例:

获取应用流量排行, 只显示前1名

GET https://192.168.1.1/api/v1/namespaces/public/topapptraffics?topNumber=1

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	分页数据
data.totalItems	int32	总共多少项目
data.totalPages	int32	总共多少页
data.pageNumber	int32	当前页码, 从 1 开始
data.pageSize	int32	每页多大
data.itemsOffset	int32	当前条目偏移, 从 0 开始
data.itemLength	int32	数据列表长度
data.items	object[]	有效数据列表
data.items.appType	string	应用类型
data.items.label	string	标签、应用描述
data.items.line	string	线路(0-256)
data.items.percentage	string	流量百分比,如: 50%
data.items.speed	object	流速
data.items.speed.up	uint64	上行流速
data.items.speed.down	uint64	下行流速
data.items.speed.total	uint64	总流速

请求成功样例:

```
{
  "data": {
    "itemsOffset": 0,
    "items": [
      {
        "line": "所有线路",
        "label": "-",
        "appType": "SSL-vpn",
        "percentage": "0.0%",
        "speed": {
          "down": 0,
          "up": 70,
          "total": 70
        }
      }
    ],
    "totalItems": 1,
    "pageNumber": 1,
    "itemLength": 1,
    "totalPages": 1,
    "pageSize": 100
  },
  "code": 0,
  "message": "成功"
}
```

```
}

```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

7.2.1.2 获取用户流量排行

功能说明: 获取用户流量排行,指定_sortby为total:表示使用总流速排行,up:表示上行流速,dow
n:表示下行流速,若不指定或指定错误,默认使用total排行

1) API请求数据

POST

https://{host}/api/v1/namespaces/@namespace/topusertraffics?_method=get

2) 输入参数

• URL参数

参数	类型	是否必选	描述
_length	int32	否	获取最大长度, 限制最大值, 如果数据不够可能返回小于 length 最小值:1 最大值:200 默认值:true
_sortby	string	否	指定排序字段 最大长度:100
_order	string	否	指定排序方式 (正序/倒序) 默认值:true 枚举值:[asc:按升序排列 desc:按降序排列]
_start	int32	否	起始位置, 从 0 开始 最小值:0 最大值:2000000

• Body参数

参数	类型	是否必选	描述
obj	object	是	用户流量请求过滤数据
obj.topNumber	int32	否	显示前N名的排行(默认为60,范围1-60) 最小值:1 最大值:60
obj.line	uint16	否	要过滤的线路,0代表全部(默认为0,范围0-256) 最小值:0 最大值:256
obj.applicationType	string[]	否	要过滤的应用类型列表, 若不选该字段, 则默认不过滤

obj.filterObject	object	否	选择过滤对象, 若不选择, 则默认不进行过滤, 返回全部
obj.filterObject.objectType	string	是	要过滤的对象类型 枚举值: GROUP:以用户组过滤 USER:以用户过滤 IP:以ip地址过滤]
obj.filterObject.groups	string[]	是	用户组列表,objectType需指定为GROUP才生效
obj.filterObject.users	string[]	是	用户名列表,objectType需指定为USER才生效
obj.filterObject.ip	string[]	是	IP列表,objectType需指定为IP才生效

请求样例:

获取用户流量排行, 只显示前1名

```
POST https://192.168.1.1/api/v1/namespaces/public/topusertraffics?_method=GET
{
  "topNumber": 1
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	分页数据
data.totalItems	int32	总共多少项目
data.totalPages	int32	总共多少页
data.pageNumber	int32	当前页码, 从 1 开始
data.pageSize	int32	每页多大
data.itemsOffset	int32	当前条目偏移, 从 0 开始
data.itemLength	int32	数据列表长度
data.items	object[]	有效数据列表
data.items.name	string	用户名
data.items.group	string	所属组
data.items.ip	string	ip地址 格式:ip-address
data.items.lockStatus	bool	是否冻结上网
data.items.speed	object	流速
data.items.speed.up	uint32	上行流速
data.items.speed.down	uint32	下行流速
data.items.speed.total	uint32	总流速

data.items.composition	object[]	流量构成列表
data.items.composition.percent	string	流量百分比,如: 50%
data.items.composition.applicationType	string	过滤的应用类型
data.items.composition.line	int32	所属线路
data.items.composition.speed	object	流速
data.items.composition.speed.up	uint32	上行流速
data.items.composition.speed.down	uint32	下行流速
data.items.composition.speed.total	uint32	总流速

请求成功样例:

```
{
  "data": {
    "itemsOffset": 0,
    "items": [
      {
        "lockStatus": true,
        "group": "/",
        "composition": [
          {
            "applicationType": "QQ游戏",
            "percent": "0",
            "line": 1,
            "speed": {
              "down": 346,
              "up": 1016,
              "total": 1362
            }
          }
        ]
      },
      {
        "applicationType": "P2PSrv",
        "percent": "92",
        "line": 1,
        "speed": {
          "down": 216029,
          "up": 4232556,
          "total": 4448585
        }
      },
      {
        "applicationType": "酷我音乐",
        "percent": "8",
        "line": 1,
        "speed": {
          "down": 283310,
          "up": 107223,
          "total": 390533
        }
      },
      {
        "applicationType": "其他",
        "percent": "0",
        "line": 1,
        "speed": {
          "down": 108,
          "up": 162,
          "total": 270
        }
      }
    ]
  }
}
```

```
    },
    "ip": "20.20.2.125",
    "name": "20.20.2.125",
    "speed": {
      "down": 499793,
      "up": 4340957,
      "total": 4840750
    }
  },
  "totalItems": 1,
  "pageNumber": 1,
  "itemLength": 1,
  "totalPages": 1,
  "pageSize": 100
},
"code": 0,
"message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

7.2.1.3 获取IP流量趋势

功能说明: 获取IP流量趋势曲线, 可指定前5或10名的IP流量趋势, 指定最近时间范围和流速单位获取相应的变化曲线

1) API请求数据

GET

https://{host}/api/v1/namespaces/@namespace/iptraffics

2) 输入参数

• URL 参数

参数	类型	是否必选	描述
speedType	string	否	流速类型,默认为TOTAL 默认值:true 枚举值:[UP:ip stateCenter:traffics UP DOWN:ip stateCenter:traffics DOWN TOTAL:ip stateCenter:traffics TOTAL]
ip	string	否	ip地址 格式:ip-address
timeFilter	string	否	最近多长时间段内的IP流量趋势,默认为1H OUR 默认值:true 枚举值:[1HOUR:最近1小时的IP流量趋势 12HOUR:最近12小时的IP流量趋势 24HOUR:最近24小时的IP流量趋势]

topNumber	string	否	显示前5名/前10名的流量趋势,默认为TOP 5 默认值:true 枚举值:[TOP5:前5名的流量趋势 TOP10:前10名的流量趋势]
-----------	--------	---	--

请求样例:

