

# Sandblast Mobile - Dashboard External API

## Authentication

The general authentication schema uses basic authentication over SSL.

Each API call requires an API authentication key that is supplied in the request header. The format is  
Authorization: ApiKey <username>:<api\_key>

For example:

Authorization: ApiKey api\_user:204db7bcfafb2deb7506b89eb3b9b715b0990

Generate API key: see - create\_api\_key.pdf

## Throttling

In order to avoid a system overload due to a bug in the calling system API throttling will be used. The API key will have an X (depending on policy) amount of API calls per second. In case the amount exceeded then the server will respond with status code 429.

## Devices

Get device status for device ID/MDM UUID

Device details such as manufacturer, model, OS and status can be retrieved in multiple ways using an API. Search requests can be made by Check Point's internal device ID or MDM UUID in case the device was enrolled via MDM. The details returned depend on what the device reports; different devices may return different information.

## Examples

Get the device details for the device that has a specified device ID:

[https://mydashboard.locsec.net/external\\_api/v1/device\\_status/50](https://mydashboard.locsec.net/external_api/v1/device_status/50)

Get the device details for the device that has a specified MDM UUID:

[https://mydashboard.locsec.net/external\\_api/v1/device\\_status/6f778977-70fc-4c35-a003-1b3faf2437ad](https://mydashboard.locsec.net/external_api/v1/device_status/6f778977-70fc-4c35-a003-1b3faf2437ad)

## **REST API**

<b>1. URI:</b> <code>https://{host-name}/external_api/v1/device_status/\${device_id}/?format=json</code>	<b>Get device details by Check Point's device ID.</b>
HTTP Method:	GET
Format:	Json
<code>\${device_id}</code>	Check Point's internal identification of the device.
<b>Response</b>	
client_version	
device_type	
email	User email
internal_id	Internal Check Point's ID
last_connection	Last time the device contacted the system.
mail_sent	True if registration mail sent, else, False
mdm: { uuid }	In case the device was provisioned by a MDM then it will contain the unique identifier as reported by the MDM.
model	Device model
name	User name
number	Phone Number
os_type	Operation system name. Can be either "Android" or "IOS"
os_version	Version of the operation system
risk	The current risk of the device.
status	The status of the device, is it connected to the system, User Notified registration, etc.

<b>2. URI: https://{host-name}/external_api/v1/device_status/?mdm__uuid=\${mdmuuid}&amp;format=json</b>	<b>Get device details by the mdm uuid.</b>
HTTP Method:	GET
Format:	Json
`\${mdmuuid}`	The mdm unique identifier. The format changes depending on the MDM. If the device wasn't enrolled via MDM then you'll get back an empty result
<b>Response</b>	
"meta": { "limit": 20, "next": null, "offset": 0, "previous": null, "total_count": 1 }	Each response will contain a header indicating how many pages the request contains and how many results are found in the current response.
"objects":	The results are part of an array.
client_version	
device_type	
email	User email
internal_id	Internal Check Point's ID
last_connection	Last time the device contacted the system.
mail_sent	True if registration mail sent, else, False
mdm: { uuid }	In case the device was provisioned by a MDM then it will contain the unique identifier as reported by the MDM.
model	Device model
name	User name
number	Phone Number
os_type	Operation system name. Can be either "Android" or "IOS"
os_version	Version of the operation system
risk	The current risk of the device.
status	The status of the device, is it connected to the system, registration, etc.

<b>3. URI: https://{host-name}/external_api/v1/device_status/?format=json&amp;risk__in=&lt;&gt;&amp;status__in=&lt;&gt;</b>	<b>Get all device details. You can filter by status and risk level. The list supplied in risk and status should be ‘,’ separated.</b>
HTTP Method:	GET
Format:	Json
<b>Response</b>	
{ "meta": { "limit": 20, "next": null, "offset": 0, "previous": null, "total_count": 1 }	Each response will contain a header indicating how many pages the request contains and how many results are found in the current response.
"objects":	The results are part of an array.
status	The status of the device, is it connected to the system, registration, etc.
risk	The current risk of the device.
last_connection	Last time the device contacted the system.
os_type	Operation system name. Can be either “Android” or “IOS”
os_version	Version of the operation system
model	Device model
internal_id	Internal Check Point's ID
mdm: { uuid }	In case the device was provisioned by a MDM then it will contain the unique identifier as reported by the MDM.

