

ispapp / collect-v2 Private

<> Code

Issues 34

Pull requests

Actions

Projects

Security



master ▾

[collect-v2](#) / [docs](#) / SERVER_API_DOC.md

andrewhodel put host type field not required

History

1 contributor

UPDATE TYPES for <https://domain.tld/api/v1/endpoint>

- GET returns info on an object
- POST creates an object
- PUT updates an object
- DELETE deletes an object

AUTHORIZATION

If AUTH REQUIRED is specified for a particular endpoint, make sure you send the parameters `email` and `password` with your request.

GET /auth - test auth

AUTH REQUIRED

REQUEST PARAMS

email*
password*

RESPONSE CODES

200 - Valid
returns {success:1}
400 - Unauthorized
returns {error:err}

GET /backup - get a server backup

AUTH REQUIRED

REQUEST PARAMS

RESPONSE CODES

200 - Valid
returns json document containing all collections that are required for a backup
400 - Unauthorized
returns nothing

POST /backup - restore from a backup json document

AUTH REQUIRED

REQUEST PARAMS

backup - JSON the document returned from GET /backup

RESPONSE CODES

200 - Valid
returns {success: 1}

400 – Unauthorized
returns nothing

GET /all_host_keys - get all host keys

AUTH REQUIRED

REQUEST PARAMS

RESPONSE CODES

200 – Valid
returns json document with all_host_keys
400 – Unauthorized
returns nothing

GET /adminLog - get adminLog

AUTH REQUIRED

REQUEST PARAMS

`countOnly` `bool` if true, this will
only return the count for the last 24 hours

RESPONSE CODES

200 – Valid
returns json document with adminLog
400 – Unauthorized
returns nothing

GET /admins - get all admins

AUTH REQUIRED

REQUEST PARAMS

countOnly bool if true, this will
only return the count for the last 24 hours

RESPONSE CODES

200 - Valid
returns json document with all admins
400 - Unauthorized
returns nothing

GET /admin - get an admin

AUTH REQUIRED

REQUEST PARAMS

adminEmail*

RESPONSE CODES

200 - Valid
returns json document admin
400 - Unauthorized
returns nothing

POST /admin - create an admin

AUTH REQUIRED

REQUEST PARAMS

☰ 817 lines (648 sloc) | 14.4 KB

...

adminEmail* - STR email of the admin
adminPassword* - STR password of the admin
adminAuthLevel* - INT admin auth level
0 - read only
1 - modify, create
2 - shell access

RESPONSE CODES

200 - Valid
returns json document admin

400 - Unauthorized
returns nothing

PUT /admin - update an admin

AUTH REQUIRED

REQUEST PARAMS

adminEmail - STR email of the admin

adminPassword - STR password of the admin

adminAuthLevel - INT auth level of the admin

notificationsDisabled - INT 0 for enabled, 1 for disabled



RESPONSE CODES

200 - Valid
returns json document admin

400 - Unauthorized
returns nothing

DELETE /admin - delete an admin

AUTH REQUIRED

REQUEST PARAMS

adminEmail* - STR email of the admin

RESPONSE CODES

200 - Valid
returns json document admin

400 - Unauthorized
returns nothing

GET /search - search everything

AUTH REQUIRED

REQUEST PARAMS

s - the search string
type - type of search: host, group or sub
skip - number of results to skip for pagination

RESPONSE CODES

200 - Valid
returns json document with search results
400 - Unauthorized
returns nothing

GET /zones - get all zones

AUTH REQUIRED

REQUEST PARAMS

RESPONSE CODES

200 - Valid
returns json document with all zones
400 - Unauthorized
returns nothing