获取IP流量趋势

GET https://192.168.1.1/api/v1/namespaces/public/iptraffics

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	所有IP对应趋势数据
data.ip	string	IPv4或者IPv6地址 格式:ip-address
data.mac	string	MAC地址限制
data.trendData	object[]	IP流量实时数据队列
data.trendData.ipTraffic	uint64	IP实时流量
data.trendData.time	string	日期时间

请求成功样例:

```
{
  "data": [
    {
      "trendData": [
        {
          "time": "2020-07-15 15:38:00",
          "ipTraffic": 0
        },
        {
          "time": "2020-07-15 15:39:00",
          "ipTraffic": 1926862
        },
        {
          "time": "2020-07-15 15:40:00",
          "ipTraffic": 0
        },
        {
          "time": "2020-07-15 15:41:00",
          "ipTraffic": 0
        },
        {
          "time": "2020-07-15 15:42:00",
          "ipTraffic": 0
        }
      ],
      "ip": "20.20.2.162",
      "mac": "00-21-05-C9-E8-FD"
    },
    {
      "trendData": [
        {
          "time": "2020-07-15 15:38:00",
```

```
        "ipTraffic": 0
      },
      {
        "time": "2020-07-15 15:39:00",
        "ipTraffic": 0
      },
      {
        "time": "2020-07-15 15:40:00",
        "ipTraffic": 0
      },
      {
        "time": "2020-07-15 15:41:00",
        "ipTraffic": 1157861
      },
      {
        "time": "2020-07-15 15:42:00",
        "ipTraffic": 0
      }
    ],
    "ip": "20.20.2.34",
    "mac": "00-21-05-C9-E8-FD"
  },
  {
    "trendData": [
      {
        "time": "2020-07-15 15:38:00",
        "ipTraffic": 0
      },
      {
        "time": "2020-07-15 15:39:00",
        "ipTraffic": 0
      },
      {
        "time": "2020-07-15 15:40:00",
        "ipTraffic": 0
      },
      {
        "time": "2020-07-15 15:41:00",
        "ipTraffic": 0
      },
      {
        "time": "2020-07-15 15:42:00",
        "ipTraffic": 0
      }
    ],
    "ip": "20.20.2.88",
    "mac": "00-21-05-C9-E8-FD"
  },
  {
    "trendData": [
      {
        "time": "2020-07-15 15:38:00",
        "ipTraffic": 1384994
      },
      {
        "time": "2020-07-15 15:39:00",
        "ipTraffic": 0
      },
      {
        "time": "2020-07-15 15:40:00",
        "ipTraffic": 0
      },
      {
        "time": "2020-07-15 15:41:00",
        "ipTraffic": 1401540
      },
      {
        "time": "2020-07-15 15:42:00",
        "ipTraffic": 0
      }
    ],
    "ip": "20.20.2.99",
    "mac": "00-21-05-C9-E8-FD"
  },
  {
    "trendData": [
      {
```

```
{
  "time": "2020-07-15 15:38:00",
  "ipTraffic": 1077207
},
{
  "time": "2020-07-15 15:39:00",
  "ipTraffic": 0
},
{
  "time": "2020-07-15 15:40:00",
  "ipTraffic": 0
},
{
  "time": "2020-07-15 15:41:00",
  "ipTraffic": 0
},
{
  "time": "2020-07-15 15:42:00",
  "ipTraffic": 0
},
},
{
  "ip": "20.20.2.85",
  "mac": "00-21-05-C9-E8-FD"
},
},
{
  "code": 0,
  "message": "成功"
}
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

7.2.2 会话排行

7.2.2.1 获取会话数量排行

功能说明: 获取会话数量排行,可指定排行方式为总会话排行或新建会话排行, 可指定获取前10、20、30、60名排行

1) API请求数据

GET <https://{host}/api/v1/namespaces/@namespace/topsessionnumbers>

2) 输入参数

• URL 参数

参数	类型	是否必选	描述
_length	int32	否	获取最大长度, 限制最大值, 如果数据不够可能返回小于 length 最小值:1 最大值:200 默认值:true
_start	int32	否	起始位置, 从 0 开始 最小值:0 最大值:2000000

topNumber	string	否	显示前10/20/30/60名的流量趋势(默认为TOP60) 默认值:true 枚举值:[TOP10:前10名的会话排行 TOP20:前20名的会话排行 TOP30:前30名的会话排行 TOP60:前60名的会话排行]
topFilter	string	否	指定获取哪种排行(默认为NEW) 默认值:true 枚举值:[NEW:按每秒新建会话排行 TOTAL:按总会话数量排行]

请求样例:

获取会话数量排行

GET https://192.168.1.1/api/v1/namespaces/public/topsessionnumbers

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	分页数据
data.totalItems	int32	总共多少项目
data.totalPages	int32	总共多少页
data.pageNumber	int32	当前页码, 从 1 开始
data.pageSize	int32	每页多大
data.itemsOffset	int32	当前条目偏移, 从 0 开始
data.itemLength	int32	数据列表长度
data.items	object[]	有效数据列表
data.items.ip	string	IPv4或者IPv6地址 格式:ip-address
data.items.sessionNumber	object	会话数量
data.items.sessionNumber.total	uint32	总会话数量
data.items.sessionNumber.ICMP	uint32	ICMP会话数量
data.items.sessionNumber.UDP	uint32	UDP会话数量
data.items.sessionNumber.TCP	uint32	TCP会话数量

data.items.sessionNumber.other	uint32	其他类型会话数量
data.items.sessionNumber.newSession	uint32	每秒新建会话

请求成功样例：

```
{
  "data": {
    "itemsOffset": 0,
    "items": [
      {
        "ip": "20.20.2.120",
        "sessionNumber": {
          "newSession": 37,
          "UDP": 3342,
          "total": 27802,
          "TCP": 24454,
          "other": 0,
          "ICMP": 6
        }
      },
      {
        "ip": "20.20.2.164",
        "sessionNumber": {
          "newSession": 36,
          "UDP": 3383,
          "total": 27750,
          "TCP": 24363,
          "other": 0,
          "ICMP": 4
        }
      },
      {
        "ip": "20.20.2.210",
        "sessionNumber": {
          "newSession": 36,
          "UDP": 3422,
          "total": 27871,
          "TCP": 24445,
          "other": 0,
          "ICMP": 4
        }
      },
      {
        "ip": "20.20.2.170",
        "sessionNumber": {
          "newSession": 36,
          "UDP": 3389,
          "total": 27735,
          "TCP": 24343,
          "other": 0,
          "ICMP": 3
        }
      },
      {
        "ip": "20.20.2.46",
        "sessionNumber": {
          "newSession": 35,
          "UDP": 3422,
          "total": 28239,
          "TCP": 24809,
          "other": 0,
          "ICMP": 8
        }
      },
      {
        "ip": "20.20.2.101",
        "sessionNumber": {
          "newSession": 35,
          "UDP": 3394,
          "total": 27761,
          "TCP": 24362,
          "other": 0,
          "ICMP": 0
        }
      }
    ]
  }
}
```



