Black•Vault HSM

REST API

9565 Soquel Drive
Aptos, CA 95003
Phone +1 831.688.1021
1 877.ENGAGE4 (364.2434)
sales@engageblack.com

Version 1.0.3 October 13, 2021

Disclaimer and Warranty

Engage Black is a business unit of Engage Communication.

Engage Communication makes no representations or warranties with respect to the contents of this document and specifically disclaims any implied warranties of merchantability of fitness for any particular purpose. Information in this document is subject to change without notice and does not represent a commitment on the part of Engage Communication, Inc. Product specifications are subject to change without notice. Engage Communication assumes no responsibility for any inaccuracies in this document or for any obligation to update the information in this document.

Engage Communications, Inc.

9565 Soquel Drive Aptos, CA 95003

Phone +1 (831) 688-1021

https://www.engageblack.com/

https://www.engageinc.com/

Table of Contents

1. Introduction and Caution

⚠CAUTION: If upgrading a BlackVault HSM to the RS model (Firmware version 7.1.1 and above),
it is critical to review how Backup and Restore should be performed properly during the transition.
Restoring from a backup made prior to upgrading to the RS model will **not** be compatible with RS model firmware.
Please see more details in the "BlackVault HSM upgrade to RS Model (Multi User and REST API) Guide"

This document will describe the use and functionality of the BlackVault HSM Representational State Transfer (REST) Application Programming Interface (API). The REST API allows a BlackVault HSM operator to request a resource or service through a URL path, these paths contain endpoints that perform an action. Requests and responses are `application/json` content type and follow the common HTTP response status codes for success and failure.

As per RESTful design, the BlackVault HSM implements the following HTTP actions:

- `GET` – read resource
- `PUT` – configure existing resource
- `POST` – create new resource

The REST API endpoint URL for the BlackVault HSM is as follows:

```
https://192.168.1.7:9999
```

Note: example is shown for the default BlackVault HSM IP address, 192.168.1.7

2. Prerequisites

There are a few different ways to utilize the BlackVault HSM REST API. The two following sections explain the two most common methods.

2.1. Curl

The command line tool **curl** is a tool for transferring data with URLs. Please see the appendix for examples of how to use curl with the BlackVault HSM REST API.

To download curl, visit this website, select Download, and then the correct version for your operating system.

2.2. libcurl

libcurl is a library of functions and the engine in the **curl** command. libcurl is for applications written in C and C++ that need to perform Internet data transfers, such as a BlackVault HSM REST API integrator. Although libcurl is restricted to C and C++ programmers, there are many bindings and interfaces available for various environments and programming languages. Please see the appendix for examples of how to use libcurl with the BlackVault HSM REST API.

3. Client Token

A client token is required for all HTTP requests. This token should be in universally unique identifier (UUID) format. The token is client generated and established with the first HTTP request. It is used by the BlackVault HSM to identify individual clients. The exit endpoint is used to break the connection with a specific client and token. The conn endpoint will return the IP address of the current client.

4. Endpoints

This section will contain information about the different REST endpoints available for use with the BlackVault HSM. Each endpoint has an authorization requirement specified in their section.

The endpoint URLs shown in each section are for the default BlackVault HSM IP address, 192.168.1.7.

4.1. Information

The **information** endpoint performs a `GET` action and will return information about your BlackVault HSM including the bootloader version, firmware version, boot partition, if FIPS mode is enabled, serial number, MAC address, and model. This endpoint does not need to authorize a BlackVault HSM operator, the device can be logged in or out.

Endpoint URL:

```
https://192.168.1.7:9999/info
```

Example Response:

```
{
    "bootloader": "8.2.10",
    "fw": "07.01.01",
    "partition": "user",
    "fips_mode": "on",
    "serial": "700000",
    "mac": "00:C0:F7:2A:00:00",
    "model": "BlackVault HSM",
    "slots": 1
}
```

4.2. List Keys

The **list keys** endpoint performs a `GET` action and, if a user operator is logged in, will return the keys associated with the user on the current slot. This endpoint needs to authorize a BlackVault HSM user operator to successfully return the keys for that user.

Endpoint URL:

```
https://192.168.1.7:9999/list_keys
```

Example Response:

```
{
    "keys": [
        "RSA:rsa2048"
    ]
}
```

4.3. User Card Information

The **user card information** endpoint performs a `GET` action and, if a crypto officer is logged in, will return the serial numbers of all user smart cards. This endpoint needs to authorize a BlackVault HSM crypto officer.

Endpoint URL:

```
https://192.168.1.7:9999/user_card_info
```

Example Response:

```
{
    "cards": [
        "XXXXXXXXXXXXXXXX",
        "XXXXXXXXXXXXXXXX"
    ]
}
```

All user card serial numbers will be returned, no X's.

4.4. Crypto Officer Card Information

The **crypto officer card information** endpoint performs a `GET` action and, if a crypto officer is logged in, will return the serial number(s) of the crypto officer cards. The number of serial numbers returned depends on the crypto officer M of N quorum set up during initialization. This endpoint needs to authorize a BlackVault HSM crypto officer to successfully return the crypto officer card information.

Endpoint URL:

```
https://192.168.1.7:9999/co_card_info
```

Example Response:

```
{
    "cards": [
        "XXXXXXXXXXXXXXXX"
    ]
}
```

All crypto officer card serial numbers will be returned, no X's.

4.5. USB Certificate Information

The **USB certificate information** endpoint performs a `GET` action and will return information about the certificates on a USB flash drive currently inserted in the BlackVault HSM. Please note that the certificate must be on the root directory of the flash drive. This endpoint needs to authorize a BlackVault HSM crypto officer.