GET /zone - get zone

AUTH REQUIRED

REQUEST PARAMS

zoneId*

RESPONSE CODES

200 - Valid
returns json document zone
400 - Unauthorized
returns nothing

POST /zone - create a zone

AUTH REQUIRED

REQUEST PARAMS

name* - STR name of the zone
notes* - STR notes for the zone

RESPONSE CODES

200 - Valid
returns json document zone
400 - Unauthorized
returns nothing

PUT /zone - update a zone

AUTH REQUIRED

REQUEST PARAMS

zoneId* - STR id of the zone
name - STR name of the zone
notes - STR notes for the zone

RESPONSE CODES

200 - Valid
returns json document admin

400 - Unauthorized
returns nothing

DELETE /zone - delete a zone

AUTH REQUIRED

REQUEST PARAMS

zoneId*

RESPONSE CODES

200 - Valid
returns json document zone

400 - Unauthorized
returns nothing

GET /groups - get all groups for zoneId

AUTH REQUIRED

REQUEST PARAMS

zoneId* - STR id of parent zone

RESPONSE CODES

200 - Valid
returns json document groups

400 - Unauthorized
returns nothing

GET /group - get group data

AUTH REQUIRED

REQUEST PARAMS

groupId

RESPONSE CODES

200 - Valid
returns json document zone

400 - Unauthorized
returns nothing

POST /group - create a group

AUTH REQUIRED

REQUEST PARAMS

name* - STR name for group
notes* - STR notes for group
zoneId* - STR zone _id of parent zone

RESPONSE CODES

200 - Valid
returns json document group

400 - Unauthorized
returns nothing

PUT /group - update a group

AUTH REQUIRED

REQUEST PARAMS

groupId* - STR id of the group
name - STR name of the group
notes - STR notes for the group

RESPONSE CODES

200 - Valid
returns json document group

400 - Unauthorized
returns nothing

DELETE /group - delete a group

AUTH REQUIRED

REQUEST PARAMS

groupId*

RESPONSE CODES

200 - Valid
returns json document group

400 - Unauthorized
returns nothing

GET /subs - get all subscribers by any field

AUTH REQUIRED

REQUEST PARAMS

subEmail - STR email address of subscriber

firstName - STR first name of sub

lastName - STR last name of sub

telephone - STR telephone number of sub

bAddress - STR billing address of sub

bCity - STR billing city

bState - STR billing state

bZip - STR billing zip

bCountry - STR billing country

sAddress - STR shipping address

sCity - STR shipping city

sState - STR shipping state

sZip – STR shipping zip
sCountry – STR shipping country
hostId – STR id of associated host
groupId – STR id of associated group
zoneId – STR id of associated zone

RESPONSE CODES

200 – Valid
returns json document subs
400 – Unauthorized
returns nothing

GET /sub - get subscriber data

AUTH REQUIRED

REQUEST PARAMS

subId

RESPONSE CODES

200 – Valid
returns json document sub
400 – Unauthorized
returns nothing

POST /sub - create a subscriber

AUTH REQUIRED

REQUEST PARAMS

subEmail – STR email address of subscriber
firstName – STR first name of sub
lastName – STR last name of sub
telephone – STR telephone number of sub
bAddress – STR billing address of sub

bCity - STR billing city
bState - STR billing state
bZip - STR billing zip
bCountry - STR billing country
sAddress - STR shipping address
sCity - STR shipping city
sState - STR shipping state
sZip - STR shipping zip
sCountry - STR shipping country
hostId - ARRAY:STR id of associated host
groupId - ARRAY:STR id of associated group
zoneId - ARRAY:STR id of associated zone

RESPONSE CODES

200 - Valid
returns json document sub
400 - Unauthorized
returns nothing

PUT /sub - update a subscriber

AUTH REQUIRED

REQUEST PARAMS

subEmail - STR email address of subscriber
firstName - STR first name of sub
lastName - STR last name of sub
telephone - STR telephone number of sub
bAddress - STR billing address of sub
bCity - STR billing city
bState - STR billing state
bZip - STR billing zip
bCountry - STR billing country
sAddress - STR shipping address
sCity - STR shipping city
sState - STR shipping state
sZip - STR shipping zip
sCountry - STR shipping country
hostId - ARRAY:STR id of associated host
groupId - ARRAY:STR id of associated group
zoneId - ARRAY:STR id of associated zone