```
        "ICMP": 5
      }
    },
    {
      "ip": "20.20.2.239",
      "sessionNumber": {
        "newSession": 35,
        "UDP": 3593,
        "total": 28074,
        "TCP": 24475,
        "other": 0,
        "ICMP": 6
      }
    },
    {
      "ip": "20.20.2.31",
      "sessionNumber": {
        "newSession": 34,
        "UDP": 3387,
        "total": 27691,
        "TCP": 24298,
        "other": 0,
        "ICMP": 6
      }
    },
    {
      "ip": "20.20.2.77",
      "sessionNumber": {
        "newSession": 33,
        "UDP": 3390,
        "total": 28074,
        "TCP": 24681,
        "other": 0,
        "ICMP": 3
      }
    },
    {
      "ip": "20.20.2.144",
      "sessionNumber": {
        "newSession": 33,
        "UDP": 3336,
        "total": 27898,
        "TCP": 24556,
        "other": 0,
        "ICMP": 6
      }
    }
  ],
  "totalItems": 10,
  "pageNumber": 1,
  "itemLength": 10,
  "totalPages": 1,
  "pageSize": 100
},
{
  "code": 0,
  "message": "成功"
}
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

7.2.2.2 获取监听列表IP

功能说明: 获取监听列表IP, 最多可监听128个IP

1) API请求数据

```
GET https://{host}/api/v1/namespaces/@namespace/monitorip
```

2) 输入参数

- URL参数

参数	类型	是否必选	描述
_length	int32	否	获取最大长度, 限制最大值, 如果数据不够可能返回小于 length 最小值:1 最大值:200 默认值:true
_start	int32	否	起始位置, 从 0 开始 最小值:0 最大值:2000000

请求样例:

获取监听列表IP

GET https://192.168.1.1/api/v1/namespaces/public/monitorip

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	分页数据
data.totalItems	int32	总共多少项目
data.totalPages	int32	总共多少页
data.pageNumber	int32	当前页码, 从 1 开始
data.pageSize	int32	每页多大
data.itemsOffset	int32	当前条目偏移, 从 0 开始
data.itemLength	int32	数据列表长度
data.items	string[]	有效数据列表

请求成功样例:

```
{
  "data": {
    "itemsOffset": 0,
    "items": [
      "192.168.1.101",
      "192.168.1.102",
      "192.168.1.103",
      "192.168.1.104",
      "192.168.1.105",
      "192.168.1.106",
      "192.168.1.107",
      "192.168.1.108"
    ],
    "totalItems": 8,
    "pageNumber": 1,
    "itemLength": 8,
    "totalPages": 1,
    "pageSize": 100
  },
  "code": 0,
  "message": "成功"
}
```

```
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

7.2.2.3 添加一条监听列表IP范围

功能说明: 添加监听列表IP, 指定一条IP地址范围进行范围内IP的删除

1) API请求数据

POST https://{host}/api/v1/namespaces/@namespace/monitorip

2) 输入参数

- Body参数

参数	类型	是否必选	描述
obj	object	是	监听IP范围
obj.monitorIP	object	是	监听IP范围 格式:ip-range
obj.monitorIP.start	string	是	ip范围的起始位置 格式:ip-address
obj.monitorIP.end	string	否	ip范围的终止位置 格式:ip-address
obj.monitorIP.bits	uint8	否	ip掩码简写
obj.monitorIP.mask	string	否	ip掩码 格式:ip-address

请求样例:

添加一条监听列表IP范围1.1.1.1-1.1.1.2

```
POST https://192.168.1.1/api/v1/namespaces/public/monitorip
{
  "monitorIP": {
    "start": "1.1.1.1",
    "end": "1.1.1.2"
  }
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	监听IP范围
data.monitorIP	object	监听IP范围 格式:ip-range
data.monitorIP.start	string	ip范围的起始位置 格式:ip-address

data.monitorIP.end	string	ip范围的终止位置 格式:ip-address
data.monitorIP.bits	uint8	ip掩码简写
data.monitorIP.mask	string	ip掩码 格式:ip-address

请求成功样例:

```
{
  "data": {
    "monitorIP": {
      "start": "1.1.1.1",
      "end": "1.1.1.2"
    }
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

7.2.2.4 删除监听列表IP范围

功能说明: 删除监听列表IP, 指定一条IP地址范围进行范围内IP的删除

1) API请求数据

POST https://{host}/api/v1/namespaces/@namespace/monitorip?_method=delete

2) 输入参数

• Body参数

参数	类型	是否必选	描述
obj	object	是	监听IP范围
obj.monitorIP	object	是	监听IP范围 格式:ip-range
obj.monitorIP.start	string	是	ip范围的起始位置 格式:ip-address
obj.monitorIP.end	string	否	ip范围的终止位置 格式:ip-address
obj.monitorIP.bits	uint8	否	ip掩码简写
obj.monitorIP.mask	string	否	ip掩码 格式:ip-address

请求样例:

```
删除0.0.0.0-255.255.255.255范围内的所有监听ip
POST https://192.168.1.1/api/v1/namespaces/public/monitorip?_method=DELETE
{
  "monitorIP": {
    "start": "0.0.0.0",
    "end": "255.255.255.255"
  }
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	监听IP范围
data.monitorIP	object	监听IP范围 格式:ip-range
data.monitorIP.start	string	ip范围的起始位置 格式:ip-address
data.monitorIP.end	string	ip范围的终止位置 格式:ip-address
data.monitorIP.bits	uint8	ip掩码简写
data.monitorIP.mask	string	ip掩码 格式:ip-address

请求成功样例:

```
{
  "data": {
    "monitorIP": {
      "start": "0.0.0.0",
      "end": "255.255.255.255"
    }
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

7.2.2.5 全量覆盖监听列表IP

功能说明: 全量覆盖监听列表IP, 指定IP地址范围列表进行全量覆盖

1) API请求数据

PUT

https://{host}/api/v1/namespaces/@namespace/monitorip

2) 输入参数

- Body参数

参数	类型	是否必选	描述
obj	object	是	监听IP范围列表
obj.monitorIP	object[]	是	监听IP范围列表
obj.monitorIP.start	string	是	ip范围的起始位置 格式:ip-address
obj.monitorIP.end	string	否	ip范围的终止位置 格式:ip-address

obj.monitorIP.bits	uint8	否	ip掩码简写
obj.monitorIP.mask	string	否	ip掩码 格式:ip-address

请求样例:

全量覆盖监听列表IP 1.1.1.1~1.1.1.2

```
PUT https://192.168.1.1/api/v1/namespaces/public/monitorip
{
  "monitorIP": [
    {
      "start": "1.1.1.1",
      "end": "1.1.1.2"
    }
  ]
}
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	监听IP范围列表
data.monitorIP	object[]	监听IP范围列表
data.monitorIP.start	string	ip范围的起始位置 格式:ip-address
data.monitorIP.end	string	ip范围的终止位置 格式:ip-address
data.monitorIP.bits	uint8	ip掩码简写
data.monitorIP.mask	string	ip掩码 格式:ip-address