Endpoint URL:

```
https://192.168.1.7:9999/usb_cert_info
```

Example Response:

```
{
    "filenames": [
        "BVCert.pem"
    ]
}
```

4.6. Status

The **status** endpoint performs a `GET` action and returns the status of your BlackVault HSM. This endpoint is especially useful to check which operator is logged in, if a smartcard or USB is present, and to gain information about the current command. This endpoint does not need to authorize a BlackVault HSM operator, the device can be logged in or out.

Endpoint URL:

```
https://192.168.1.7:9999/status
```

Example Response:

```
{
    "operational_state": "operational",
    "user": "co",
    "slot": 0,
    "usb_present": false,
    "card_present": true,
    "current_time": "Tue Oct 12 2021 20:09 ADT",
    "current_command": "STATE_IDLE",
    "current_command_ticks": 148,
    "state": "STATE_IDLE"
}
```

4.7. Slot List

The **slot list** endpoint performs a `GET` action and returns which user slots are in use. This endpoint does not need to authorize a BlackVault HSM operator, the device can be logged in or out.

Endpoint URL:

```
https://192.168.1.7:9999/slot_list
```

Example Response:

```
{
    "slots": [
        1
    ]
}
```

4.8. Date

The **date** endpoint can perform a `GET` or `PUT` action. As a `GET`, this endpoint returns the date on your BlackVault HSM. As a `PUT`, this endpoint can configure the date. To configure the date, you must specify the month, day, and year as shown in the example body below. This endpoint needs to authorize a BlackVault HSM crypto officer.

Endpoint URL:

```
https://192.168.1.7:9999/config_date
```

Example Response:

```
{
    "day": 5,
    "month": 8,
    "year": 2021
}
```

Example Body:

```
"{ \"day\":1, \"month\":1, \"year\":2000 }"
```

4.9. Time

The **time** endpoint can perform a `GET` or `PUT` action. As a `GET`, this endpoint returns the time (hours and minutes) on your BlackVault HSM. As a `PUT`, this endpoint can configure the current time on your BlackVault HSM. To configure the time, you must specify the hours and minutes, as shown in the example body below. This endpoint needs to authorize a BlackVault HSM crypto officer.

Endpoint URL:

```
https://192.168.1.7:9999/config_time
```

Example Response:

```
{
    "hours": 12,
    "minutes": 00
}
```

Example Body:

```
"{ \"hours\":2, \"minutes\":53 }"
```

4.10. Time Zone

The **time zone** endpoint can perform a `GET` or `PUT` action. As a `GET`, this endpoint returns the time zone currently set on your BlackVault HSM. As a `PUT`, this endpoint can configure the time zone on your BlackVault HSM. To configure the time zone, you must specify the zone, as shown in the example body below. Please view our table of time zone codes to find your area. This endpoint needs to authorize a BlackVault HSM crypto officer.

Endpoint URL:

```
https://192.168.1.7:9999/config_tz
```

Example Response:

```
{
    "timezone": "AST4ADT,M4.1.0/00:01:00,M10.5.0/00:1:00"
}
```

Example Body:

```
"{ \"timezone\":\"PST8PDT\" }"
```

4.10.1. Time Zone Codes

Australia	
Region	**Time Zone Code**
Melbourne, Canberra, Sydney	CEST-10EDT-11,M10.5.0/02:00:00,M3.5.0/03:00:00
Perth	WST-8
Brisbane	EST-10
Adelaide	CST-9:30CDT-10:30,M10.5.0/02:00:00,M3.5.0/03:00:00
Darwin	CST-9:30
Hobart	EST-10EDT-11,M10.1.0/02:00:00,M3.5.0/03:00:00

Europe	
Region	**Time Zone Code**
Amsterdam, Netherlands	CET-1CEST-2,M3.5.0/02:00:00,M10.5.0/03:00:00
Athens, Greece	EET-2EEST-3,M3.5.0/03:00:00,M10.5.0/04:00:00
Barcelona, Spain	CET-1CEST-2,M3.5.0/02:00:00,M10.5.0/03:00:00
Berlin, Germany	CET-1CEST-2,M3.5.0/02:00:00,M10.5.0/03:00:00
Brussels, Belgium	CET-1CEST-2,M3.5.0/02:00:00,M10.5.0/03:00:00
Budapest, Hungary	CET-1CEST-2,M3.5.0/02:00:00,M10.5.0/03:00:00
Copenhagen, Denmark	CET-1CEST-2,M3.5.0/02:00:00,M10.5.0/03:00:00
Dublin, Ireland	GMT+0IST-1,M3.5.0/01:00:00,M10.5.0/02:00:00

Geneva, Switzerland	CET-1CEST-2,M3.5.0/02:00:00,M10.5.0/03:00:00
Helsinki, Finland	EET-2EEST-3,M3.5.0/03:00:00,M10.5.0/04:00:00
Kyiv Ukraine	EET-2EEST,M3.5.0/3,M10.5.0/4
Lisbon, Portugal	WET-0WEST-1,M3.5.0/01:00:00,M10.5.0/02:00:00
London, Great Britain	vGMT+0BST-1,M3.5.0/01:00:00,M10.5.0/02:00:00
Madrid, Spain	CET-1CEST-2,M3.5.0/02:00:00,M10.5.0/03:00:00
Oslo, Norway	CET-1CEST-2,M3.5.0/02:00:00,M10.5.0/03:00:00
Paris, France	CET-1CEST-2,M3.5.0/02:00:00,M10.5.0/03:00:00
Prague, Czech Republic	CET-1CEST-2,M3.5.0/02:00:00,M10.5.0/03:00:00
Roma, Italy	CET-1CEST-2,M3.5.0/02:00:00,M10.5.0/03:00:00
Moscow, Russia	MSK-3MSD,M3.5.0/2,M10.5.0/3
St. Petersburg, Russia	MST-3MDT,M3.5.0/2,M10.5.0/3
Stockholm, Sweden	CET-1CEST-2,M3.5.0/02:00:00,M10.5.0/03:00:00
Tallinn, Estonia	EET-2EEST-3,M3.5.0/03:00:00,M10.5.0/04:00:00