RESPONSE CODES

200 - Valid
returns json document sub
400 - Unauthorized
returns nothing

DELETE /sub - delete a subscriber

AUTH REQUIRED

REQUEST PARAMS

subId*

RESPONSE CODES

200 - Valid
returns {success: 1}
400 - Unauthorized
returns nothing

GET /unknownHosts - get all unknown hosts, those that have made a BOOT request to the listener but have not been added to collect

AUTH REQUIRED

REQUEST PARAMS

countOnly bool if true, this will
only return the count

RESPONSE CODES

200 - Valid
returns json document hosts

400 - Unauthorized
returns nothing

GET /hosts - get all hosts for groupId

AUTH REQUIRED

REQUEST PARAMS

groupId*

RESPONSE CODES

200 - Valid
returns json document hosts
400 - Unauthorized
returns nothing

GET /hostsForZone - get all hosts for zoneId

AUTH REQUIRED

REQUEST PARAMS

zoneId*

RESPONSE CODES

200 - Valid
returns json document hosts
400 - Unauthorized
returns nothing

GET /host - get host data

AUTH REQUIRED

REQUEST PARAMS

hostId*

RESPONSE CODES

200 - Valid
 returns json document zone

400 - Unauthorized
 returns nothing

POST /host - create a host

AUTH REQUIRED

REQUEST POST PARAMS

type*	STR	normal or snmp
login authentication key authentication	STR	host login for host password for
snmpIp for snmp access	STR	ip address of host
snmpPort snmp access	STR	port of host for
snmpRelayId to use	STR	id of snmp relay
name*	STR	host display name
latitude	FLOAT	
longitude	FLOAT	
notes* host	STR	notes about the
groupId*	STR	parent group id
wirelessMode (must be one of: ap_bridge, ap_router, sta)	STR	wireless mode ap_bridge ap_router
Create a layer 2 bridge between the WAN and the wireless		
Act as a NAT router between the WAN and the wireless, serve DHCP over wireless		

		sta	
Connect to the first SSID specified in wirelessConfigs[]			
wirelessChannel	INT	wifi channel for	
host (specify 0 for auto)			
wirelessBeaconInt	INT	beacon interval in	
milliseconds			
wirelessConfigs	ARRAY	list of wireless	
configurations			
wirelessConfigs.ssid	STR	ssid (could be for	
a vap)			
wirelessConfigs.encType	STR	encryption type	
(must be one of: sae, sae-mixed, psk2, psk, owe, none)			
WPA3 Personal (SAE)		sae	
WPA2/WPA3 Personal (PSK/SAE) mixed mode		sae-mixed	
WPA2 Personal (PSK)		psk2	
WPA Personal (PSK)		psk	
Opportunistic Wireless Encryption (OWE)		owe	
Layer 2 Encryption		none	No
wirelessConfigs.encKey	STR	encryption key	
wirelessConfigs.vlanId	INT	vlan ID which is	
between 1 and 4094			
wirelessConfigs.clientIsolation	BOOL	enable client	
isolation for this ssid			
wirelessConfigs.hiddenSsid	BOOL	hide the ssid	
wirelessConfigs.doth	BOOL	enable 802.11h	
(DFS, cooperate with radar)			
wirelessConfigs.dotw	BOOL	enable 802.11w	
(improved security of management frames)			
wirelessConfigs.sp	BOOL	enable short	
preamble			
wirelessConfigs.dtimPeriod	INT	set the DTIM	
Period			