请求成功样例:

```
{
  "data": {
    "monitorIP": [
      {
        "start": "1.1.1.1",
        "end": "1.1.1.2"
      }
    ]
  },
  "code": 0,
  "message": "成功"
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

7.2.2.6 获取每日新建会话信息

功能说明：获取每日新建会话信息

1) API请求数据

GET <https://{host}/api/v1/namespaces/@namespace/sessiondailys>

2) 输入参数

- URL参数

参数	类型	是否必选	描述
_length	int32	否	获取最大长度，限制最大值，如果数据不够可能返回小于 length 最小值:1 最大值:200 默认值:true
ip	string	否	IPv4地址
_start	int32	否	起始位置, 从 0 开始 最小值:0 最大值:2000000

请求样例：

GET https://192.168.1.1/api/v1/namespaces/public/sessiondailys

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	分页数据
data.totalItems	int32	总共多少项目
data.totalPages	int32	总共多少页
data.pageNumber	int32	当前页码，从 1 开始
data.pageSize	int32	每页多大
data.itemsOffset	int32	当前条目偏移, 从 0 开始
data.itemLength	int32	数据列表长度
data.items	object[]	有效数据列表
data.items.date	string	日期
data.items.sessionNumber	object	会话数量
data.items.sessionNumber.total	uint32	总会话数量
data.items.sessionNumber.ICMP	uint32	ICMP会话数量

data.items.sessionNumber.UDP	uint32	UDP会话数量
data.items.sessionNumber.TCP	uint32	TCP会话数量
data.items.sessionNumber.other	uint32	其他类型会话数量

请求成功样例：

```
{
  "message": "",
  "data": {
    "itemsOffset": 0,
    "items": [
      {
        "date": "1949-10-01",
        "sessionNumber": {
          "UDP": 0,
          "other": 0,
          "TCP": 0,
          "total": 0,
          "ICMP": 0
        }
      }
    ],
    "totalItems": 1,
    "pageNumber": 0,
    "pageSize": 200,
    "itemLength": 1,
    "totalPages": 1
  },
  "code": 0
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

7.2.2.7 获取会话详情

功能说明：获取会话详情数据列表，必需指定源IP地址,目的IP地址可选默认为所有,可指定协议和连接状态进行过滤操作

1) API请求数据

GET

https://{host}/api/v1/namespaces/@namespace/sessiondetails

2) 输入参数

- URL 参数

参数	类型	是否必选	描述
_length	int32	否	获取最大长度，限制最大值，如果数据不够可能返回小于 length 最小值:1 最大值:200 默认值:true

connectStatus	string	否	过滤连接状态(默认为ALL) 默认值:true 枚举值:[ALL:所有连接状态 ESTABLISH:已建立连接 CLOSE:已关闭连接 HANDSHAKE:半连接状态]
dstIP	string	否	目的IP (不选默认为全部)
protocol	string	否	过滤协议(默认为ALL) 默认值:true 枚举值:[ALL:所有协议 TCP:TCP协议 UDP:UDP协议 ICMP:ICMP协议 OTHER:其他协议]
srcIP	string	否	源IP
_start	int32	否	起始位置, 从 0 开始 最小值:0 最大值:2000000

请求样例:

获取会话详情, 源ip为20.20.2.126, 获取最大长度为3

GET https://192.168.1.1/api/v1/namespaces/public/sessiondetails?srcIP=20.20.2.126&_length=3

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	分页数据
data.totalItems	int32	总共多少项目
data.totalPages	int32	总共多少页
data.pageNumber	int32	当前页码, 从 1 开始
data.pageSize	int32	每页多大
data.itemsOffset	int32	当前条目偏移, 从 0 开始
data.itemLength	int32	数据列表长度
data.items	object[]	有效数据列表
data.items.source	object	会话源IP详情
data.items.source.srcIP	string	源IP 格式:ip-address
data.items.source.srcNATIP	string	NAT源IP 格式:ip-address

data.items.source.srcPort	int32	源端口 最小值:0 最大值:65535
data.items.source.srcNATPort	int32	NAT源端口 最小值:0 最大值:65535
data.items.source.srcZone	string	源区域
data.items.destination	object	会话目的IP详情
data.items.destination.dstIP	string	目的IP（不选默认为全部） 格式:ip-address
data.items.destination.dstNATIP	string	NAT目的IP 格式:ip-address
data.items.destination.dstPort	int32	目的端口 最小值:0 最大值:65535
data.items.destination.dstNATPort	int32	NAT源端口 最小值:0 最大值:65535
data.items.destination.dstZone	string	目的区域
data.items.protocol	string	协议
data.items.connectStatus	string	连接状态(已建立连接/已关闭连接/半连接状态)
data.items.appName	string	应用名称

请求成功样例:

```
{
  "data": {
    "itemsOffset": 0,
    "items": [
      {
        "source": {
          "srcPort": 53997,
          "srcZone": "Virtual_trust_A",
          "srcNATPort": 53997,
          "srcIP": "20.20.2.126",
          "srcNATIP": "20.20.2.126"
        },
        "destination": {
          "dstNATIP": "20.20.1.202",
          "dstPort": 3582,
          "dstIP": "20.20.1.202",
          "dstNATPort": 3582,
          "dstZone": "Virtual_trust_B"
        },
        "protocol": "TCP",
        "appName": "P2PSrv",
        "connectStatus": "建立"
      }
    ]
  },
  "source": {
    "srcPort": 2214,
    "srcZone": "Virtual_trust_A",
    "srcNATPort": 2214,
    "srcIP": "20.20.2.126",
    "srcNATIP": "20.20.2.126"
  },
  "destination": {
```

```
        "dstNATIP": "20.20.1.72",
        "dstPort": 5820,
        "dstIP": "20.20.1.72",
        "dstNATPort": 5820,
        "dstZone": "Virtual_trust_B"
      },
      "protocol": "TCP",
      "appName": "P2PSrv",
      "connectStatus": "建立"
    },
    {
      "source": {
        "srcPort": 10395,
        "srcZone": "Virtual_trust_B",
        "srcNATPort": 10395,
        "srcIP": "20.20.1.114",
        "srcNATIP": "20.20.1.114"
      },
      "destination": {
        "dstNATIP": "20.20.2.126",
        "dstPort": 2478,
        "dstIP": "20.20.2.126",
        "dstNATPort": 2478,
        "dstZone": "Virtual_trust_A"
      },
      "protocol": "TCP",
      "appName": "P2PSrv",
      "connectStatus": "建立"
    }
  ],
  "totalItems": 21756,
  "pageNumber": 1,
  "itemLength": 3,
  "totalPages": 7252,
  "pageSize": 3
},
{
  "code": 0,
  "message": "成功"
}
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

7.2.2.8 获取会话数量趋势折线图

功能说明：获取会话数量趋势，必须指定要获取趋势的IP地址，可选择会话类型，时间范围和趋势对应的日期进行过滤获取

1) API请求数据

GET

https://{host}/api/v1/namespaces/@namespace/sessionnumbers

2) 输入参数

- URL 参数

参数	类型	是否必选	描述
sessionType	string	否	并发会话数量或新建会话数量(默认NEW) 默认值:true 枚举值:[CONCURRENT:并发会话数量 NEW:每秒新建会话数量]

timeFilter	string	否	最近多长时间段内的总会话数量趋势,该字段不指定ip地址时才生效 枚举值:[REAL-TIME:最近的总会话数量趋势 24HOUR:最近24小时的总会话数量趋势 7DAY:最近7天的总会话数量趋势]
ip	string	否	要过滤的ip地址,若不选该字段,默认全部
dateFilter	string	否	指定获取一周内某天日期(yyyy-mm-dd)的数量趋势,指定week代表一周的总数量趋势 该字段指定了ip地址才生效

请求样例:

获取会话数量趋势折线图,过滤ip为20.20.2.126,时间为周

GET https://192.168.1.1/api/v1/namespaces/public/sessionnumbers?ip=20.20.2.126&dateFilter=week

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	会话数量趋势数据
data.sessionNumbers	object[]	会话数量趋势数据
data.sessionNumbers.sessionNumber	uint32	实时会话数量
data.sessionNumbers.time	string	日期时间

请求成功样例:

```
{
  "data": {
    "sessionNumbers": [
      {
        "sessionNumber": 0,
        "time": "2020-07-09 00:00:00"
      },
      {
        "sessionNumber": 0,
        "time": "2020-07-09 01:00:00"
      },
      {
        "sessionNumber": 0,
        "time": "2020-07-09 02:00:00"
      },
      {
        "sessionNumber": 0,
        "time": "2020-07-09 03:00:00"
      },
      {
        "sessionNumber": 0,
        "time": "2020-07-09 04:00:00"
      },
      {
        "sessionNumber": 0,
        "time": "2020-07-09 05:00:00"
      }
    ]
  }
}
```

```

    {
      "sessionNumber": 0,
      "time": "2020-07-09 06:00:00"
    },
    {
      "sessionNumber": 0,
      "time": "2020-07-09 07:00:00"
    },
    {
      "sessionNumber": 0,
      "time": "2020-07-09 08:00:00"
    },
    {
      "sessionNumber": 0,
      "time": "2020-07-09 09:00:00"
    }
  ],
  "code": 0,
  "message": "成功"
}

```

请求失败样例:

```

{
  "data": "",
  "code": 22,
  "message": "无效参数"
}

```

7.2.2.9 获取指定源IP的会话详情

功能说明: 获取会话排行

1) API请求数据

GET <https://{host}/api/v1/namespaces/@namespace/sessionsrcip>

2) 输入参数

• URL 参数

参数	类型	是否必选	描述
_length	int32	否	获取最大长度, 限制最大值, 如果数据不够可能返回小于 length 最小值:1 最大值:200 默认值:true
srcIP	string	否	源IP 格式:ip-address
_start	int32	否	起始位置, 从 0 开始 最小值:0 最大值:2000000

请求样例:

获取源IP 172.168.0.1的会话详情

GET <https://192.168.1.1/api/v1/namespaces/public/sessionsrcip?srcIP=172.168.0.1>

3) 输出参数

• API返回数据

参数	类型	描述
----	----	----

code	int	错误码
message	string	错误信息
data	object	分页数据
data.totalItems	int32	总共多少项目
data.totalPages	int32	总共多少页
data.pageNumber	int32	当前页码, 从 1 开始
data.pageSize	int32	每页多大
data.itemsOffset	int32	当前条目偏移, 从 0 开始
data.itemLength	int32	数据列表长度
data.items	object[]	有效数据列表
data.items.dstIP	string	目的IP
data.items.dstIPLocation	string	目的IP归属地
data.items.sessionNumber	object	会话数量
data.items.sessionNumber.total	uint32	总会话数量
data.items.sessionNumber.ICMP	uint32	ICMP会话数量
data.items.sessionNumber.UDP	uint32	UDP会话数量
data.items.sessionNumber.TCP	uint32	TCP会话数量
data.items.sessionNumber.other	uint32	其他类型会话数量

请求成功样例:

```
{
  "data": {
    "itemsOffset": 0,
    "items": [
      {
        "dstIP": "172.168.10.61",
        "dstIPLocation": "美国",
        "sessionNumber": {
          "UDP": 0,
          "total": 482,
          "TCP": 482,
          "other": 0,
          "ICMP": 0
        }
      },
      {
        "dstIP": "172.168.10.66",
        "dstIPLocation": "美国",
        "sessionNumber": {
          "UDP": 0,
          "total": 417,
          "TCP": 417,
          "other": 0,
          "ICMP": 0
        }
      },
      {
        "dstIP": "172.168.10.62",
```

```
        "dstIPLocation": "美国",
        "sessionNumber": {
            "UDP": 0,
            "total": 299,
            "TCP": 299,
            "other": 0,
            "ICMP": 0
        }
    },
    {
        "dstIP": "172.168.10.60",
        "dstIPLocation": "美国",
        "sessionNumber": {
            "UDP": 0,
            "total": 251,
            "TCP": 251,
            "other": 0,
            "ICMP": 0
        }
    },
    {
        "dstIP": "172.168.10.63",
        "dstIPLocation": "美国",
        "sessionNumber": {
            "UDP": 0,
            "total": 241,
            "TCP": 241,
            "other": 0,
            "ICMP": 0
        }
    },
    {
        "dstIP": "172.168.10.65",
        "dstIPLocation": "美国",
        "sessionNumber": {
            "UDP": 0,
            "total": 221,
            "TCP": 221,
            "other": 0,
            "ICMP": 0
        }
    },
    {
        "dstIP": "172.168.10.64",
        "dstIPLocation": "美国",
        "sessionNumber": {
            "UDP": 0,
            "total": 197,
            "TCP": 197,
            "other": 0,
            "ICMP": 0
        }
    },
    {
        "dstIP": "所有",
        "dstIPLocation": "-",
        "sessionNumber": {
            "UDP": 0,
            "total": 2109,
            "TCP": 2109,
            "other": 0,
            "ICMP": 0
        }
    }
],
"totalItems": 8,
"pageNumber": 1,
"itemLength": 8,
"totalPages": 1,
"pageSize": 100
},
"code": 0,
"message": "成功"
}
```

请求失败样例:

```
{  
  "data": "",  
  "code": 22,  
  "message": "无效参数"  
}
```


8 扩展API

8.1 引用关系

8.1.1 查询指定认证用户的引用关系

功能说明：查询指定认证用户的引用关系

1) API请求数据

GET

https://{host}/api/v1/namespaces/@namespace/ref/authuser/@name

2) 输入参数

请求样例：