New Zealand	
Region	**Time Zone Code**
Auckland, Wellington	NZST-12NZDT-13,M10.1.0/02:00:00,M3.3.0/03:00:00

United States and Canada	
Region	**Time Zone Code**
Hawaii Time	HAW10

Alaska Time	AKST9AKDT
Pacific Time	PST8PDT
Mountain Time	MST7MDT
Mountain Time (Arizona, no DST)	MST7
Central Time	CST6CDT
Eastern Time	EST5EDT
Atlantic Time	AST4ADT
Atlantic Time (New Brunswick)	AST4ADT,M4.1.0/00:01:00,M10.5.0/00:01:00
Newfoundland Time	NST+3:30NDT+2:30,M4.1.0/00:01:00,M10.5.0/00:01:00

Asia	
Region	**Time Zone Code**
Jakarta	WIB-7
Singapore	SGT-8
Ulaanbaatar	ULAT-8ULAST,M3.5.0/2,M9.5.0/2

Central and South America	
Region	**Time Zone Code**
Brazil, São Paulo	BRST+3BRDT+2,M10.3.0,M2.3.0
Argentina	UTC+3
Central America	CST+6

4.11. NTP Pool

The **NTP pool** endpoint can perform a `GET` or `PUT` action. As a `GET`, this endpoint returns the NTP pool currently set up on your BlackVault HSM. As a `PUT`, this endpoint can configure the NTP pool on your BlackVault HSM. To configure the NTP pool, you must specify the address(es) in the body, as shown in the example below. This endpoint needs to authorize a BlackVault HSM crypto officer.

Endpoint URL:

```
https://192.168.1.7:9999/config_ntp
```

Example Response:

```
{
    "addresses": [
        "us.pool.ntp.org"
    ]
}
```

Example Body:

```
"{ \"addresses\": [ \"us.pool.ntp.org\" ] }"
```

4.12. Firewall

The **firewall** endpoint can perform a `GET` or `PUT` action. As a `GET`, this endpoint returns the firewall configuration currently on your BlackVault HSM. As a `PUT`, this endpoint can configure the firewall on your BlackVault HSM. To configure the firewall, you must enable the whitelist and specify the addresses in the body, as shown in the example below. This endpoint needs to authorize a BlackVault HSM crypto officer.

Endpoint URL:

```
https://192.168.1.7:9999/config_firewall
```

Example Response:

```
{
    "enabled": true,
    "addresses": [
        "192.168.1.3"
    ]
}
```

Example Body:

```
"{ \"enabled\": false, \"addresses\": [ \"192.168.1.3\" ] }"
```

4.13. Network

The **network** endpoint can perform a `GET` or `PUT` action. As a `GET`, this endpoint returns the network configuration currently on your BlackVault HSM. As a `PUT`, this endpoint can configure the network on your BlackVault HSM. To configure the network, you must specify the address, mask, and gateway in the body, as shown in the example below. This endpoint needs to authorize a BlackVault HSM crypto officer.

Endpoint URL:

```
https://192.168.1.7:9999/config_network
```

Example Response:

```
{
    "address": "192.168.1.7",
    "mask": "255.255.255.0",
    "gateway": "192.168.1.1"
}
```

Example Body:

```
"{ \"address\":\"192.168.1.7\", \"mask\":\"255.255.255.0\",
\"gateway\":\"192.168.1.1\" }"
```

4.14. DNS

The **DNS** endpoint can perform a `GET` or `PUT` action. As a `GET`, this endpoint returns the DNS configuration currently on your BlackVault HSM. As a `PUT`, this endpoint can configure the DNS on your BlackVault HSM. To configure the DNS, you must specify the desired addresses in the body, as shown in the example below. This endpoint needs to authorize a BlackVault HSM crypto officer.

Endpoint URL:

```
https://192.168.1.7:9999/config_dns
```

Example Response:

```
{
    "addresses": [
        "8.8.8.8"
    ]
}
```

Example Body:

```
"{ "addresses\": [ \"8.8.8.8\", \"0.0.0.0\" ] }"
```

This endpoint needs to authorize a BlackVault HSM crypto officer to successfully configure the DNS.

4.15. TLS Port

The **TLS port** endpoint can perform a `GET` or `PUT` action. As a `GET`, this endpoint returns the current TLS port of your BlackVault HSM. As a `PUT`, this endpoint can configure a new TLS port on your BlackVault HSM. To configure the TLS port, you must specify the new port in the body, as shown in the example below. This endpoint needs to authorize a BlackVault HSM crypto officer.

Endpoint URL:

```
https://192.168.1.7:9999/config_tlsport
```

Example Response:

```
{
    "port": 5002
}
```

Example Body:

```
"{ \"port\":5002 }"
```

4.16. TLS Client Authentication

The **TLS client authentication** endpoint can perform a `GET` or `PUT` action. As a `GET`, this endpoint returns the TLS client authentication configuration currently on your BlackVault HSM. As a `PUT`, this endpoint can enable or disable TLS client authentication on your BlackVault HSM. This endpoint needs to authorize a BlackVault HSM crypto officer.