RESPONSE CODES

- 200 – Valid
returns json document host
- 400 – Unauthorized
returns nothing

PUT /host - update a host

AUTH REQUIRED

REQUEST PARAMS

type	STR	normal or snmp
login authentication key authentication	STR	host login for
	STR	host password for
snmpIp for snmp access	STR	ip address of host
snmpPort snmp access	STR	port of host for
snmpRelayId to use	STR	id of snmp relay
name*	STR	host display name
latitude	FLOAT	
longitude	FLOAT	
notes* host	STR	notes about the
groupId*	STR	parent group id
reboot on next update	INT	1 to reboot host
alertDisabled disable alerts for this host, 0 to enable	INT	Number of hours to
wirelessMode (must be one of: ap_bridge, ap_router, sta)	STR	wireless mode
		ap_bridge
Create a layer 2 bridge between the WAN and the wireless		ap_router
Act as a NAT router between the WAN and the wireless, serve DHCP over wireless		sta
Connect to the first SSID specified in wirelessConfigs[]		
wirelessChannel host (specify 0 for auto)	INT	wifi channel for
wirelessBeaconInt milliseconds	INT	beacon interval in
wirelessConfigs configurations	ARRAY	list of wireless
wirelessConfigs.ssid	STR	ssid (could be for a vap)

wirelessConfigs.encType	STR	encryption type
(must be one of: sae, sae-mixed, psk2, psk, owe, none)		sae
WPA3 Personal (SAE)		sae-mixed
WPA2/WPA3 Personal (PSK/SAE) mixed mode		psk2
WPA2 Personal (PSK)		psk
WPA Personal (PSK)		owe
Opportunistic Wireless Encryption (OWE)		none
		No
Layer 2 Encryption		
wirelessConfigs.encKey	STR	encryption key
wirelessConfigs.vlanId	INT	vlan ID which is
between 1 and 4094		
wirelessConfigs.clientIsolation	BOOL	enable client
isolation for this ssid		
wirelessConfigs.hiddenSsid	BOOL	hide the ssid
wirelessConfigs.doth	BOOL	enable 802.11h
(DFS, cooperate with radar)		
wirelessConfigs.dotw	BOOL	enable 802.11w
(improved security of management frames)		
wirelessConfigs.sp	BOOL	enable short
preamble		
wirelessConfigs.dtimPeriod	INT	set the DTIM
Period		

RESPONSE CODES

- 200 - Valid
returns json document admin
- 400 - Unauthorized
returns nothing

DELETE /host - delete a host

AUTH REQUIRED

REQUEST PARAMS

hostId*

RESPONSE CODES

200 - Valid
returns json document zone

400 - Unauthorized
returns nothing

GET /globalCollectors - list all active collectors in the system

AUTH REQUIRED

REQUEST PARAMS

RESPONSE CODES

200 - Valid
returns json collectors

400 - Unauthorized
returns nothing

GET /collectors - get collectors with overview for a host

AUTH REQUIRED

REQUEST PARAMS

hostId* - STR id of the host

RESPONSE CODES

200 - Valid
returns json document collectors

400 - Unauthorized
returns nothing

GET /collector - get collector detail for a host

AUTH REQUIRED

REQUEST PARAMS

hostId* - STR id of the host
collector* - STR name of the collector
hash* - STR id of the data to return

RESPONSE CODES

200 - Valid
returns json document collector
400 - Unauthorized
returns nothing

GET /firmware - get list of firmware files

AUTH REQUIRED 2

REQUEST PARAMS

RESPONSE CODES

200 - Valid
returns json document
400 - Unauthorized
returns nothing

POST /firmware - request that the firmware generator generates and sends firmware files to the server

AUTH REQUIRED 2

REQUEST PARAMS

hostKey* - STR host key to include in the firmware

RESPONSE CODES

200 - Valid
returns json with status: 'success'
400 - Unauthorized
returns nothing

GET /firmware_temp - get a temporary hash that lasts 24 hours to download a firmware file from anywhere

AUTH REQUIRED 2

REQUEST PARAMS

filepath* - STR full path of the file, as returned from GET /firmware

RESPONSE CODES

200 - Valid
returns json document with tempHash: hash for download at
/firmware_temp?tempHash=hash
400 - Unauthorized
returns nothing