```
GET https://192.168.1.1/api/v1/namespaces/public/ref/authuser/admin
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	分页数据
data.totalItems	int32	总共多少项目
data.totalPages	int32	总共多少页
data.pageNumber	int32	当前页码，从 1 开始
data.pageSize	int32	每页多大
data.itemsOffset	int32	当前条目偏移，从 0 开始
data.itemLength	int32	数据列表长度
data.items	object[]	有效数据列表
data.items.uuid	string	资源唯一id
data.items.name	string	引用者的名称
data.items.moduleId	string	引用模块的id
data.items.moduleName	string	引用模块的名称

请求成功样例：

```
{
  "message": "",
  "data": {
    "itemsOffset": 0,
    "items": [
      {
        "moduleId": "test",
        "uuid": "07150a520899461e956bd9f0a69ea82d",
        "name": "test",
        "moduleName": "test"
      }
    ],
    "totalItems": 1,
  }
}
```

```
      "pageNumber": 0,
      "pageSize": 200,
      "itemLength": 1,
      "totalPages": 1
    },
    "code": 0
  }
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

8.1.2 查询指定认证用户组的引用关系

功能说明: 查询指定认证用户组的引用关系

1) API请求数据

GET

https://{host}/api/v1/namespaces/@namespace/ref/authorg/@name

2) 输入参数

请求样例:

GET https://192.168.1.1/api/v1/namespaces/public/ref/authorg/admin

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	分页数据
data.totalItems	int32	总共多少项目
data.totalPages	int32	总共多少页
data.pageNumber	int32	当前页码, 从 1 开始
data.pageSize	int32	每页多大
data.itemsOffset	int32	当前条目偏移, 从 0 开始
data.itemLength	int32	数据列表长度
data.items	object[]	有效数据列表
data.items.uuid	string	资源唯一id
data.items.name	string	引用者的名称
data.items.moduleId	string	引用模块的id
data.items.moduleName	string	引用模块的名称

请求成功样例:

```
{
  "message": "",
  "data": {
    "itemsOffset": 0,
    "items": [
```

```
{
  {
    "moduleld": "test",
    "uuid": "07150a520899461e956bd9f0a69ea82d",
    "name": "test",
    "moduleName": "test"
  }
},
"totalItems": 1,
"pageNumber": 0,
"pageSize": 200,
"itemLength": 1,
"totalPages": 1
},
"code": 0
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

8.1.3 查询指定ipgroup的引用关系

功能说明: 查询指定ipgroup的引用关系

1) API请求数据

GET <https://{host}/api/v1/namespaces/@namespace/ref/ipgroup/@name>

2) 输入参数

- URL 参数

参数	类型	是否必选	描述
_start	int32	否	起始位置, 从 0 开始 最小值:0 最大值:2000000
_search	string	否	模糊搜索关键字 最大长度:95
_length	int32	否	获取最大长度, 限制最大值, 如果数据不够 可能返回小于 length 最小值:1 最大值:200 默认值:true

请求样例:

GET <https://192.168.1.1/api/v1/namespaces/public/ref/ipgroup/admin>

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	分页数据
data.totalItems	int32	总共多少项目

data.totalPages	int32	总共多少页
data.pageNumber	int32	当前页码, 从 1 开始
data.pageSize	int32	每页多大
data.itemsOffset	int32	当前条目偏移, 从 0 开始
data.itemLength	int32	数据列表长度
data.items	object[]	有效数据列表
data.items.uuid	string	资源唯一id
data.items.name	string	引用者的名称
data.items.moduleId	string	引用模块的id
data.items.moduleName	string	引用模块的名称

请求成功样例:

```
{
  "message": "",
  "data": {
    "itemsOffset": 0,
    "items": [
      {
        "moduleId": "test",
        "uuid": "07150a520899461e956bd9f0a69ea82d",
        "name": "test",
        "moduleName": "test"
      }
    ],
    "totalItems": 1,
    "pageNumber": 0,
    "pageSize": 200,
    "itemLength": 1,
    "totalPages": 1
  },
  "code": 0
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

8.1.4 查询指定service的引用关系

功能说明: 查询指定service的引用关系

1) API请求数据

GET

https://{host}/api/v1/namespaces/@namespace/ref/service/@name

2) 输入参数

- URL参数

参数	类型	是否必选	描述
_start	int32	否	起始位置, 从 0 开始 最小值:0 最大值:2000000

_search	string	否	模糊搜索关键字 最大长度:95
_length	int32	否	获取最大长度，限制最大值，如果数据不够可能返回小于 length 最小值:1 最大值:200 默认值:true

请求样例：

GET https://192.168.1.1/api/v1/namespaces/public/ref/service/admin

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	分页数据
data.totalItems	int32	总共多少项目
data.totalPages	int32	总共多少页
data.pageNumber	int32	当前页码，从 1 开始
data.pageSize	int32	每页多大
data.itemsOffset	int32	当前条目偏移, 从 0 开始
data.itemLength	int32	数据列表长度
data.items	object[]	有效数据列表
data.items.uuid	string	资源唯一id
data.items.name	string	引用者的名称
data.items.moduleId	string	引用模块的id
data.items.moduleName	string	引用模块的名称

请求成功样例：

```
{
  "message": "",
  "data": {
    "itemsOffset": 0,
    "items": [
      {
        "moduleId": "test",
        "uuid": "07150a520899461e956bd9f0a69ea82d",
        "name": "test",
        "moduleName": "test"
      }
    ],
    "totalItems": 1,
    "pageNumber": 0,
    "pageSize": 200,
    "itemLength": 1,
    "totalPages": 1
  },
  "code": 0
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

8.1.5 查询指定serviceGroup的引用关系

功能说明：查询指定serviceGroup的引用关系

1) API请求数据

GET

https://{host}/api/v1/namespaces/@namespace/ref/servicegroup/@name

2) 输入参数

- URL 参数

参数	类型	是否必选	描述
_start	int32	否	起始位置, 从 0 开始 最小值:0 最大值:2000000
_search	string	否	模糊搜索关键字 最大长度:95
_length	int32	否	获取最大长度，限制最大值，如果数据不够可能返回小于 length 最小值:1 最大值:200 默认值:true

请求样例：

```
GET https://192.168.1.1/api/v1/namespaces/public/ref/servicegroup/admin
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	分页数据
data.totalItems	int32	总共多少项目
data.totalPages	int32	总共多少页
data.pageNumber	int32	当前页码，从 1 开始
data.pageSize	int32	每页多大
data.itemsOffset	int32	当前条目偏移, 从 0 开始
data.itemLength	int32	数据列表长度
data.items	object[]	有效数据列表
data.items.uuid	string	资源唯一id
data.items.name	string	引用者的名称

data.items.moduleId	string	引用模块的id
data.items.moduleName	string	引用模块的名称

请求成功样例:

```
{
  "message": "",
  "data": {
    "itemsOffset": 0,
    "items": [
      {
        "moduleId": "test",
        "uuid": "07150a520899461e956bd9f0a69ea82d",
        "name": "test",
        "moduleName": "test"
      }
    ],
    "totalItems": 1,
    "pageNumber": 0,
    "pageSize": 200,
    "itemLength": 1,
    "totalPages": 1
  },
  "code": 0
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