Endpoint URL:

```
https://192.168.1.7:9999/config_tlsclientauth
```

Example Response:

```
{
    "enabled": true,
    "certificate_ids": [
        "XXXXXXXX",
        "XXXXXXXX"
    ]
}
```

- `"certificate_ids"` is a list of all certificates on the BlackVault HSM for TLD client authentication

Example Body:

```
"{ \"enabled\":true }"
```

4.17. Certificate Information

The **certificate information** endpoint performs either a `GET` or a `PUT` action. As a `GET`, this endpoint returns information about a given certificate on a BlackVault HSM. As a `PUT`, this endpoint can update or change permissions of a given certificate on a BlackVault HSM. This endpoint needs to authorize a BlackVault HSM crypto officer.

Endpoint URL:

```
https://192.168.1.7:9999/config_certinfo?id=XXXXXXXX
```

The id parameter should contain the ID of the certificate you need information about. To get all certificate IDs on a BlackVault HSM, use the TLS Client Authentication endpoint.

Example Response:

```
{
    "certificate_info": "serial=XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX\n",
    "dn": "CN=CA Certificate,OU=Black,O=Engage,L=Aptos,ST=CA,C=US",
    "slots": [
        1
    ]
}
```

- `"certificate_info"` is the serial number of the certificate with the ID in the parameter
- `"dn"` is the distinguished name of the certificate with the ID in the parameter
- `"slots"` lists the slots that can use the certificate, if it is empty, no slots can use it.

Example Body:

```
"{ \"is_root\":true, \"del\":false, \"slots\": [ 1 ] }"
```

- `"del"` can delete the certificate specified in the parameter; true is delete, false is keep
- `"slots"` can determine which slots can use the certificate specified in the parameter

4.18. Auto Logout

The **auto logout** endpoint can perform a `GET` or `PUT` action. As a `GET`, this endpoint returns the auto logout configuration currently on your BlackVault HSM. As a `PUT`, this endpoint can configure auto logout on your BlackVault HSM. To configure auto logout, you must enable it and specify the desired timeout (**in seconds**) in the body, as shown in the example below. To disable auto logout, set the timeout to 0. This endpoint needs to authorize a BlackVault HSM crypto officer.

Endpoint URL:

```
https://192.168.1.7:9999/config_autologout
```

Example Response:

```
{
    "tls_disconnect": true,
    "timeout": 120
}
```

Example Body:

```
"{ \"tls_disconnect\":false, \"timeout\":53 }'
```

4.19. P11 API

The **PKCS#11 API** endpoint can perform a `GET` or `PUT` action. As a `GET`, this endpoint returns the PKCS#11 API configuration currently on your BlackVault HSM. As a `PUT`, this endpoint can enable or disable PKCS#11 API features on your BlackVault HSM. To configure the API, you must specify true or false for block destroy, log operations, and global login in the body, as shown in the example below. This endpoint needs to authorize a BlackVault HSM crypto officer.

For more information about these features, please consult the BlackVault HSM User Guide.

Endpoint URL:

```
https://192.168.1.7:9999/config_p11api
```

Example Response:

```
{
    "block_destroy": false,
    "log_operations": true,
    "global_login": false
}
```

Example Body:

```
"{ \"block_destroy\":false, \"log_operations\":true, \"global_login\":false
}"
```

4.20. Remote Logging

The **remote logging** endpoint can perform a `GET` or `PUT` action. As a `GET`, this endpoint returns the remote logging configuration currently on your BlackVault HSM. As a `PUT`, this endpoint can

configure a remote logging server for your BlackVault HSM. To configure a remote log, you must specify the configuration features in the body, as shown in the example below. This endpoint needs to authorize a BlackVault HSM crypto officer.

Endpoint URL:

```
https://192.168.1.7:9999/config_remotelog
```

Example Response:

```
{
    "enabled": true,
    "address": "192.168.1.4",
    "use_tls":  true,
    "port": 6514
}
```

Example Body:

```
"{ \"enabled\":false, \"address\":\"0.0.0.0\", \"use_tls\":true,
\"port\":6514 }"
```

4.21. Boot Version Selection

The **boot select** endpoint can perform a `GET` or `PUT` action. As a `GET`, this endpoint returns the boot image currently set on your BlackVault HSM. As a `PUT`, this endpoint can change the boot image on your BlackVault HSM. To change the boot partition, you must specify the desired image (user or factory) in the body, as shown in the example below. This endpoint needs to authorize a BlackVault HSM crypto officer.

Please note that the factory image is the firmware that your BlackVault HSM was shipped with. If you have upgraded to a new firmware, that is the user image.

Endpoint URL:

```
https://192.168.1.7:9999/config_bootsel
```

Example Response:

```
{
    "image": "user"
}
```

Example Body:

```
"{ \"image\":\"user\" }"
```

4.22. Upload

The **upload** endpoint is used in conjunction with a smartcard database import or user database restore, please view those sections for more information. This endpoint does not need to authorize a specific BlackVault HSM operator, the authorization depends on which database backup you are uploading.

When making this HTTP request, the encrypted backup.bv should be in the directory in which the request is made.

HTTP Request:

```
curl https://192.168.1.7:9999/upload --upload-file backup.bv -H 'Expect:'
```

4.23. Download

The **download** endpoint is used in conjunction with a smartcard database export or user database backup, please view those sections for more information. This endpoint does not need to authorize a specific BlackVault HSM operator, the authorization depends on which database backup you are uploading.