<b>4. URI: https://{host-name}/external_api/v1/device_status?status=\${status}&amp;time_gte=\${time_from}&amp;time_lte=\${time_to}&amp;format=json</b>	<b>Get all devices that had a specific status during time interval</b>
HTTP Method:	GET
Format:	Json
\${status}	A status that a device has. the value is taken from a predefined list of statuses.
\${time_from}	Epoch time in milliseconds. Inclusive.
\${time_to}	Epoch time in milliseconds. Exclusive.
<b>Response</b>	
{ "meta": { "limit": 20, "next": null, "offset": 0, "previous": null, "total_count": 1 }	Each response will contain a header indicating how many pages the request contains and how many results are found in the current response.
"objects":	The results are part of an array.
status	The status of the device, is it connected to the system, registration, etc.
risk	The current risk of the device.
last_connection	Last time the device contacted the system.
os_type	Operation system name. Can be either "Android" or "IOS"
os_version	Version of the operation system
model	Device model
internal_id	Internal Check Point's ID
mdm: { uuid }	In case the device was provisioned by a MDM then it will contain the unique identifier as reported by the MDM.

<b>5. URI:</b> https://{host-name}/external_api/v1/device_status/\${device_id}/resend	<b>Resend email notification to a specific device id. An email will be sent only to devices that are in User Notified status. see Device Status</b>
HTTP Method:	POST
\${device_id}	Check Point's internal identification of the device.
Request body:	
{ "send_reg_sms": True/False, "send_reg_email": True/False }	Optional
<b>Response</b>	
HTTP 200	Email was sent successfully
HTTP 500	Error while trying to send an email

<b>6. URI:</b> https://{host-name}/external_api/v1/device_status/\${device_id}/mark_email_sent/	<b>Change the device status to "User Notified".</b>
HTTP Method	Post
\${device_id}	Check Point's internal identification of the device.
<b>Response</b>	
HTTP 200	Marked email sent
HTTP 404	Device not found

<b>7. URI:</b> https://{host-name}/external_api/v1/device_status/\${device_id}/get_registration_details	<b>Enable getting device registration code for existing device by device id</b>
HTTP Method:	GET
\${device_id}	Check Point's internal identification of the device
<b>Response</b>	
device_hash	Device registration code
id	Check Point's internal identification of the device
server	Check Point's server

<b>8. URI:</b> <code>https://{host-name}/external_api/v1/device_status/get_registration_details?mi_id=\${MI_id}&amp;device_id=\${device_id}&amp;email=\${email}</code>	<b>Enable getting device registration code for existing device (by device id, email and/or MI id)</b>
HTTP Method:	GET
<code>\${MI_id}</code>	MobileIron Device UUID
<code>\${device_id}</code>	Check Point's internal identification of the device
<code>\${email}</code>	Device owner email
<b>Response</b>	
device_hash	Device registration code
id	Check Point's internal identification of the device
server	Check Point's server

<b>9. URI:</b> <code>https://{host-name}/external_api/v1/device/?user_email=\${email}</code>	<b>Delete device</b>
HTTP Method:	DELETE
<code>\${email}</code>	Device owner email

<b>10. URI:</b> <code>https://{host-name}/external_api/v1/device/\${device_id}/</code>	<b>Delete device</b>
HTTP Method:	DELETE
<code>\${device_id}</code>	Internal device id

<b>11. URI:</b> <code>https://{host-name}/external_api/v2/device/</code>	<b>Add device</b>
HTTP Method:	POST
POST body should contains the following json:	
<code>{"name": \${user_name}</code>	Required
<code>"email": \${user_email}</code>	Required
<code>"number": \${user_number}</code>	Optional.
<code>"send_reg_email":false&gt;true</code>	Optional.
<code>"send_reg_sms":false&gt;true</code>	Optional.
<code>"groups":["ALL\"," \"]}</code>	Optional.
<b>Response</b>	
device_id	Check Point's internal identification of the device
status	The status of the request (success/failure)

<b>12. URI: https://{host-name}/external_api/v2/alert/?id__gt=&lt;alert_id&gt;</b>	<b>Get dashboards alerts</b>
HTTP Method:	GET
<alert_id>	Optional.
<b>Response</b>	
id	Check Point's internal identification of the alert
number	The device phone number
user_action	
mdm_uuid	The device MDM unique identifier
alertType	
risk_level	
backend_last_updated	
email	
alertDetails	
updated	
description	
timestamp	The time stamp for the alert update in the backend
user_action_timestamp	
model	
device_id	
risk_level_desc	
mtp_client_version	
alertEvent	
name	
created	
event_timestamp	The time stamp for the alert update in the device
resolutionState	
os_type	
device_rooted	
For Network Attack:	
location	
captive_network	
network_certificate	
For Suspicious Package (Application):	
app_version	
app_repackage	
threat_summary	
id__gt (optional field)	Return all the alerts with ID grater than the one provided

## **Appendix**

### **Device Status**

State	Enum	Description
Processing	-1	Device was added to the dashboard, but not to the GW
Provisioned/User Notified	0	Device was added to the system, however it did not finish the registration flow
Active	1	Device registered successfully and is under Check Point's protection
In Active	4	User removed Check Point's app from the device

### **Device Risk**

None	0	Device has no risk associated with it
Low	1	Device has low risk associated with it
Medium	2	Device has medium risk associated with it
High	3	Device has high risk associated with it