8.1.6 查询指定ip地址库的引用关系

功能说明: 查询指定ip地址库的引用关系

1) API请求数据

GET https://{host}/api/v1/namespaces/@namespace/ref/iparea/@name

2) 输入参数

- URL参数

参数	类型	是否必选	描述
_start	int32	否	起始位置, 从 0 开始 最小值:0 最大值:2000000
_search	string	否	模糊搜索关键字 最大长度:95
_length	int32	否	获取最大长度, 限制最大值, 如果数据不够可能返回小于 length 最小值:1 最大值:200 默认值:true

请求样例:

GET https://192.168.1.1/api/v1/namespaces/public/ref/iparea/admin

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	分页数据
data.totalItems	int32	总共多少项目
data.totalPages	int32	总共多少页
data.pageNumber	int32	当前页码，从 1 开始
data.pageSize	int32	每页多大
data.itemsOffset	int32	当前条目偏移，从 0 开始
data.itemLength	int32	数据列表长度
data.items	object[]	有效数据列表
data.items.uuid	string	资源唯一id
data.items.name	string	引用者的名称
data.items.moduleId	string	引用模块的id
data.items.moduleName	string	引用模块的名称

请求成功样例：

```
{
  "message": "",
  "data": {
    "itemsOffset": 0,
    "items": [
      {
        "moduleId": "test",
        "uuid": "07150a520899461e956bd9f0a69ea82d",
        "name": "test",
        "moduleName": "test"
      }
    ],
    "totalItems": 1,
    "pageNumber": 0,
    "pageSize": 200,
    "itemLength": 1,
    "totalPages": 1
  },
  "code": 0
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

8.1.7 查询指定循环时间计划的引用关系

功能说明：查询指定循环时间计划的引用关系

1) API请求数据

GET `https://{host}/api/v1/namespaces/@namespace/ref/schedule/@name`

2) 输入参数

• URL 参数

参数	类型	是否必选	描述
_start	int32	否	起始位置, 从 0 开始 最小值:0 最大值:2000000
_search	string	否	模糊搜索关键字 最大长度:95
_length	int32	否	获取最大长度, 限制最大值, 如果数据不够可能返回小于 length 最小值:1 最大值:200 默认值:true

请求样例:

GET `https://192.168.1.1/api/v1/namespaces/public/ref/schedule/admin`

3) 输出参数

• API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	分页数据
data.totalItems	int32	总共多少项目
data.totalPages	int32	总共多少页
data.pageNumber	int32	当前页码, 从 1 开始
data.pageSize	int32	每页多大
data.itemsOffset	int32	当前条目偏移, 从 0 开始
data.itemLength	int32	数据列表长度
data.items	object[]	有效数据列表
data.items.uuid	string	资源唯一id
data.items.name	string	引用者的名称
data.items.moduleId	string	引用模块的id
data.items.moduleName	string	引用模块的名称

请求成功样例:

```
{
  "message": "",
  "data": {
    "itemsOffset": 0,
    "items": [
```

```
{
  {
    "moduleld": "test",
    "uuid": "07150a520899461e956bd9f0a69ea82d",
    "name": "test",
    "moduleName": "test"
  },
  "totalItems": 1,
  "pageNumber": 0,
  "pageSize": 200,
  "itemLength": 1,
  "totalPages": 1
},
"code": 0
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

8.1.8 查询指定单次时间计划的引用关系

功能说明: 查询指定单次时间计划的引用关系

1) API请求数据

GET <https://{host}/api/v1/namespaces/@namespace/ref/singleschedule/@name>

2) 输入参数

- URL 参数

参数	类型	是否必选	描述
_start	int32	否	起始位置, 从 0 开始 最小值:0 最大值:2000000
_search	string	否	模糊搜索关键字 最大长度:95
_length	int32	否	获取最大长度, 限制最大值, 如果数据不够 可能返回小于 length 最小值:1 最大值:200 默认值:true

请求样例:

GET <https://192.168.1.1/api/v1/namespaces/public/ref/singleschedule/admin>

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	分页数据
data.totalItems	int32	总共多少项目

data.totalPages	int32	总共多少页
data.pageNumber	int32	当前页码, 从 1 开始
data.pageSize	int32	每页多大
data.itemsOffset	int32	当前条目偏移, 从 0 开始
data.itemLength	int32	数据列表长度
data.items	object[]	有效数据列表
data.items.uuid	string	资源唯一id
data.items.name	string	引用者的名称
data.items.moduleId	string	引用模块的id
data.items.moduleName	string	引用模块的名称

请求成功样例:

```
{
  "message": "",
  "data": {
    "itemsOffset": 0,
    "items": [
      {
        "moduleId": "test",
        "uuid": "07150a520899461e956bd9f0a69ea82d",
        "name": "test",
        "moduleName": "test"
      }
    ],
    "totalItems": 1,
    "pageNumber": 0,
    "pageSize": 200,
    "itemLength": 1,
    "totalPages": 1
  },
  "code": 0
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

8.1.9 查找接口被双机和sslvpn的引用情况

功能说明: 查询指定接口的vpn和双机引用情况

1) API请求数据

GET

https://{host}/api/v1/namespaces/@namespace/ifvpnhaRef/@name

2) 输入参数

- URL参数

参数	类型	是否必选	描述
_search	string	否	模糊搜索关键字 最大长度:95

请求样例:

GET https://192.168.1.1/api/v1/namespaces/public/ivfvpnharef/admin

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	分页数据
data.totalItems	int32	总共多少项目
data.totalPages	int32	总共多少页
data.pageNumber	int32	当前页码, 从 1 开始
data.pageSize	int32	每页多大
data.itemsOffset	int32	当前条目偏移, 从 0 开始
data.itemLength	int32	数据列表长度
data.items	object[]	有效数据列表
data.items.uuid	string	资源唯一id
data.items.name	string	引用者的名称
data.items.moduleId	string	引用模块的id
data.items.moduleName	string	引用模块的名称

请求成功样例:

```
{
  "message": "",
  "data": {
    "itemsOffset": 0,
    "items": [
      {
        "moduleId": "test",
        "uuid": "07150a520899461e956bd9f0a69ea82d",
        "name": "test",
        "moduleName": "test"
      }
    ],
    "totalItems": 1,
    "pageNumber": 0,
    "pageSize": 200,
    "itemLength": 1,
    "totalPages": 1
  },
  "code": 0
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

8.1.10 查询指定区域的引用关系

功能说明：查询指定区域的引用关系

1) API请求数据

GET `https://{host}/api/v1/namespaces/@namespace/ref/zone/@name`

2) 输入参数

- URL 参数

参数	类型	是否必选	描述
_start	int32	否	起始位置, 从 0 开始 最小值:0 最大值:2000000
_search	string	否	模糊搜索关键字 最大长度:95
_length	int32	否	获取最大长度, 限制最大值, 如果数据不够可能返回小于 length 最小值:1 最大值:200 默认值:true