HTTP Request:

```
curl https://192.168.1.7:9999/download --output backup.bv
```

After making this HTTP request, the encrypted backup.bv file will be in the directory in which the request was made.

4.24. Conn

The **connection (conn)** endpoint returns the IP address of the connected BlackVault HSM client. To close the connection, use the exit endpoint. This endpoint does not need to authorize a BlackVault HSM operator, the device can be logged out.

Endpoint URL:

```
https://192.168.1.7:9999/conn
```

Example Response:

```
{
    "ip": "127.0.0.1"
}
```

4.25. Exit

The **exit** endpoint breaks the connection of the current client. Once this connection is broken, a new client can connect to the BlackVault HSM (through bvgui, the touchscreen interface, or a new REST connection). The connection will break after 60 seconds without the exit endpoint. To check the IP address of the current client, use the conn endpoint. This endpoint does not need to authorize a BlackVault HSM operator, the device can be logged out.

Endpoint URL:

```
https://192.168.1.7:9999/exit
```

4.26. Crypto Officer Login

The **crypto officer login** endpoint will make a login call to the BlackVault HSM. A smartcard and the crypto officer PIN are both required for this endpoint. To login successfully, complete the following steps:

1. Insert the crypto officer smartcard to the BlackVault HSM.
2. Make the crypto officer login `POST` HTTP Request.
3. Make the PIN HTTP request with the crypto officer pin and context `"STATE_LOGIN_CO"`
4. Make the status HTTP request to verify the crypto officer is logged in.

Endpoint URL:

`https://192.168.1.7:9999/command_login_co`

4.27. User Login

The **user login** endpoint will make a login call to the BlackVault HSM. User smartcard(s) and PIN are both required for this endpoint. To login successfully, complete the following steps (more smartcards are required for a larger M of N quorum):

5. Insert the user smartcard to the BlackVault HSM.
6. Make the user login `POST` HTTP Request with the desired slot in the body.
7. Make the PIN `POST` HTTP Request with the user pin and context `"STATE_LOGIN_USER"`.
8. Make the status HTTP Request to verify the user is logged in

Endpoint URL:

`https://192.168.1.7:9999/command_login_user`

Example Body:

```
"{ \"slot\":1 }"
```

4.28. PIN

The **PIN** endpoint is used in conjunction with any command that requires a PIN (for example, login, create user, initialization, etc..). When using this endpoint, you must specify the context. To verify if a command requires a PIN, after issuing the command, check the status.

This endpoint does not directly need to authorize a BlackVault HSM operator, it depends on the context of which it is needed.

Contexts that require a PIN include:

```
"STATE_RESET_PIN"
"STATE_WAIT_PIN"
"STATE_REPLACE_CARD"
"STATE_LOGIN_CO"
"STATE_LOGIN_USER"
"STATE_INITIALIZE"
"STATE_CREATE_USER"
```

Endpoint URL:

```
https://192.168.1.7:9999/pin
```

Example Body:

```
"{ \"pin\":\"XXXXXXX\", \"context\":\"STATE_WAIT_PIN\" }'
```

4.29. Logout

If an operator is currently logged in, the **logout** endpoint will log that operator out of the BlackVault HSM. If no one is logged in, this endpoint will do nothing.

Endpoint URL:

```
https://192.168.1.7:9999/command_logout
```

4.30. Reboot

The **reboot** endpoint performs a `POST` action and will reboot the BlackVault HSM. This endpoint does not need to authorize a BlackVault HSM operator, the device can be logged in or out.

Endpoint URL:

```
https://192.168.1.7:9999/command_reboot
```

4.31. Upgrade

The **upgrade** endpoint will initialize an upgrade on a BlackVault HSM. This endpoint needs to authorize a BlackVault HSM crypto officer. To successfully upgrade, you will need to use the command line tool **bvupgrade.** This application is automatically installed when you run the BlackVault HSM installer fit for your operating system (see the BlackVault HSM User Guide for more information). To successfully perform an upgrade, please complete the following steps:

1. Login to the BlackVault HSM as crypto officer
2. Make the upgrade `POST` HTTP request
3. Open a terminal or command prompt and issue the following command:
    a. Bvupgrade 192.168.1.7 bvhsm_7.1.1.upg

Note: example is shown with a BlackVault HSM with IP address 192.168.1.7 and firmware version 7.1.1

4. Set the BlackVault HSM boot version to user
5. Reboot the BlackVault HSM

Endpoint URL:

```
https://192.168.1.7:9999/command_upgrade
```

4.32. User Database Backup

The **user database backup** endpoint, when done correctly, will back up all keys and objects on a BlackVault HSM. This endpoint needs to authorize a BlackVault HSM crypto officer. This backup requires at least two additional smartcards and the use of the download endpoint. To do a user database backup, complete the following steps:

Note: this example is for a backup smartcard quorum of 2 of 2. With a larger quorum, you will have to insert and set up additional smartcards

1. Login to the BlackVault HSM as user.
2. Insert one of the user database backup smartcards (order does not matter)
3. Make the user database backup `POST` HTTP request
4. Check the BlackVault HSM status, if it displays `"state":"STATE_WAIT_CONFIRM_CARD"`, make the proceed HTTP request
5. Wait until the BlackVault HSM status displays `"state":"STATE_REMOVE_CARD"`
6. Remove the first card and insert the second.
7. Repeat steps 4 and 5
8. When status displays `"state":"STATE_WAIT_MEDIA",` in a terminal, make the download HTTP request:
   ```
   curl https://192.168.1.7:9999/download --output user_backup.bv
   ```
9. Verify that the encrypted user_backup.bv file is in the directory in which the download HTTP request was made

For more information on how to restore a user database on a BlackVault HSM, refer to the User Database Backup section.

Endpoint URL:

```
https://192.168.1.7:9999/command_db_backup
```

Example Body:

```
"{ \"m\":2, \"n\":2, \"authenticated\":false }"
```

4.33. User Database Restore

The **user database restore** endpoint, when done correctly, will restore all keys and objects that were saved during a backup. To do the restore, you will need the encrypted user_backup.bv file and the smartcard shares that were created during the backup. Complete the following steps to do a user database restore:

1. Login to the BlackVault HSM as user.
2. Insert one of the user database backup smartcards (order does not matter)

3. Make the user database restore `POST` HTTP request (shown below)
4. Wait until the BlackVault HSM status displays `"state":"STATE_REMOVE_CARD"`
5. Remove the first card and insert the second.
6. Repeat steps 4 and 5
7. After the last `"state":"STATE_REMOVE_CARD"` is displayed, in a terminal, make the upload HTTP request (in the directory with the user_backup.bv file):

```
curl https://192.168.1.7:9999/upload --upload-file backup.bv -H
     'Expect:'
```

Endpoint URL:

```
https://192.168.1.7:9999/command_db_restore
```

4.34. Smartcard Database Export

The **smartcard database export** endpoint, when used correctly, will create a backup of all crypto officers and users. We recommend always keeping an up-to-date smartcard database export, in case of emergency. This endpoint needs to authorize a BlackVault HSM crypto officer.

You will need at least two extra smartcards to do this properly. To do a smartcard database export, please complete the following steps:

Note: this example is for a backup smartcard quorum of 2 of 2. With a larger quorum, you will have to insert and set up additional smartcards

1. Login to the BlackVault HSM as crypto officer.
2. Insert the one of the smartcard database export smartcards (order does not matter)
3. Make the smartcard database export `POST` HTTP request
4. Check the BlackVault HSM status, if it displays `"state":"STATE_WAIT_CONFIRM_CARD"`, make the proceed HTTP request
5. Wait until the BlackVault HSM status displays `"state":"STATE_REMOVE_CARD"`
6. Remove the first card and insert the second.
7. Repeat steps 4 and 5
8. When status displays `"state":"STATE_WAIT_MEDIA"`, in a terminal, make the download HTTP request:

   ```
   curl https://192.168.1.7:9999/download --output sc_backup.bv
   ```

9. Verify that the encrypted sc_backup.bv file is in the directory in which the download HTTP request was made

Please view the initialization section for information on how to import a smartcard database.

Endpoint URL:

```
https://192.168.1.7:9999/command_card_export
```

Example Body:

```
"{ \"m\":2, \"n\":2 }"
```

4.35. Reset PIN

The **reset PIN** endpoint can reset the PIN for a BlackVault HSM crypto officer or user. To reset the PIN for a given smartcard, complete the following steps:

1. Login to the BlackVault HSM as crypto officer.
2. Insert the smartcard you wish to reset the PIN for.
3. Make the reset PIN `POST` HTTP request (shown below)
4. Make the PIN `POST` HTTP request with context `"STATE_RESET_PIN"`
5. Verify new PIN works with given smartcard.

Endpoint URL:

```
https://192.168.1.7:9999/command_reset_pin
```

4.36. Replace Smartcard

The **replace smartcard** endpoint can replace a lost or dysfunctional smartcard for either a BlackVault crypto officer or user. To replace a smartcard, you must have a replacement prior to making the replace smartcard `POST` HTTP request. To replace a smartcard, complete the following steps:

1. Login to the BlackVault HSM as crypto officer.
2. Obtain the serial number for the card to be replaced.
3. Insert the replacement card into the BlackVault HSM.
4. Make the replace smartcard `POST` HTTP request.
5. Check the BlackVault HSM status, if the status displays `"state":"STATE_WAIT_CONFIRM_CARD"`, make the proceed HTTP request.
6. Check the BlackVault HSM status again, when the status displays `"state": "STATE_WAIT_PIN"`, make the PIN HTTP request with context `"STATE_REPLACE_CARD"`
7. Once complete, verify new smartcard works

Endpoint URL:

```
https://192.168.1.7:9999/command_replace_card
```

Example Body:

```
"{ \"card\":\"XXXXXXXXXXXXXXXX\" }"
```

4.37. Import Client Certificate

The **import client certificate** endpoint imports a client certificate from the root directory of a USB flash drive. This endpoint needs to authorize a BlackVault HSM crypto officer. To import the client certificate, complete the following steps:

1. Login to the BlackVault HSM as crypto officer.
2. Insert the USB flash drive into the BlackVault HSM.
3. Make the import client certificate `POST` HTTP request with the correct certificate file name.

Endpoint URL:

```
https://192.168.1.7:9999/command_import_client_cert
```

Example Body:

```
‟{ \"filename\":\"BVCert.pem\" }'
```

4.38. Import Server Certificate

The **import server certificate** endpoint imports a server certificate from the root directory of a USB flash drive. This endpoint needs to authorize a BlackVault HSM crypto officer. To import the sever certificate, complete the following steps:

4. Login to the BlackVault HSM as crypto officer.
5. Insert the USB flash drive into the BlackVault HSM.
6. Make the import server certificate `POST` HTTP request with the correct certificate file name.