请求样例:

GET `https://192.168.1.1/api/v1/namespaces/public/ref/zone/admin`

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	分页数据
data.totalItems	int32	总共多少项目
data.totalPages	int32	总共多少页
data.pageNumber	int32	当前页码, 从 1 开始
data.pageSize	int32	每页多大
data.itemsOffset	int32	当前条目偏移, 从 0 开始
data.itemLength	int32	数据列表长度
data.items	object[]	有效数据列表
data.items.uuid	string	资源唯一id
data.items.name	string	引用者的名称
data.items.moduleId	string	引用模块的id
data.items.moduleName	string	引用模块的名称

请求成功样例:

```
{
  "message": "",
  "data": {
    "itemsOffset": 0,
    "items": [
      {
        "moduleId": "test",
        "uuid": "07150a520899461e956bd9f0a69ea82d",
        "name": "test",
        "moduleName": "test"
      }
    ],
    "totalItems": 1,
    "pageNumber": 0,
    "pageSize": 200,
    "itemLength": 1,
    "totalPages": 1
  },
  "code": 0
}
```

请求失败样例：

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

8.1.11 查询Net指定应用的引用关系

功能说明：查询Net指定应用的引用关系

1) API请求数据

GET

https://{host}/api/v1/namespaces/@namespace/ref/appidentify/@appPath

2) 输入参数

- URL 参数

参数	类型	是否必选	描述
_start	int32	否	起始位置, 从 0 开始 最小值:0 最大值:2000000
_search	string	否	模糊搜索关键字 最大长度:95
_length	int32	否	获取最大长度, 限制最大值, 如果数据不够 可能返回小于 length 最小值:1 最大值:200 默认值:true

请求样例：

GET https://192.168.1.1/api/v1/namespaces/public/ref/appidentify/test

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息

data	object	分页数据
data.totalItems	int32	总共多少项目
data.totalPages	int32	总共多少页
data.pageNumber	int32	当前页码, 从 1 开始
data.pageSize	int32	每页多大
data.itemsOffset	int32	当前条目偏移, 从 0 开始
data.itemLength	int32	数据列表长度
data.items	object[]	有效数据列表
data.items.uuid	string	资源唯一id
data.items.name	string	引用者的名称
data.items.moduleId	string	引用模块的id
data.items.moduleName	string	引用模块的名称

请求成功样例:

```
{
  "message": "",
  "data": {
    "itemsOffset": 0,
    "items": [
      {
        "moduleId": "test",
        "uuid": "07150a520899461e956bd9f0a69ea82d",
        "name": "test",
        "moduleName": "test"
      }
    ],
    "totalItems": 1,
    "pageNumber": 0,
    "pageSize": 200,
    "itemLength": 1,
    "totalPages": 1
  },
  "code": 0
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

8.1.12 查询指定应用的引用关系

功能说明: 查询指定应用的引用关系

1) API请求数据

GET

https://{host}/api/v1/namespaces/@namespace/ref/appidentifyrule/@name

2) 输入参数

- URL参数

参数	类型	是否必选	描述
----	----	------	----

_start	int32	否	起始位置, 从 0 开始 最小值:0 最大值:2000000
_search	string	否	模糊搜索关键字 最大长度:95
_length	int32	否	获取最大长度, 限制最大值, 如果数据不够 可能返回小于 length 最小值:1 最大值:200 默认值:true

请求样例:

GET https://192.168.1.1/api/v1/namespaces/public/ref/appidentifyrule/admin

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	分页数据
data.totalItems	int32	总共多少项目
data.totalPages	int32	总共多少页
data.pageNumber	int32	当前页码, 从 1 开始
data.pageSize	int32	每页多大
data.itemsOffset	int32	当前条目偏移, 从 0 开始
data.itemLength	int32	数据列表长度
data.items	object[]	有效数据列表
data.items.uuid	string	资源唯一id
data.items.name	string	引用者的名称
data.items.moduleId	string	引用模块的id
data.items.moduleName	string	引用模块的名称

请求成功样例:

```
{
  "message": "",
  "data": {
    "itemsOffset": 0,
    "items": [
      {
        "moduleId": "test",
        "uuid": "07150a520899461e956bd9f0a69ea82d",
        "name": "test",
        "moduleName": "test"
      }
    ],
    "totalItems": 1,
    "pageNumber": 0,
    "pageSize": 200,
    "itemLength": 1,
    "totalPages": 1
  }
}
```



```
    },  
    "code": 0  
  }  
}
```

请求失败样例:

```
{  
  "data": "",  
  "code": 22,  
  "message": "无效参数"  
}
```

8.1.13 查询应规则引用状态

功能说明: 查询指定应用列表的引用状态

1) API请求数据

POST

https://{host}/api/v1/namespaces/@namespace/ref/appidentifyrules?_method=get

2) 输入参数

- Body参数

参数	类型	是否必选	描述
objs	object[]	是	用户启禁用应用规则列表
objs.apppath	string	是	规则名称 最小长度:1 最大长度:287
objs.hasref	bool	否	启禁用

请求样例:

```
POST https://192.168.1.1/api/v1/namespaces/public/ref/appidentifyrules?_method=get  
[  
  {  
    "apppath": "test",  
    "hasref": true  
  }  
]
```

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object[]	用户启禁用应用规则列表
data.apppath	string	规则名称 最小长度:1 最大长度:287
data.hasref	bool	启禁用

请求成功样例:

```
{  
  "message": "",  
  "data": [  
    {  
      "apppath": "test",  
      "hasref": true  
    }  
  ]  
}
```

```
{
  "apppath": "test",
  "hasref": true
},
"code": 0
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```

8.2 配置完整性校验

8.2.1 配置完整性校验

功能说明: 配置完整性校验

1) API请求数据

GET <https://{host}/api/v1/namespaces/@namespace/integcheck>

2) 输入参数

- URL参数

参数	类型	是否必选	描述
resource	string	否	查询api对应的URL 最小长度:1 最大长度:4096
uuid	string	否	资源全局唯一ID 最大长度:32
name	string	否	资源名 最大长度:1440

请求样例:

GET <https://192.168.1.1/api/v1/namespaces/public/integcheck>

3) 输出参数

- API返回数据

参数	类型	描述
code	int	错误码
message	string	错误信息
data	object	返回配置完整性校验结果
data.name	string	配置名称 最小长度:1 最大长度:95
data.uuid	string	配置uuid
data.data	bool	配置是否完整

data.msg	string	配置错误的具体提示消息 最小长度:1 最大长度:255
----------	--------	-----------------------------------

请求成功样例:

```
{
  "message": "",
  "data": {
    "uuid": "07150a520899461e956bd9f0a69ea82d",
    "msg": "错误提示信息",
    "data": true,
    "name": "admin"
  },
  "code": 0
}
```

请求失败样例:

```
{
  "data": "",
  "code": 22,
  "message": "无效参数"
}
```