Endpoint URL:

```
https://192.168.1.7:9999/command_import_server_cert
```

Example Body:

```
‟{ \"filename\": \"BVCert.pem\" }”
```

4.39. Export Server Certificate

The **export server certificate** endpoint will export the server certificate that is currently on the BlackVault HSM. This endpoint needs to authorize a BlackVault HSM crypto officer and requires a USB flash drive.

Note: If you use the regenerate server certificate endpoint and want to export the server certificate, the export HTTP request must be made after a reboot.

To export the server certificate, complete the following steps:

1. Login to the BlackVault HSM as crypto officer.
2. Insert a USB flash drive into the BlackVault HSM.
3. Make the export server certificate `POST` HTTP request
4. Verify the server certificate is on the root directory of your flash drive

Endpoint URL:

```
https://192.168.1.7:9999/command_export_server_cert
```

4.40. Export Server CSR

The **export server CSR** endpoint will export the server CSR on the BlackVault HSM. This endpoint needs to authorize a BlackVault HSM crypto officer and requires a USB flash drive. To export the server CSR, complete the following steps:

1. Login to the BlackVault HSM as crypto officer.
2. Insert a USB flash drive into the BlackVault HSM.
3. Make the export server CSR `POST` HTTP request.
4. Verify the server CSR is on the root directory of your flash drive

Endpoint URL:

```
https://192.168.1.7:9999/command_export_server_csr
```

4.41. Export Logs

The **export logs** endpoint will export the BlackVault HSM audit logs onto the root directory of a USB flash drive. This endpoint needs to authorize a BlackVault HSM crypto officer. To export the logs, complete the following steps:

1. Login to the BlackVault HSM as crypto officer.
2. Insert a USB flash drive into the BlackVault HSM.
3. Make the export logs `POST` HTTP request.

4. Verify the logs are on the root directory of your flash drive

Endpoint URL:

`https://192.168.1.7:9999/command_export_logs`

4.42. Zeroize

The **zeroize** endpoint should be used with **extreme caution.** This endpoint will wipe all data off a BlackVault HSM, including the user and smartcard databases. The only way to recover the data lost after a zeroize is to restore a user database and import a smartcard database onto a new BlackVault HSM. **These backups must be made prior to a zeroize.**

Endpoint URL:

`https://192.168.1.7:9999/command_zeroize`

4.43. Shutdown

The **shutdown** endpoint will shutdown a BlackVault HSM. This endpoint does not need to authorize a BlackVault HSM operator, the device can be logged in or out.

Endpoint URL:

`https://192.168.1.7:9999/command_shutdown`

4.44. Cancel

The **cancel** endpoint will cancel a BlackVault HSM's current command. This endpoint does not need to authorize a BlackVault HSM operator, the device can be logged in or out.

Endpoint URL:

`https://192.168.1.7:9999/command_cancel`

4.45. Proceed

The **proceed** endpoint is used when the BlackVault HSM needs to verify something to complete a command. In that case, **proceed** means yes and **cancel** means no. This endpoint does not need to authorize a BlackVault HSM operator, the device can be logged in or out. After issuing this HTTP request, the proceeding endpoint may need to authorize an operator, please check the section for that endpoint for the required authorization.

Endpoint URL:

`https://192.168.1.7:9999/command_proceed`

4.46. Create User

The **create user** endpoint performs a `POST` action and, if the crypto officer is logged in, can create a user on a specific slot. To do so, you will need the desired number of smartcards for that user (M of N quorum) and a secure PIN. Please complete the steps below to create a user:

1. Login to the BlackVault HSM as crypto officer.
2. Once logged in as crypto officer, insert a user smartcard (to create).
3. Make the create user `POST` HTTP request (shown below) with the desired slot and M of N quorum.
4. Check the status of the BlackVault HSM, if it displays `"state":"STATE_WAIT_CONFIRM_CARD"`, make a proceed `POST` HTTP request.
5. Make the PIN HTTP request with context `"STATE_CREATE_USER"`
6. Check the status, when it displays `"state":"STATE_REMOVE_CARD"`, remove the smartcard.
7. Continue to check the status and insert the correct number of smartcards for your desired M of N quorum.
8. Once all smartcards have been created, make the reboot HTTP request.
9. Verify you can login to the BlackVault HSM as the new user.

Endpoint URL:

`https://192.168.1.7:9999/command_create_user`

Example Body:

```
"{ \"slot\":1, \"m\":1, \"n\":1 }"
```

4.47. Regenerate Server Certificate

The **regenerate server certificate** endpoint can generate a new server certificate with a desired key type on the BlackVault HSM. To generate the server certificate, the desired key type must be specified in the body of the HTTP request. This endpoint needs to authenticate a crypto officer.

The server certificate key option include:

- `"RSA 2048"`
- `"EC secp384r1"`
- `"EC secp521r1"`

Endpoint URL:

```
https://192.168.1.7:9999/command_regen_server_cert
```

Example Body:

```
"{ \"type\":\"EC secp384r1\" }"
```

4.48. Initialize, Smartcard Database Import

The **initialize** endpoint can only be used when a BlackVault HSM is in the uninitialized state. This should only occur upon receiving a new BlackVault HSM or after a zeroize. When using this endpoint, you will also be creating the crypto officer card set with the desired M of N quorum.

To do an initialization **without a smartcard database import,** complete the following steps:

1. Make the status HTTP request and verify that the device is uninitialized (should return `"operational_state":"STATE_UNINITIALIZED"`).
2. Insert your first crypto officer card (the number of times you create cards depends on your M of N quorum).
3. Make the initialize `POST` HTTP request with your desired configuration.
4. Make the status HTTP request, if it returns `"state":"STATE_CREATE_CO_STATE_WAIT_CONFIRM_CARD"`, make the proceed HTTP request.
5. Make the status HTTP request again, it should return `"state":"STATE_CREATE_CO_STATE_WAIT_PIN"`. Make the PIN HTTP request with context `"STATE_INITIALIZE"` and the desired crypto officer PIN.
6. Check the status, if it displays `"STATE_CREATE_CO_STATE_REMOVE_CARD"`, remove the card. The device will reboot, and initialization is complete.

To do an initialization **with a smartcard database import,** complete the following steps:

Please note this example is for a backup card set quorum of 1 of 1.

1. Make the status HTTP request and verify that the device is uninitialized (should return `"operational_state":"STATE_UNINITIALIZED"`).
2. Insert the first smartcard database export card.
3. Make the initialize `POST` HTTP request, make sure `"import_db"` is set to true.
4. Make the status HTTP request, when it returns `"state":"STATE_IMPORT_CARDS_WAIT_MEDIA"`
5. Make the upload HTTP request with the encrypted user_backup.bv file that was create during the smartcard database export:

```
curl https://192.168.1.7:9999/upload --upload-file backup.bv -H
   'Expect:'
```

6. Make the status HTTP request once more, when you see `"message":"MSG_SUCCESS"`, the process is complete, remove the card, and the device will reboot.

Endpoint URL:

```
'https://192.168.1.7:9999/command_initialize
```

Example Body:

```
"{ \"fips_mode\":true, \"import_db\":false, \"co_m\":1, \"co_n\":1 }"
```

5. Appendix

5.1. Curl Examples

Example `GET` **Endpoint:**

```
curl –location \
     –request GET 'https://192.168.1.7:9999/info' \
     –header 'client_token: fc18f7c8-0758-11ec-9a03-0242ac130003'
```

Example `PUT` **Endpoint:**

```
curl --location \
     --request PUT 'https://192.168.1.7:9999/config_date' \
     --header 'client_token: fc18f7c8-0758-11ec-9a03-0242ac130003' \
     --header 'Content-Type: application/json' \
     --data-raw '{ "day":1, "month":1, "year":2000 }'
```

Example `POST` **Endpoint:**

```
curl --location /
     --request POST 'https://192.168.1.7:9999/command_cancel' \
     --header 'client_token: fc18f7c8-0758-11ec-9a03-0242ac130003' \
     --header 'Content-Type: application/json'
```

5.2. libcurl

Example `GET` **Endpoint:**

```
CURL *curl;
CURLcode res;
curl = curl_easy_init();
if(curl) {
  curl_easy_setopt(curl, CURLOPT_CUSTOMREQUEST, "GET");
  curl_easy_setopt(curl, CURLOPT_URL, "https://192.168.1.7:9999/info");
  curl_easy_setopt(curl, CURLOPT_FOLLOWLOCATION, 1L);
  curl_easy_setopt(curl, CURLOPT_DEFAULT_PROTOCOL, "https");
  struct curl_slist *headers = NULL;
```

```
  headers = curl_slist_append(headers, "Content-Type: application/json");
  headers = curl_slist_append(headers, "client_token: fc18f7c8-0758-11ec-
9a03-0242ac130003");
  curl_easy_setopt(curl, CURLOPT_HTTPHEADER, headers);
  res = curl_easy_perform(curl);
}
curl_easy_cleanup(curl);
```

Example `PUT` **Endpoint:**

```
CURL *curl;
CURLcode res;
curl = curl_easy_init();
if(curl) {
  curl_easy_setopt(curl, CURLOPT_CUSTOMREQUEST, "PUT");
  curl_easy_setopt(curl, CURLOPT_URL,
"https://192.168.1.7:9999/config_date");
  curl_easy_setopt(curl, CURLOPT_FOLLOWLOCATION, 1L);
  curl_easy_setopt(curl, CURLOPT_DEFAULT_PROTOCOL, "https");
  struct curl_slist *headers = NULL;
  headers = curl_slist_append(headers, "client_token: fc18f7c8-0758-11ec-
9a03-0242ac130003");
  headers = curl_slist_append(headers, "Content-Type: application/json");
  curl_easy_setopt(curl, CURLOPT_HTTPHEADER, headers);
  const char *data = "{ \"day\": 12, \"month\": 08, \"year\": 2021 }";
  curl_easy_setopt(curl, CURLOPT_POSTFIELDS, data);
  res = curl_easy_perform(curl);
}
curl_easy_cleanup(curl);
```

Example `POST` **Endpoint:**

```
CURL *curl;

CURLcode res;

curl = curl_easy_init();

if(curl) {

  curl_easy_setopt(curl, CURLOPT_CUSTOMREQUEST, "POST");

  curl_easy_setopt(curl, CURLOPT_URL,
"https://192.168.1.7:9999/command_cancel");

  curl_easy_setopt(curl, CURLOPT_FOLLOWLOCATION, 1L);

  curl_easy_setopt(curl, CURLOPT_DEFAULT_PROTOCOL, "https");

  struct curl_slist *headers = NULL;

  headers = curl_slist_append(headers, "client_token: 1234");

  curl_easy_setopt(curl, CURLOPT_HTTPHEADER, headers);

  const char *data = "";

  curl_easy_setopt(curl, CURLOPT_POSTFIELDS, data);

  res = curl_easy_perform(curl);

}

curl_easy_cleanup(curl);
```